A. Algorithms

In order to conduct a fair comparison with the baseline algorithms regarding the reduction of number of training iterations, and demonstrate that REPAINT improves the sample efficiency in knowledge transfer, we use an alternating variant of REPAINT with Clipped PPO in the experiments. The algorithm is provided in Algorithm 2, where we adopt on-policy representation transfer and off-policy instance transfer alternately, so that the REPAINT performs policy update one time per iteration using less samples. Indeed, Algorithm 2 can be easily extended with different alternating ratios other than 1:1 alternating. The corresponding results and discussion can be found in Section C.1.

Note that in Algorithm 1 and Algorithm 2, we can use different learning rates α_1 and α_2 to control the update from representation transfer and instance transfer, respectively. Moreover, it is straightforward to using multiple and different teacher policies in each transfer step, and our algorithm can be directly applied to any advantage-based policy gradient RL algorithms. Assume there are m previously trained teacher policies $\pi_1, \ldots, \pi_m$. In the instance transfer, we can form the replay buffer $\widetilde{\mathcal{S}}$ by collecting samples from all teacher policies. Then in the representation transfer, the objective function can be written in a more general way:

$$L^k_{\text{rep}}(\theta) = L_{\text{clip}}(\theta) - \sum_{i=1}^{m} \beta_i^k H\left(\pi_i(a|s) \| \pi_\theta(a|s)\right) , \quad \text{(A.1)}$$

where we can impose different weighting parameters for different teacher policies.

In addition, the first term in (A.1), i.e., $L_{\text{clip}}(\theta)$, can be naturally replaced by the objective of other RL algorithms, e.g., Advantage Actor-Critic (A2C) (Sutton et al., 2000):

$$L_{\text{A2C}}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a|s) \hat{A}_t\right] ,$$

and Trust Region Policy Optimization (TRPO) (Schulman et al., 2015a):

$$L_{\text{TRPO}}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \hat{A}_t - \beta \text{KL}[\pi_{\theta_{\text{old}}}(\cdot|s), \pi_\theta(\cdot|s)]\right]$$

for some coefficient β of the maximum KL divergence computed over states.

REPAINT can also be adapted to other policy-gradient-based algorithms that are not based on advantage values. To this end, one can define a different metric for relatedness. For example, we can use REPAINT with REINFORCE (Williams, 1992), by defining the relatedness metric to be $\hat{R} - b$, where $\hat{R}$ is the off-policy return and b is the baseline function in REINFORCE, which can be state-dependent. Then the experience selection approach can be built based on the new relatedness metric.

Algorithm 2 Alternating REPAINT with Clipped PPO

Initialize parameters ν, θ
Load teacher policy $\pi_{\text{teacher}}(\cdot)$
Set hyper-parameters ζ, α_1, α_2, and β_k's in (4.1)
for iteration $k = 1, 2, \ldots$ **do**
  if k is odd **then** *// representation transfer*
    Collect samples $\mathcal{S} = \{(s, a, s', r)\}$ using $\pi_{\theta_{\text{old}}}(\cdot)$
    Fit state-value network V_ν using $\mathcal{S}$ to update ν
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
    Compute sample gradient of $L^k_{\text{rep}}(\theta)$ in (4.1)
    Update policy network by $\theta \leftarrow \theta + \alpha_1 \nabla_\theta L^k_{\text{rep}}(\theta)$
  else *// instance transfer*
    Collect samples $\widetilde{\mathcal{S}} = \{(\widetilde{s}, \widetilde{a}, \widetilde{s}', \widetilde{r})\}$ using $\pi_{\text{teacher}}(\cdot)$
    Compute advantage estimates $\hat{A}'_1, \ldots, \hat{A}'_{T'}$
    for t=1,...,T' **do** *// experience selection*
      if $\hat{A}'_t < \zeta$ **then**
        Remove $\hat{A}'_t$ and the corresponding transition $(\widetilde{s}_t, \widetilde{a}_t, \widetilde{s}_{t+1}, \widetilde{r}_t)$ from $\widetilde{\mathcal{S}}$
    Compute sample gradient of $L_{\text{ins}}(\theta)$ in (4.2)
    Update policy network by $\theta \leftarrow \theta + \alpha_2 \nabla_\theta L_{\text{ins}}(\theta)$

We now discuss how the REPAINT algorithm can be extended to Q-learning. Since Q-learning is an off-policy algorithm, it is easy to notice that the kickstarting approach cannot be directly used. Although some other representation transfer approaches are suitable for Q-learning, e.g., using a neural network for feature abstraction, we skip the discussion as it is not our goal in this paper. Instead, we will focus on how to extend our experience selection approach to the instance transfer of Q-learning.

In the instance transfer for Q-learning, the Q-value network, parameterized by ϕ, is updated by minimizing the following loss function:

$$L(\phi) = \frac{1}{2} \sum_i \|Q_\phi(s_i, a_i) - y_i\|^2 ,$$

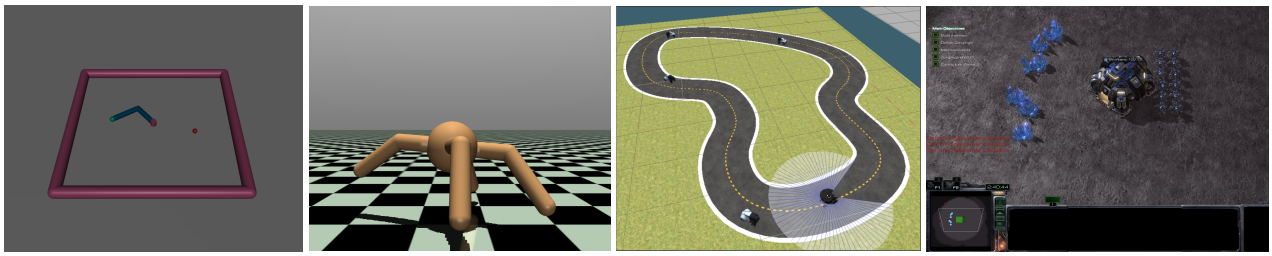
where the samples are drawn from the replay buffer that is collected following some teacher policy, and the target values y_i's are defined by

$$y_i = r(s_i, a_i) + \gamma \max_{a'_i} Q_\phi(s'_i, a'_i) ,$$

with γ the discount factor. Now given some threshold $\zeta \geq 0$, we can select the samples that satisfy the following condition to update ϕ:

$$y_i - Q_\phi(s_i, a_i) > \zeta .$$

Similar to REPAINT with actor-critic methods, the threshold ζ here is task specific, but it needs more careful treatment in Q-learning. Since we aim to obtain an optimal Q-function $Q^*(s, a)$, we should use a ζ such that $Q^*(s, a) \geq y > Q_\phi(s, a) + \zeta$. For actor-critic methods, we can empirically



(a) MuJoCo-Reacher (b) MuJoCo-Ant (c) DeepRacer multi-car racing (d) StarCraft II

Figure 9. The simulation environments used in the experiments. Note that in DeepRacer multi-car racing, the racing car is equipped by a stereo camera and a Lidar, as shown in (c). However, in the single-car time trial, the racing car only has a monocular camera. In addition, for the StarCraft II environment, 40 Zerglings are randomly generated in the unrevealed areas around the map.

show that REPAINT is robust to the advantage threshold. However, for Q-learning, we usually need to set ζ to be very small and use the experience selection only in the early training stage. Motivated by Oh et al. (2018), we can also set $\zeta = 0$. Then the convergence of Q-value follows $Q^*(s,a) \geq y > Q_\phi(s,a)$. By filtering out the samples such that $Q_\phi(s_i,a_i) \geq y_i$, one can expect the instance transfer to improve the sample efficiency and reduce the total training time for complex target tasks.

In practice, to trade off the exploitation with exploration in Q-learning, we can collect some samples following the ϵ-greedy policy from the online Q-value network, and add those samples to the replay buffer as well.

B. Details of Experimental Setup

B.1. Environments

We now provide the details of our experimental setup. The graphical illustration of the environments used is presented in Figure 9. First of all, MuJoCo is a well-known physics simulator for evaluating agents on continuous motor control tasks with contact dynamics, hence we omit the further description of MuJoCo in this paper.

DeepRacer simulator. In AWS DeepRacer simulator[3], the RL agent, i.e., an autonomous car, learns to drive by interacting with its environment, e.g., the track with moving bot cars, by taking an action in a given state to maximize the expected reward. Figure 9(c) presents the environmental setting for racing against moving bot cars, where four bot cars are generated randomly on the track and the RL agent learns to finish the lap with overtaking bot cars. Another racing mode we used in this paper is the single-car time-trial race, where the goal is to finish a lap in the shortest time.

In single-car racing, we only install a front-facing camera on the RL agent, which obtains an RGB image with size $120 \times 160 \times 3$. The image is then transformed to gray scale and fed into an input embedder. For simplicity, the input embedder is set to be a three-layer convolutional neural network (CNN) (Goodfellow et al., 2016). For the RL agent in racing against bot cars, we use a stereo camera and a Lidar as sensors. The stereo camera obtains two images simultaneously, transformed to gray scale, and concatenates the two images as the input, which leads to a $120 \times 160 \times 2$ input tensor. The input embedder for stereo camera is also a three-layer CNN by default. The stereo camera is used to detect bot cars in the front of learner car, while the Lidar is used to detect any car behind. The backward-facing Lidar has an angle range of 300 degree and a 64 dimensional signal. Each laser can detect a distance from 12cm to 1 meter. The input embedder for Lidar sensor is set to be a two-layer dense network. In both environments, the output has two heads, V head for state value function output and policy head for the policy function output, each of which is set to be a two-layer dense networks but with different output dimensions. The action space consists of a combination of five different steering angles and two different throttle degrees, which forms a 10-action discrete space. In the evaluation of DeepRacer experiments, the generalization around nearby states and actions is also considered (Balaji et al., 2019), where we add small noises to the observations and actions.

StarCraft II learning environments (SC2LE). The *BuildMarines* mini-game is shown in Figure 9(d), where it limits the possible actions that the agent can take to either of selecting points, building workers, building supply depots, building barracks, and training marines. For *BuildMarines+FindAndDefeatZerglings* (*BM+FDZ*), we extend the action space to allow the agent to select the army and to attack with the army. As mentioned before, we keep the state and action spaces the same between source and target tasks in the experiments. Therefore, we provide the two army-related actions in *BuildMarines* but they are always unavailable.

[3] https://github.com/awslabs/amazon-sagemaker-examples/tree/master/reinforcement_learning/rl_deepracer_robomaker_coach_gazebo

Table 2. StarCraft II *BM+FDZ* reward scheme.

Condition	Reward
Performing an unavailable action	-0.01
A marine dying	-4
Selecting an unavailable point	-0.01
Selecting an available point	0.01
Training an SCV	0.5
Building barrack	0.2

The default observations provided in the SC2LE are used and we follow a similar network architecture to the baseline presented in Vinyals et al. (2017). Spatial features, including screen features (size = $84 \times 84 \times 9$) and mini-map features (size = $84 \times 84 \times 3$), are each fed through input embedders consisting of CNN with two layers. Non-spatial features, including the measurements (size = 5; e.g., mineral count, food count, army count) and the one-hot encoded vector of available actions (size = 7, e.g., build worker, select screen) are fed into an input embedder consisting of linear layers with a $tanh$ activation.

The action space consists of one discrete action to determine the command to take (i.e., build supply depot, build barrack, train SCV, train marine, select point on screen, attack point on screen, select army) and two actions to indicate where to commence the action on the screen (spatial action). For example, with a command to build a barrack, the spatial action determines where the barrack will be built, and to attack the spatial action determines where the marines will attack. It is important to note that not all commands rely on a corresponding spatial action. For example, when issuing a command to train a marine, the spatial action is ignored.

In addition, several rules are implemented to ensure that the mini-game progresses as expected. Firstly, workers (SCVs) cannot attack so that the agent will not find and attempt to defeat Zerglings with the workers. Secondly, Zerglings cannot enter the base so that Zerglings do not overrun the base before marines are built.

The *BM+FDZ* agent is rewarded for each marine built and each Zergling killed. Specifically, a +5 reward is imposed when a marine is built and +10 reward when a Zergling is killed. Table 2 shows small rewards and penalties that are given to facilitate the agent to achieve these goals.

B.2. Hyper-parameters

We have implemented our algorithms based on Intel Coach[4]. The MuJoCo environments are from OpenAI Gym[5]. The StarCraft II learning environments are from DeepMind's PySC2[6]. Regarding the advantage estimates, we use the generalized advantage estimator (GAE) (Schulman et al., 2015b). If not specified explicitly in the paper, we always use Adam as the optimizer with minibatch size as 64, clipping parameter ϵ as 0.2, and $\beta_{k+1} = 0.95\beta_k$ throughout the experiments. The other hyper-parameters are presented in Tables 3-5.

Table 3. Hyper-parameters used in the MuJoCo simulations.

Hyperparameter	Value
Num. of rollout steps	2048
Num. training epochs	10
Discount (γ)	0.99
Learning rate	3e-4
GAE parameter (λ)	0.95
Beta entropy	0.0001
Cross-entropy weight (β_0)	0.2
Reacher - Advantage Threshold (ζ)	0.8
Reacher - Num. REPAINT iterations	15
Ant - Num. REPAINT iterations	50

Table 4. Hyper-parameters used in the DeepRacer simulations.

Hyperparameter	Value
Num. of rollout episodes	20
Num. of rollout episodes when using π_{teacher}	2
Num. training epochs	8
Discount (γ)	0.999
Learning rate	3e-4
GAE parameter (λ)	0.95
Beta entropy	0.001
Cross-entropy weight (β_0)	0.2
Advantage Threshold (ζ)	0.2
Single-car - Num. REPAINT iterations	4
Multi-car - Num. REPAINT iterations	20

Table 5. Hyper-parameters used in the StarCraft II simulations.

Hyperparameter	Value
Num. of rollout episodes	2
Num. of rollout episodes when using π_{teacher}	2
Num. training epochs	6
Discount (γ)	0.99
Learning rate	3e-5
GAE parameter (λ)	0.95
Beta entropy	0.01
Cross-entropy weight (β_0)	0.1
Advantage Threshold (ζ)	0.2
Num. REPAINT iterations	25

[4] https://github.com/NervanaSystems/coach
[5] https://gym.openai.com/envs/#mujoco
[6] https://github.com/deepmind/pysc2

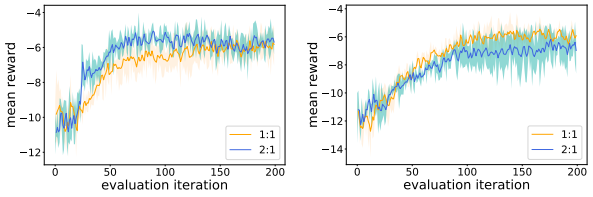


Figure 10. Evaluation performance for MuJoCo-Reacher, averaged across five runs. Left: Teacher task is similar to the target task. Right: Teacher task is dissimilar to the target task.

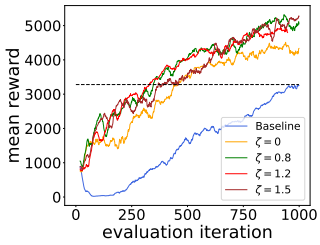


Figure 11. REPAINT performance for MuJoCo-Ant with different advantage thresholds in experience selection, averaged across three runs. Same teacher policy is used for all thresholds.

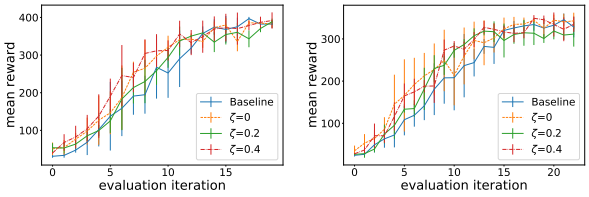
C. Extensive Experimental Results

C.1. Discussion on Alternating Ratios

In Algorithm 2, we alternate representation transfer and instance transfer after each iteration. Here, we aim to illustrate the effect of using different alternating ratios by the MuJoCo-Reacher environment. We compare the 1:1 alternating with a 2:1 ratio, namely, two on-policy representation transfer (kickstarting) iterations before and after an off-policy instance transfer iteration. The evaluation performance is shown in Figure 10. When the teacher task is similar to the target task, adopting more kickstarted training iterations leads to faster convergence, due to the policy distillation term in the loss function. On the other hand, when the task similarity is low, instance transfer contributes more to the knowledge transfer due to the advantage-based experience selection. Therefore, we suggest to set the alternating ratio in Algorithm 2, or the α_1 and α_2 parameters in Algorithm 1 and Algorithm 2, according to the task similarity between source and target tasks. However, the task similarity is usually unknown in most of the real-world applications, or the similarities are mixed when using multiple teacher policies. It is interesting to automatically learn the task similarity and determine the best ratio/parameters before actually starting the transfer learning. We leave the investigation of this topic as a future work.

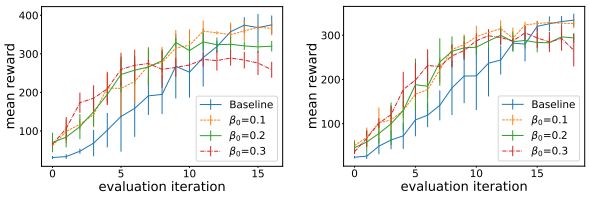
C.2. Advantage Threshold Robustness in MuJoCo-Ant

Figure 11 indicates that our REPAINT algorithm is robust to the threshold parameter when $\zeta > 0$. Similar learning progresses are observed from training with different ζ values.



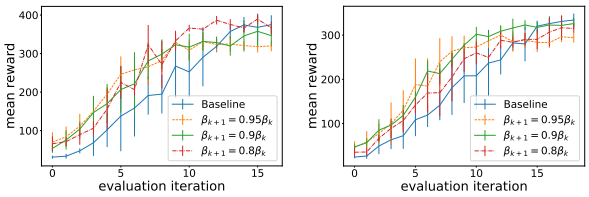
(a) Outer-lane task with inner-lane teacher (b) Inner-lane task with outer-lane teacher

Figure 12. Evaluation performance with respect to different ζ's, averaged across five runs.



(a) Outer-lane task with inner-lane teacher (b) Inner-lane task with outer-lane teacher

Figure 13. Evaluation performance with respect to different initial β_0's, averaged across five runs. Here we fix the β update to be $\beta_{k+1} = 0.95\beta_k$.



(a) Outer-lane task with inner-lane teacher (b) Inner-lane task with outer-lane teacher

Figure 14. Evaluation performance with respect to different β schedules, averaged across five runs.

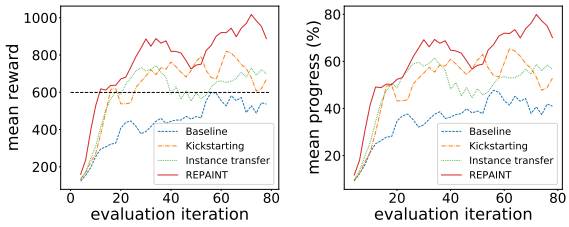
C.3. More Results on DeepRacer Single-car Time Trial

In the DeepRacer single-car time-trial task, we also study the effect of different cross-entropy weights β_k and instance filtering thresholds ζ, as mentioned in the paper. We first present the results of instance transfer learning with different ζ values in Figure 12, where we can again see that our proposed advantage-based experience replay is robust to the threshold parameter.

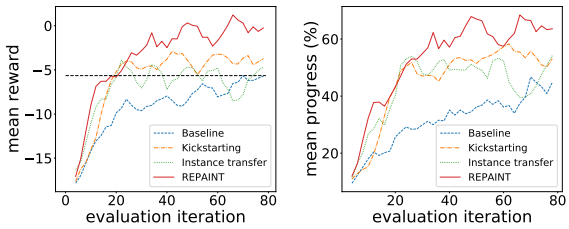
We then study the performance of training with different cross-entropy loss weights β_k. First, we fix the diminishing factor to be 0.95, namely, $\beta_{k+1} = 0.95\beta_k$, and test different β_0's. From Figure 13, we can see that training with all β_0 values can improve the initial performance compared to the baseline. However, when the teacher task is different

Table 6. Summary of wall-clock time of experiments.

Env.	Training hardware	Teacher type	Target score	$T_{Baseline}$ (hrs)	T_{KS} (pct. reduced)	T_{IT} (pct. reduced)	$T_{REPAINT}$ (pct. reduced)
Reacher	laptop	similar	-7.4	2.1	0.6 (71.4%)	1.1 (47.6%)	0.4 (81.0%)
		different			0.9 (57.1%)	1.4 (33.3%)	0.6 (71.4%)
Ant	laptop	similar	3685	19.1	8.0 (58.1%)	12.8 (33.0%)	7.5 (60.7%)
Single-car	AWS, p2	different	394	2.2	Not achieved	Not achieved	1.5 (31.8%)
	AWS, p2	different	345	2.3	Not achieved	Not achieved	1.5 (34.8%)
Multi-car	AWS, p2	sub-task	1481	16.4	4.8 (70.7%)	12.6 (23.2%)	4.5 (72.6%)
	AWS, p2	diff/sub-task	2.7	9.6	9.3 (3.1%)	8.3 (13.5%)	3.7 (61.5%)

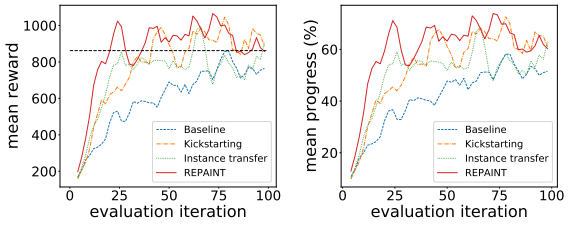


(a) Task with advanced reward

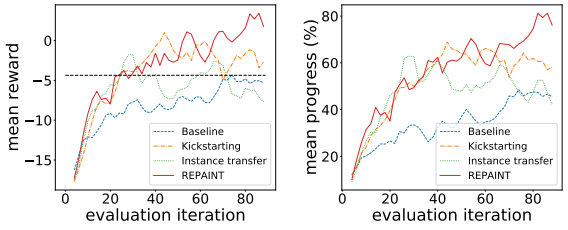


(b) Task with progress-based reward

Figure 15. Evaluation performance for DeepRacer multi-car racing against bot cars, using 4-layer CNN. The plots are smoothed for visibility.



(a) Task with advanced reward



(b) Task with progress-based reward

Figure 16. Evaluation performance for DeepRacer multi-car racing against bot cars, using 5-layer CNN. The plots are smoothed for visibility.

from the target task, larger β_0 values, like 0.3, may reduce the agent's asymptotic performance since the agent overshoots learning from teacher policy. In addition, we then fix $\beta_0 = 0.2$ and test different β_k schedules. The results are shown in Figure 14. We can observe some trade-offs between training convergence time and final performance. By reducing the β values faster, one can improve the final performance but increase the training time that needed to achieve some certain performance level. It is of interest to automatically determine the best β_k values during training, which needs further investigation. We leave it as another future work.

C.4. Neural Network Architectures

For completeness of the experiments, we also provide some results regarding different neural network architectures in this section. Take the DeepRacer task of multi-car racing against bot cars as an example, we have used three-layer CNN as the default architecture in experiments. Here, we present the comparison of REPAINT against other baselines with the evaluation performance using four-layer CNN (Figure 15) and five-layer CNN (Figure 16).

C.5. Summary of Wall-Clock Training Time

In addition to the summary of reduction performance with respect to number of training iterations presented in Table 1, we also provide the data of wall-clock time in Table 6. Note that we run StarCraft II experiments using different laptops, the comparison might not be convincing, and hence is omitted here. Again, we can see a significant reduction by training with REPAINT, which reaches at least 60% besides the DeepRacer single-car time trial. The kickstarted training performs well when a similar teacher policy is used. Although training with only instance transfer cannot boost the initial performance, it still reduces the training cost to achieve some specific performance level.

D. Convergence of Off-policy Instance Transfer

In order to apply the two time-scale stochastic approximation theory (Bhatnagar et al., 2009; Karmakar & Bhatnagar, 2018) for the convergence proof, the off-policy instance transfer learning is required to satisfy Assumptions (A1)-(A7) in Holzleitner et al. (2020). We now discuss what assumptions we need to impose and how our instance transfer meets those properties.

First of all, regarding Assumptions (A1) and (A7), we can add some regularization terms in practice. For example, in our experiments for this paper, we have added weight decay, entropy regularization, and KL divergence terms.

Similar to Holzleitner et al. (2020), in order to satisfy Assumptions (A2) and (A6), we need to make assumptions on the loss functions for actor and critic, i.e., Assumptions (L1)-(L3) in Holzleitner et al. (2020). As mentioned before, the actor loss is denoted by $J_{\text{ins}}(\theta)$. Since the Q-function estimates in J_{ins} involves the critic function, we denote the loss by $J_{\text{ins}}(\theta, \nu)$. We also denote the critic loss by $J_{\text{critic}}(\theta, \nu)$. Since the actor π_θ and the critic V_ν are approximated by deep neural networks, they are considered to be sufficiently smooth. Moreover, we should also assume sufficient smoothness for the two loss functions.

Assumption D.1. The loss functions $J_{\text{ins}}(\theta, \nu)$ and $J_{\text{critic}}(\theta, \nu)$ have compact support and are at least three times continuously differentiable with respect to θ and ν.

Next, for each starting point (θ_0, ν_0), we want to find a neighborhood such that it contains only one critical point. Therefore, we further make the following two assumptions.

Assumption D.2. For each θ, all critical points of $J_{\text{critic}}(\theta, \nu)$ are isolated local minima and there are only finitely many. The local minima $\{\lambda_i(\theta)\}_{i=1}^{k(\theta)}$ can be expressed locally as at least twice continuously differentiable functions with associated domains of definitions $\{W_{\lambda_i(\theta)}\}_{i=1}^{k(\theta)}$.

Assumption D.3. Locally in $W_{\lambda_i(\theta)}$, $J_{\text{ins}}(\theta, \lambda_i(\theta))$ has only one local minimum.

Based on the above assumptions, for a fixed starting point (θ_0, ν_0), we can construct a neighborhood $W_0 \times U_0$, which contains unique local minimum. Assumption (A3) is not related to the instance transfer developed here, hence is omitted. We can either make the assumption explicitly for the update process, or follow the treatment mentioned in Holzleitner et al. (2020), e.g., using online stochastic gradient descent (SGD) for update.

The next assumption we need to make is on the learning rates, i.e., Assumption (A4) in Holzleitner et al. (2020). Denote the learning rates for actor and critic by a_k and b_k, respectively.

Assumption D.4. The learning rates a_k and b_k should satisfy:

$$\sum_k a_k = \infty, \quad \sum_k a_k^2 < \infty,$$
$$\sum_k b_k = \infty, \quad \sum_k b_k^2 < \infty,$$

and $\lim_{k\to\infty} a_k/b_k = 0$. Moreover, a_k and b_k are non-increasing for all $k \geq 0$.

At last, Assumption (A5) is satisfied as long as the transition kernels for the MDPs are continuous with respect to the weak topology in the space of probability measures. Therefore, after imposing Assumptions D.1-D.4, we can directly follow the two time-scale stochastic approximation theory (Karmakar & Bhatnagar, 2018) and get that our proposed off-policy instance transfer can converge to some local optimum almost surely under the assumptions.

E. Convergence Rate and Sample Complexity for REPAINT

The analysis and proof in this section is adapted from Kumar et al. (2019). Without loss of generality, we first assume that the teacher (source) task and student (target) task share the same state and action spaces $\mathcal{S} \times \mathcal{A}$. Then we make the following assumptions on the regularity of the student task and the parameterized student policy π_θ.

Assumption E.1. The reward function for student task is uniformly bounded. Namely, denote the reward by R_{student}. Then there exists a positive constant U_{student}, such that $R_{\text{student}}(s, a) \in [0, U_{\text{student}}]$ for any $(s, a) \in \mathcal{S} \times \mathcal{A}$.

Since the policy (actor) is parameterized by neural networks, it is easy to see that π_θ is differentiable. In addition, we make an assumption on the corresponding score function.

Assumption E.2. The score function $\nabla \log \pi_\theta(a|s)$ is Lipschitz continuous and has bounded norm, namely, for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, there exist positive constants L_Θ and B_Θ, such that

$$||\nabla \log \pi_{\theta_1}(a|s) - \nabla \log \pi_{\theta_2}(a|s)|| \leq L_\Theta \|\theta_1 - \theta_2\|, \forall \theta_1, \theta_2, \tag{E.1}$$

and

$$||\nabla \log \pi_\theta(a|s)|| \leq B_\Theta, \forall \theta. \tag{E.2}$$

Note that by the above two assumptions, one can also obtain that the corresponding Q-function and objective function are also absolutely upper bounded. In order to prove our theorem, we also need the following i.i.d. assumption.

Assumption E.3. In both teacher and student tasks, the random tuples $(s_t, a_t, s'_t, a'_t), t = 0, 1, \dots$ are drawn from the stationary distribution of the Markov reward process independently across time.

In practice, the i.i.d assumption does not hold (Dalal et al., 2017). But it is common when dealing with the convergence bounds in RL.

To ensure the Q-function evaluation and the stochastic estimate of the gradient unbiased, we consider the case where the Q-function admits a linear parameterization of the form $\hat{Q}^{\pi_\theta}(s,a) = \xi^T \varphi(s,a)$ where ξ is a finite vector of real numbers of size p and $\varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^p$ is a nonlinear feature map. In practice, we normalize the feature representation to guarantee the feature norm is bounded. Therefore, we can assume the norm boundedness of the feature map.

Assumption E.4. For any state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, the norm of the student's feature representation $\varphi(s,a)$ is bounded by a constant C_{student}.

To simplify the proofs, we will consider that the experiences are not filtered. The experience filtering results in possibly biased estimate of the gradient and impacts the variance bounds (Greensmith et al., 2004). Next, we will assume that the update of the critic (Q-function) converges by some rate.

Assumption E.5. The expected error of the critic parameter for the student task is bounded by $O(k^{-b})$ for some $b \in (0,1]$, i.e., there exists a positive constant L_1, such that

$$\mathbb{E}(||\xi_k - \xi_\star||) \leq L_1 k^{-b}. \tag{E.3}$$

Now we consider the update for actor in REPAINT. Assume that the learning rates α_1 and α_2 are also iteration dependent. Then we rewrite the actor update in Algorithm 1 as

$$\theta_{k+1} = \theta_k + \alpha_{1,k} \nabla_\theta J_{\text{rep}}(\theta_k) + \alpha_{2,k} \nabla_\theta J_{\text{ins}}(\theta_k). \tag{E.4}$$

For on-policy representation transfer, the gradient is defined by

$$\nabla J_{\text{rep}}(\theta) = \nabla J_{\text{RL}}(\theta) - \beta_k \nabla J_{\text{aux}}(\theta). \tag{E.5}$$

More specifically,

$$\nabla J_{\text{RL}}(\theta) = \mathbb{E}_{\substack{s \sim d_{\text{student}} \\ a \sim \pi_{\text{student}}}} [\nabla \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)], \tag{E.6}$$

where d_{student} is the limiting distribution of states under π_θ, and

$$\nabla J_{\text{aux}}(\theta) = \nabla H(\pi_{\text{teacher}} || \pi_\theta) = -\mathbb{E}_{\pi_{\text{teacher}}} [\nabla_\theta \log \pi_\theta(a|s)]. \tag{E.7}$$

For simplicity, we assume $\beta_k = 0$ for all $k \geq 0$. Namely, we ignore the cross-entropy term in the proof of our theorem. However, we will present the extension of $\beta_k > 0$ cases later. Then the stochastic estimate of the gradient is unbiased when the Q-function evaluation is unbiased, and is given by

$$\hat{\nabla} J_{\text{rep}}(\theta) = \hat{Q}^{\pi_\theta}(s_T, a_T) \nabla \log \pi_\theta(a_T|s_T), \tag{E.8}$$

where s_T, a_T is the state-action pair collected following the student policy π_θ with some time step T.

The derivation of the instance transfer gradient estimate is similar to the off-policy actor-critic (Degris et al., 2012), which is defined as

$$\hat{\nabla} J_{\text{ins}}(\theta) = \frac{\pi_\theta(\widetilde{a}_T|\widetilde{s}_T)}{\pi_{\text{teacher}}} \hat{Q}^{\pi_\theta}(\widetilde{s}_T, \widetilde{a}_T) \nabla \log \pi_\theta(\widetilde{a}_T|\widetilde{s}_T), \tag{E.9}$$

with some sample $\widetilde{s}_T, \widetilde{a}_T$ collected following the teacher policy π_{teacher}.

In summary, when updating the actor network. We collect rollouts following both teacher policy and student policy, and randomly select two samples for the following online update:

$$\begin{aligned}\theta_{k+1} - \theta_k &= \alpha_{1,k} \hat{Q}^{\pi_\theta}(s_{T_k}, a_{T_k}) \nabla \log \pi_\theta(a_{T_k}|s_{T_k}) \\ &+ \alpha_{2,k} \frac{\pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})}{\pi_{\text{teacher}}(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})} \hat{Q}^{\pi_\theta}(\widetilde{s}_{T_k}, \widetilde{a}_{T_k}) \nabla \log \pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k}).\end{aligned} \tag{E.10}$$

Next, we assume that the estimate of the objectives' gradient conditioned on some filtration is bounded by some finite variance.

Assumption E.6. Let $\hat{\nabla} L_{\text{rep}}(\theta)$ and $\hat{\nabla} L_{\text{ins}}(\theta)$ be the estimators of $\nabla L_{\text{rep}}(\theta)$ and $\nabla L_{\text{ins}}(\theta)$, respectively. Then, there exist finite σ_{rep} and σ_{ins} such that

$$\mathbb{E}(||\hat{\nabla} L_{\text{rep}}(\theta)||^2 | \mathcal{F}_k) \leq \tfrac{\sigma_{\text{rep}}^2}{4}, \tag{E.11}$$

$$\mathbb{E}(||\hat{\nabla} L_{\text{ins}}(\theta)||^2 | \mathcal{F}_k) \leq \tfrac{\sigma_{\text{ins}}^2}{4}. \tag{E.12}$$

Since the teacher policy is a deterministic policy distribution. It is also common to assume some boundedness for π_{teacher} (Degris et al., 2012).

Assumption E.7. The teacher policy has a minimum positive value $b_{\min} \in (0,1]$, such that $\pi_{\text{teacher}}(a|s) \geq b_{\min}$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Now we have stated all assumptions that are needed for deriving the convergence rate and sample complexity. Next, we introduce the proofs of two lemmas. The first lemma is on the Lipschitz continuity of the objective gradients. The proof can be found in, e.g., Zhang et al. (2020).

Lemma E.8. The objective gradients ∇J_{rep} and ∇J_{ins} are Lipschitz continuous, namely, there exist constants L_{rep} and L_{ins}, such that for any θ_1 and θ_2,

$$||\nabla J_{\text{rep}}(\theta_1) - \nabla J_{\text{rep}}(\theta_2)|| \leq L_{\text{rep}} ||\theta_1 - \theta_2||, \tag{E.13}$$

$$||\nabla J_{\text{ins}}(\theta_1) - \nabla J_{\text{ins}}(\theta_2)|| \leq L_{\text{ins}} ||\theta_1 - \theta_2||. \tag{E.14}$$

For simplicity, we can let $L := \max(L_{\text{rep}}, L_{\text{ins}})$, so L is the Lipschitz constant for both inequalities above. Next we will derive an approximate ascent lemma for a random variable W_k defined by

$$\begin{aligned} W_k =& J_{\text{rep}}(\theta_k) + J_{\text{ins}}(\theta_k) \\ & - L\left(\sigma_{\text{rep}}^2 \sum_{j=k}^{\infty} \alpha_{1,j}^2 + \sigma_{\text{ins}}^2 \sum_{j=k}^{\infty} \alpha_{2,j}^2\right). \end{aligned} \tag{E.15}$$

Since the rewards and score functions are bounded above (see Assumptions E.1 and E.2), then we can also get there exist constants C_{rep} and C_{ins}, such that

$$||\nabla J_{\text{rep}}|| \le C_{\text{rep}} \quad \text{and} \quad ||\nabla J_{\text{ins}}|| \le C_{\text{ins}}. \tag{E.16}$$

Lemma E.9. The sequence W_k defined above satisfies the inequality

$$\begin{aligned} &\mathbb{E}[W_{k+1}|\mathcal{F}_k] \ge W_k \\ &- (C_{\text{rep}} + C_{\text{ins}}) C_{\text{student}} B_\Theta (\alpha_{1,k} + \frac{\alpha_{2,k}}{b_{\min}}) \mathbb{E}[||\xi_k - \xi_*|||\mathcal{F}_k] \\ &\quad + \alpha_{1,k}||\nabla J_{\text{rep}}(\theta_k)||^2 + \alpha_{2,k}||\nabla J_{\text{ins}}(\theta_k)||^2 \\ &\quad + (\alpha_{1,k} + \alpha_{2,k}) \nabla J_{\text{rep}}(\theta_k)^\top \nabla J_{\text{ins}}(\theta_k) \end{aligned} \tag{E.17}$$

Proof. By definition, we can write

$$\begin{aligned} W_{k+1} =& J_{\text{rep}}(\theta_{k+1}) + J_{\text{ins}}(\theta_{k+1}) \\ & - L\left(\sigma_{\text{rep}}^2 \sum_{j=k+1}^{\infty} \alpha_{1,j}^2 + \sigma_{\text{ins}}^2 \sum_{j=k+1}^{\infty} \alpha_{2,j}^2\right). \end{aligned} \tag{E.18}$$

By the Mean Value Theorem, there exists $\widetilde{\theta}_k \in [\theta_k, \theta_{k+1}]$, such that

$$J_{\text{rep}}(\theta_{k+1}) = J_{\text{rep}}(\theta_k) + (\theta_{k+1} - \theta_k)^\top \nabla J_{\text{rep}}(\widetilde{\theta}_k). \tag{E.19}$$

By Cauchy Schwartz inequality, we have

$$\begin{aligned} &(\theta_{k+1} - \theta_k)^\top (\nabla J_{\text{rep}}(\widetilde{\theta}_k) - \nabla J_{\text{rep}}(\theta_k)) \\ \ge& - \|\theta_{k+1} - \theta_k\| \|\nabla J_{\text{rep}}(\widetilde{\theta}_k) - \nabla J_{\text{rep}}(\theta_k)\| \\ \ge& - L_{\text{rep}} \|\theta_{k+1} - \theta_k\|^2 \\ \ge& - L \|\theta_{k+1} - \theta_k\|^2. \end{aligned} \tag{E.20}$$

After similar treatment for $J_{\text{ins}}(\theta)$, we can get

$$\begin{aligned} W_{k+1} \ge& W_k + (\theta_{k+1} - \theta_k)^\top (\nabla J_{\text{rep}}(\theta_k) + J_{\text{ins}}(\theta_k)) \\ & - 2L \|\theta_{k+1} - \theta_k\|^2. \end{aligned} \tag{E.21}$$

Take the expectation with respect to the filtration $\mathcal{F}_k$ and substitue the definition for the actor updat. Since

$$\begin{aligned} &\mathbb{E}[\|\theta_{k+1} - \theta_k\|^2 | \mathcal{F}_k] \\ &= \mathbb{E}[\|\alpha_{1,k} \hat{\nabla} J_{\text{rep}}(\theta_k) + \alpha_{2,k} \hat{\nabla} J_{\text{ins}}(\theta_k)\|^2 | \mathcal{F}_k] \\ &\le 2(\mathbb{E}[\|\alpha_{1,k} \hat{\nabla} J_{\text{rep}}(\theta_k)\|^2 | \mathcal{F}_k] + \mathbb{E}[\|\alpha_{2,k} \hat{\nabla} J_{\text{ins}}(\theta_k)\|^2 | \mathcal{F}_k]) \\ &\le \frac{1}{2}(\alpha_{1,k}^2 \sigma_{\text{rep}}^2 + \alpha_{2,k}^2 \sigma_{\text{ins}}^2), \end{aligned} \tag{E.22}$$

we can get

$$\begin{aligned} \mathbb{E}[W_{k+1}|\mathcal{F}_k] \ge& W_k + \mathbb{E}[\theta_{k+1} - \theta_k | \mathcal{F}_k]^\top \nabla J_{\text{rep}}(\theta_k) \\ & + \mathbb{E}[\theta_{k+1} - \theta_k | \mathcal{F}_k]^\top \nabla J_{\text{ins}}(\theta_k). \end{aligned} \tag{E.23}$$

Plug in the linear parameterized Q-function to the actor update, we can get

$$\begin{aligned} \theta_{k+1} - \theta_k =& \alpha_{1,k} \xi_k^T \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi_\theta(a_{T_k}|s_{T_k}) \\ & + \alpha_{2,k} \frac{\pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})}{\pi_{\text{teacher}}} \xi_k^T \varphi(\widetilde{s}_{T_k}, \widetilde{a}_{T_k}) \nabla \log \pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k}). \end{aligned} \tag{E.24}$$

To simplify the notation, let's denote

$$Z_{\text{rep}}^k(\theta) = \alpha_{1,k}(\xi_k^T - \xi_*)\varphi(s_{T_k}, a_{T_k}) \nabla \log \pi_\theta(a_{T_k}|s_{T_k}),$$

$$Z_{\text{ins}}^k(\theta) = \alpha_{2,k}(\xi_k^T - \xi_*)\varphi(\widetilde{s}_{T_k}, \widetilde{a}_{T_k}) \nabla \log \pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k}).$$

Take expectation conditioned on the filtration and get

$$\begin{aligned} \mathbb{E}[\theta_{k+1} - \theta_k | \mathcal{F}_k] =& \mathbb{E}[Z_{\text{rep}}^k(\theta)|\mathcal{F}_k] + \alpha_{1,k} \nabla J_{\text{rep}}(\theta_k) \\ & + \mathbb{E}[Z_{\text{ins}}^k(\theta, k) \frac{\pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})}{\pi_{\text{teacher}}} | \mathcal{F}_k] + \alpha_{2,k} \nabla J_{\text{ins}}(\theta_k) \end{aligned} \tag{E.25}$$

Then on both sides, take the inner product with $\nabla J_{\text{rep}}(\theta_k)$:

$$\begin{aligned} \mathbb{E}[\theta_{k+1} - \theta_k | \mathcal{F}_k]^\top \nabla J_{\text{rep}}(\theta_k) =& \mathbb{E}[Z_{\text{rep}}^k(\theta)|\mathcal{F}_k]^\top \nabla J_{\text{rep}}(\theta_k) \\ & + \alpha_{1,k} \|\nabla J_{\text{rep}}(\theta_k)\|^2 \\ & + \mathbb{E}[Z_{\text{ins}}^k(\theta) \frac{\pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})}{\pi_{\text{teacher}}} | \mathcal{F}_k]^\top \nabla J_{\text{rep}}(\theta_k) \\ & + \alpha_{2,k} \nabla J_{\text{ins}}(\theta_k)^\top \nabla J_{\text{rep}}(\theta_k) \\ \ge& - |\mathbb{E}[Z_{\text{rep}}^k(\theta)|\mathcal{F}_k]^\top \nabla J_{\text{rep}}(\theta_k)| \\ & + \alpha_{1,k} \|\nabla J_{\text{rep}}(\theta_k)\|^2 \\ & - |\mathbb{E}[Z_{\text{ins}}^k(\theta) \frac{\pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})}{\pi_{\text{teacher}}} | \mathcal{F}_k]^\top \nabla J_{\text{rep}}(\theta_k)| \\ & + \alpha_{2,k} \nabla J_{\text{ins}}(\theta_k)^\top \nabla J_{\text{rep}}(\theta_k). \end{aligned} \tag{E.26}$$

By the assumptions, we can get following bounds.

$$\begin{aligned} &||\varphi(s_{T_k}, a_{T_k})|| \cdot ||\nabla \log \pi_\theta(a_{T_k}|s_{T_k})|| \cdot ||\nabla J_{\text{rep}}(\theta_k)|| \\ &\qquad \le C_{\text{student}} B_\Theta C_{\text{rep}}, \end{aligned} \tag{E.27}$$

$$||\frac{\pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})}{\pi_{\text{teacher}}}|| \cdot ||\varphi(\widetilde{s}_{T_k}, \widetilde{a}_{T_k})|| \cdot ||\nabla \log \pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})|| \cdot ||\nabla J_{\text{rep}}(\theta_k)|| \leq C_{\text{student}} B_\Theta C_{\text{rep}}/b_{\min}. \quad \text{(E.28)}$$

Therefore, replace the bounds and we can get

$$\begin{aligned}
&\mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k]^\top \nabla J_{\text{rep}}(\theta_k) \geq \\
&-C_{\text{student}} B_\Theta C_{\text{rep}} \alpha_{1,k} \mathbb{E}[||\xi_k - \xi_*|||\mathcal{F}_k] \\
&+\alpha_{1,k}||\nabla J_{\text{rep}}(\theta_k)||^2 \\
&-C_{\text{student}} B_\Theta \frac{C_{\text{rep}}}{b_{min}} \alpha_{2,k} \mathbb{E}[||\xi_k - \xi_*|||\mathcal{F}_k] \\
&+\alpha_{2,k} \nabla J_{\text{ins}}(\theta_k)^T \nabla J_{rep}(\theta_k). \quad \text{(E.29)}
\end{aligned}$$

Similarly, for the objectives corresponding to the instance transfer, we can get

$$\begin{aligned}
&\mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k]^T \nabla J_{\text{ins}}(\theta_k) \geq \\
&-C_{\text{student}} B_\Theta C_{\text{ins}} \alpha_{1,k} \mathbb{E}[||\xi_k - \xi_*|||\mathcal{F}_k] \\
&+\alpha_{1,k} \nabla J_{\text{rep}}(\theta_k)^\top \nabla J_{\text{ins}}(\theta_k) \\
&-C_{\text{student}} B_\Theta \frac{C_{\text{ins}}}{b_{\min}} \alpha_{2,k} \mathbb{E}[||\xi_k - \xi_*|||\mathcal{F}_k] \\
&+\alpha_{2,k}||\nabla J_{\text{ins}}(\theta_k)||^2. \quad \text{(E.30)}
\end{aligned}$$

Now we add them together. Let $C = C_{\text{rep}} + C_{\text{ins}}$, then

$$\begin{aligned}
&\mathbb{E}[W_{k+1}|\mathcal{F}_k] \geq W_k \\
&- CC_{\text{student}} B_\Theta (\alpha_{1,k} + \frac{\alpha_{2,k}}{b_{\min}}) \mathbb{E}[||\xi_k - \xi_*|||\mathcal{F}_k] \\
&+ \alpha_{1,k}||\nabla J_{\text{rep}}(\theta_k)||^2 + \alpha_{2,k}||\nabla J_{\text{ins}}(\theta_k)||^2 \\
&+ (\alpha_{1,k} + \alpha_{2,k}) \nabla J_{\text{rep}}(\theta_k)^\top \nabla J_{\text{ins}}(\theta_k). \quad \text{(E.31)}
\end{aligned}$$

□

We now present the main result which is the convergence rate of Q-value-based REPAINT. Let K_ϵ be the smallest number of updates k required to attain a function gradient smaller than ϵ,

$$K_\epsilon = \min\{k : \inf_{0 \leq m \leq k} \mathcal{F}(\theta_m) < \epsilon\}, \quad \text{(E.32)}$$

where $A_k = \alpha_{2,k}/\alpha_{1,k}$ and

$$\begin{aligned}
\mathcal{F}(\theta_m) = &||\nabla J_{\text{rep}}(\theta_m)||^2 + A||\nabla J_{\text{ins}}(\theta_m)||^2 \\
&+ (1 + A_k) \nabla J_{\text{rep}}(\theta_m)^\top \nabla J_{\text{ins}}(\theta_m). \quad \text{(E.33)}
\end{aligned}$$

Theorem E.10. Suppose the representation transfer step size satisfies $\alpha_{1,k} = k^{-a}$ for $a > 0$ and the critic update satisfies Assumption E.5. The instance transfer step size satisfies $\alpha_{2,k} = A_k \alpha_{1,k}$ for $A_k \in \mathbb{R}^+$. When the critic bias converges to null as $\mathcal{O}(k^{-1})$ ($b = 1$), then $T_C(k) = k + 1$ critic updates occur per actor update. Alternatively, if the critic bias converges to null more slowly as $\mathcal{O}(k^{-b})$ with $b \in (0, 1)$ in Assumption E.5, then $T_C(k) = k$ critic updates per actor update are chosen. Then the actor sequence defined in Algorithm 1 satisfies

$$K_\epsilon \leq \mathcal{O}(\epsilon^{-1/l}), \quad \text{(E.34)}$$

where $l = \min\{a, 1 - a, b\}$. Moreover, minimizing over a, the resulting sample complexity depends on the attenuation b of the critic bias as

$$K_\epsilon \leq \begin{cases} \mathcal{O}(\epsilon^{-1/b}) & b \in (0, 1/2) \\ \mathcal{O}(\epsilon^{-2}) & b \in (1/2, 1] \end{cases} \quad \text{(E.35)}$$

Proof. Substitute for W_k in Lemma E.9,

$$\begin{aligned}
&\mathbb{E}[J_{\text{rep}}(\theta_{k+1})|\mathcal{F}_k] + \mathbb{E}[J_{\text{ins}}(\theta_{k+1})|\mathcal{F}_k] \\
&\quad - L(\sigma^2_{\text{rep}} \sum_{j=k+1}^{\infty} \alpha^2_{1,j} + \sigma^2_{\text{ins}} \sum_{j=k+1}^{\infty} \alpha^2_{2,j}) \\
&\quad \geq J_{\text{rep}}(\theta_k) + J_{\text{ins}}(\theta_k) \\
&\quad - L(\sigma^2_{\text{rep}} \sum_{j=k}^{\infty} \alpha^2_{1,j} + \sigma^2_{\text{ins}} \sum_{j=k}^{\infty} \alpha^2_{2,j}) \\
&- CC_{\text{student}} B_\Theta (\alpha_{1,k} + \frac{\alpha_{2,k}}{b_{\min}}) \mathbb{E}[||\xi_k - \xi_*|||\mathcal{F}_k] \\
&+ \alpha_{1,k}||\nabla J_{\text{rep}}(\theta_k)||^2 + \alpha_{2,k}||\nabla J_{\text{ins}}(\theta_k)||^2 \\
&+ (\alpha_{1,k} + \alpha_{2,k}) \nabla J_{\text{rep}}(\theta_k)^\top \nabla J_{\text{ins}}(\theta_k). \quad \text{(E.36)}
\end{aligned}$$

Cancel some common terms from both sides and take the total expectation, we can get

$$\begin{aligned}
&\mathbb{E}[J_{\text{rep}}(\theta_{k+1})] + \mathbb{E}[J_{\text{ins}}(\theta_{k+1})] \\
&\quad \geq \mathbb{E}[J_{\text{rep}}(\theta_k)] + \mathbb{E}[J_{\text{ins}}(\theta_k)] \\
&\quad - L(\sigma^2_{\text{rep}} \alpha^2_{1,k} + \sigma^2_{\text{ins}} \alpha^2_{2,k}) \\
&- CC_{\text{student}} B_\Theta (\alpha_{1,k} + \frac{\alpha_{2,k}}{b_{\min}}) \mathbb{E}[||\xi_k - \xi_*||] \\
&+ \mathbb{E}[\alpha_{1,k}||\nabla J_{\text{rep}}(\theta_k)||^2 + \alpha_{2,k}||\nabla J_{\text{ins}}(\theta_k)||^2 \\
&+ (\alpha_{1,k} + \alpha_{2,k}) \nabla J_{\text{rep}}(\theta_k)^\top \nabla J_{\text{ins}}(\theta_k). \quad \text{(E.37)}
\end{aligned}$$

Rearrange the terms and get

$$\begin{aligned}
&\mathbb{E}[\alpha_{1,k}||\nabla J_{\text{rep}}(\theta_k)||^2 + \alpha_{2,k}||\nabla J_{\text{ins}}(\theta_k)||^2 \\
&\quad + (\alpha_{1,k} + \alpha_{2,k}) \nabla J_{\text{rep}}(\theta_k)^\top \nabla J_{\text{ins}}(\theta_k)] \\
&\quad \leq \mathbb{E}[J_{\text{rep}}(\theta_{k+1})] - \mathbb{E}[J_{\text{rep}}(\theta_k)] \\
&\quad + \mathbb{E}[J_{\text{ins}}(\theta_{k+1})] - \mathbb{E}[J_{\text{ins}}(\theta_k)] \\
&+ CC_{\text{student}} B_\Theta (\alpha_{1,k} + \frac{\alpha_{2,k}}{b_{\min}}) \mathbb{E}[||\xi_k - \xi_*||] \\
&\quad + L(\sigma^2_{\text{rep}} \alpha^2_{1,k} + \sigma^2_{\text{ins}} \alpha^2_{2,k}). \quad \text{(E.38)}
\end{aligned}$$

Denote by LHS and RHS the left hand side and right hand side of the above equation. Define $U_k = J(\theta^*) - J(\theta_k)$ for

both rep and ins where θ^* is the optimal parameters. Then

$$\begin{aligned}\text{RHS} = \mathbb{E}[U_{k,\text{rep}}] - \mathbb{E}[U_{k+1,\text{rep}}] \\ + \mathbb{E}[U_{k,\text{ins}}] - \mathbb{E}[U_{k+1,\text{ins}}] \\ + CC_{\text{student}}B_\Theta(\alpha_{1,k} + \frac{\alpha_{2,k}}{b_{\min}})\mathbb{E}[||\xi_k - \xi_*||] \\ + L(\sigma^2_{\text{rep}}\alpha^2_{1,k} + \sigma^2_{\text{ins}}\alpha^2_{2,k}). \quad \text{(E.39)}\end{aligned}$$

Let $A_k\alpha_{2,k}/\alpha_{1,k}$, then

$$\begin{aligned}\text{LHS} = \alpha_{1,k}\mathbb{E}[||\nabla J_{\text{rep}}(\theta_k)||^2 + A_k||\nabla J_{\text{ins}}(\theta_k)||^2 \\ + (1 + A_k)\nabla J^T_{\text{rep}}(\theta_k)\nabla J_{\text{ins}}(\theta_k)], \quad \text{(E.40)}\end{aligned}$$

and

$$\begin{aligned}\text{RHS} = \mathbb{E}[U_{k,\text{rep}}] - \mathbb{E}[U_{k+1,\text{rep}}] + \\ \mathbb{E}[U_{k,\text{ins}}] - \mathbb{E}[U_{k+1,\text{ins}}] + \\ + CC_{\text{student}}B_\Theta(\alpha_{1,k} + \frac{A_k\alpha_{1,k}}{b_{\min}})\mathbb{E}[||\xi_k - \xi_*||] \\ + L(\sigma^2_{\text{rep}}\alpha^2_{1,k} + \sigma^2_{\text{ins}}A^2_k\alpha^2_{1,k}). \quad \text{(E.41)}\end{aligned}$$

Divide both sides by $\alpha_{1,k}$ and take the sum over $\{k - N, \ldots, k\}$ for some integer $1 < N < k$. Then we have

$$\begin{aligned}\text{newLHS} = \sum_{j=k-N}^{k} \mathbb{E}\left[||\nabla J_{\text{rep}}(\theta_j)||^2 + A_k||\nabla J_{\text{ins}}(\theta_j)||^2 \right. \\ \left. +(1 + A_k)\nabla J_{\text{rep}}(\theta_j)^\top \nabla J_{\text{ins}}(\theta_j)\right], \quad \text{(E.42)}\end{aligned}$$

and

$$\begin{aligned}\text{newRHS} = \sum_{j=k-N}^{k} \frac{1}{\alpha_{1,j}}(\mathbb{E}[U_{j,\text{rep}}] - \mathbb{E}[U_{j+1,\text{rep}}]) \\ + \sum_{j=k-N}^{k} \frac{1}{\alpha_{1,j}}(\mathbb{E}[U_{j,\text{ins}}] - \mathbb{E}[U_{j+1,\text{ins}}]) \\ + CC_{\text{student}}B_\Theta(1 + \frac{A_k}{b_{\min}}) \sum_{j=k-N}^{k} \mathbb{E}[||\xi_j - \xi_*||] \\ + L(\sigma^2_{\text{rep}} + \sigma^2_{\text{ins}}A^2_k) \sum_{j=k-N}^{k} \alpha_{1,j}. \quad \text{(E.43)}\end{aligned}$$

Rearrange the first two terms and get

$$\begin{aligned}\text{newRHS} = \sum_{j=k-N}^{k} (\frac{1}{\alpha_{1,j}} - \frac{1}{\alpha_{1,j-1}})\mathbb{E}[U_{j,\text{rep}}] \\ - \frac{1}{\alpha_{1,k}}\mathbb{E}[U_{k+1,\text{rep}}] + \frac{1}{\alpha_{1,k-N-1}}\mathbb{E}[U_{k-N,\text{rep}}] \\ + \sum_{j=k-N}^{k} (\frac{1}{\alpha_{1,j}} - \frac{1}{\alpha_{1,j-1}})\mathbb{E}[U_{j,\text{ins}}] \\ - \frac{1}{\alpha_{1,k}}\mathbb{E}[U_{k+1,\text{ins}}] + \frac{1}{\alpha_{1,k-N-1}}\mathbb{E}[U_{k-N,\text{ins}}] \\ + CC_{\text{student}}B_\Theta(1 + \frac{A_k}{b_{\min}}) \sum_{j=k-N}^{k} \mathbb{E}[||\xi_j - \xi_*||] \\ + L(\sigma^2_{\text{rep}} + \sigma^2_{\text{ins}}A^2_k) \sum_{j=k-N}^{k} \alpha_{1,j}. \quad \text{(E.44)}\end{aligned}$$

Since $\mathbb{E}[U_{k+1,\text{rep}}] \geq 0$ and $\mathbb{E}[U_{k+1,\text{ins}}] \geq 0$, we have

$$\begin{aligned}\text{newRHS} \leq \sum_{j=k-N}^{k} (\frac{1}{\alpha_{1,j}} - \frac{1}{\alpha_{1,j-1}})\mathbb{E}[U_{j,\text{rep}}] \\ + \frac{1}{\alpha_{1,k-N-1}}\mathbb{E}[U_{k-N,\text{rep}}] \\ + \sum_{j=k-N}^{k} (\frac{1}{\alpha_{1,j}} - \frac{1}{\alpha_{1,j-1}})\mathbb{E}[U_{j,\text{ins}}] \\ + \frac{1}{\alpha_{1,k-N-1}}\mathbb{E}[U_{k-N,\text{ins}}] \\ + CC_{\text{student}}B_\Theta(1 + \frac{A_k}{b_{\min}}) \sum_{j=k-N}^{k} \mathbb{E}[||\xi_j - \xi_*||] \\ + L(\sigma^2_{\text{rep}} + \sigma^2_{\text{ins}}A^2_k) \sum_{j=k-N}^{k} \alpha_{1,j}. \quad \text{(E.45)}\end{aligned}$$

Since we have rewards and score functions bounded above, we can find two positive constants D_{rep} and D_{ins}, such that $U_{k,\text{rep}} \leq D_{\text{rep}}$ and $U_{k,\text{ins}} \leq D_{\text{ins}}$ for all k. Substituting for

these bounds and get

$$\begin{aligned}\text{newRHS} \leq & \sum_{j=k-N}^{k}(\frac{1}{\alpha_{1,j}}-\frac{1}{\alpha_{1,j-1}})D_{\text{rep}}+\frac{1}{\alpha_{1,k-N-1}}D_{\text{rep}} \\ & + \sum_{j=k-N}^{k}(\frac{1}{\alpha_{1,j}}-\frac{1}{\alpha_{1,j-1}})D_{\text{ins}}+\frac{1}{\alpha_{1,k-N-1}}D_{\text{ins}} \\ & + CC_{\text{student}}B_{\Theta}(1+\frac{A_k}{b_{\min}})\sum_{j=k-N}^{k}\mathbb{E}[||\xi_j-\xi_*||] \\ & + L(\sigma_{\text{rep}}^2+\sigma_{\text{ins}}^2A_k^2)\sum_{j=k-N}^{k}\alpha_{1,j}. \quad \text{(E.46)}\end{aligned}$$

Unravel the telescoping sums:

$$\begin{aligned}\text{newRHS} \leq & \frac{D_{\text{rep}}+D_{\text{ins}}}{\alpha_{1,k}} \\ & + CC_{\text{student}}B_{\Theta}(1+\frac{A_k}{b_{\min}})\sum_{j=k-N}^{k}\mathbb{E}[||\xi_j-\xi_*||] \\ & + L(\sigma_{\text{rep}}^2+\sigma_{\text{ins}}^2A_k^2)\sum_{j=k-N}^{k}\alpha_{1,j}. \quad \text{(E.47)}\end{aligned}$$

Plug in the assumption that $\alpha_{1,k}=k^{-a}$, and the convergence rate of the critic (replace ξ_k by $\xi_{T_C(k)}$):

$$\begin{aligned}\text{newRHS} \leq & (D_{\text{rep}}+D_{\text{ins}})k^a \\ & + CC_{\text{student}}B_{\Theta}(1+\frac{A_k}{b_{\min}})\sum_{j=k-N}^{k}L_1T_C(j)^{-b} \\ & + L(\sigma_{\text{rep}}^2+\sigma_{\text{ins}}^2A_k^2)\sum_{j=k-N}^{k}j^{-a}. \quad \text{(E.48)}\end{aligned}$$

Let

$$\begin{aligned}\mathcal{F}(\theta_j) := & ||\nabla J_{\text{rep}}(\theta_j)||^2+A_k||\nabla J_{\text{ins}}(\theta_j)||^2 \\ & + (1+A_k)\nabla J_{\text{rep}}(\theta_j)^{\top}\nabla J_{\text{ins}}(\theta_j). \quad \text{(E.49)}\end{aligned}$$

When $b\in(0,1)$, we set $T_C(k)=k$. Since $\sum_{j=k-N}^{k}j^{-a}\leq(k^{1-a}-(k-N-1)^{1-a})/(1-a)$, we have

$$\begin{aligned}\text{newLHS} \leq & (D_{\text{rep}}+D_{\text{ins}})k^a \\ & + CC_{\text{student}}B_{\Theta}(1+\frac{A_k}{b_{\min}})\frac{L_1}{1-b}(k^{1-b}-(k-N-1)^{1-b}) \\ & + (\sigma_{\text{rep}}^2+\sigma_{\text{ins}}^2A_k^2)\frac{L}{1-a}(k^{1-a}-(k-N-1)^{1-a}). \quad \text{(E.50)}\end{aligned}$$

Divide both sides by k and set $N=k-1$,

$$\begin{aligned}\frac{1}{k}\sum_{j=1}^{k}\mathbb{E}[\mathcal{F}(\theta_j)] \leq & (D_{\text{rep}}+D_{\text{ins}})k^{a-1} \\ & + \frac{L(\sigma_{\text{rep}}^2+\sigma_{\text{ins}}^2A_k^2)}{1-a}k^{-a} \\ & + CC_{\text{student}}B_{\Theta}(1+\frac{A_k}{b_{\min}})\frac{L_1}{1-b}k^{-b}. \quad \text{(E.51)}\end{aligned}$$

By definition of K_ϵ we have that $\mathbb{E}[\mathcal{F}(\theta_j)]>\epsilon$ for $j=1,\ldots,K_\epsilon$ so

$$\epsilon \leq \frac{1}{K_\epsilon}\sum_{j=1}^{K_\epsilon}\mathbb{E}[\mathcal{F}(\theta_j)] \leq \mathcal{O}(K_\epsilon^{a-1}+K_\epsilon^{-a}+K_\epsilon^{-b}) \quad \text{(E.52)}$$

Defining $l=\min\{a,1-a,b\}$ and inverting, we have

$$K_\epsilon \leq \mathcal{O}(\epsilon^{-1/l}) \quad \text{(E.53)}$$

When $b=1$, we set $T_C(k)=k+1$. Similarly, we can get

$$\begin{aligned}\text{newLHS} \leq & (D_{\text{rep}}+D_{\text{ins}})k^a \\ & + CC_{\text{student}}B_{\Theta}(1+\frac{A_k}{b_{\min}})L_1(\log(k+1)-\log(k-N)) \\ & + (\sigma_{\text{rep}}^2+\sigma_{\text{ins}}^2A_k^2)\frac{L}{1-a}(k^{1-a}-(k-N-1)^{1-a}). \quad \text{(E.54)}\end{aligned}$$

Divide both sides by k and set $N=k-1$,

$$\begin{aligned}\frac{1}{k}\sum_{j=1}^{k}\mathbb{E}[\mathcal{F}(\theta_j)] \leq & (D_{\text{rep}}+D_{\text{ins}})k^{a-1} \\ & + \frac{L(\sigma_{\text{rep}}^2+\sigma_{\text{ins}}^2A_k^2)}{1-a}k^{-a} \\ & + CC_{\text{student}}B_{\Theta}(1+\frac{A_k}{b_{\min}})L_1\frac{\log(k+1)}{k}. \quad \text{(E.55)}\end{aligned}$$

Again, we can get

$$\epsilon \leq \frac{1}{K_\epsilon}\sum_{j=1}^{K_\epsilon}\mathbb{E}[\mathcal{F}(\theta_j)] \leq \mathcal{O}(K_\epsilon^{a-1}+K_\epsilon^{-a}+\frac{\log(K_\epsilon+1)}{K_\epsilon}). \quad \text{(E.56)}$$

Optimizing over a, we can get $\epsilon\leq\mathcal{O}(K_\epsilon^{-\frac{1}{2}})$ when $b>\frac{1}{2}$, and $\epsilon\leq\mathcal{O}(K_\epsilon^{-b})$ when $b\leq\frac{1}{2}$. □

E.1. Extension: Adding the Cross-entropy Term

In this section, we want to demonstrate that when the auxiliary cross-entropy is involved in J_{rep}, our results do not change.

By definition:

$$\nabla J_{\text{aux}}(\theta) = \nabla H(\pi_{\text{teacher}}||\pi_\theta) = -\mathbb{E}_{\pi_{\text{teacher}}}[\nabla_\theta \log \pi_\theta(a|s)]$$
$$= -\mathbb{E}_{\pi_\theta}\left[\frac{\pi_{\text{teacher}}(a|s)}{\pi_\theta(a|s)}\nabla_\theta \log \pi_\theta(a|s)\right]. \quad \text{(E.57)}$$

We now make an extra assumption on the student policy π_θ.

Assumption E.11. The student policy has a minimum positive value $d_{\text{min}} \in (0, 1] : \pi_\theta(a|s) \geq d_{\text{min}}$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$.

We now have an unbiased estimate for ∇J_{rep} with cross-entropy regularization

$$\hat{\nabla} J_{\text{rep}}(\theta) = \hat{Q}^{\pi_\theta}(s_T, a_T)\nabla \log \pi_\theta(a_T|s_T)$$
$$+ \beta_k \frac{\pi_{\text{teacher}}(a_T|s_T)}{\pi_\theta(a_T|s_T)}\nabla_\theta \log \pi_\theta(a_T|s_T). \quad \text{(E.58)}$$

where s_T, a_T is the state-action pair collected following the student policy.

Then the actor update obeys

$$\theta_{k+1} - \theta_k = \alpha_{1,k}\hat{Q}^{\pi_\theta}(s_{T_k}, a_{T_k})\nabla \log \pi_\theta(a_{T_k}|s_{T_k})$$
$$+\alpha_{1,k}\beta_k \frac{\pi_{\text{teacher}}(a_{T_k}|s_{T_k})}{\pi_\theta(a_{T_k}|s_{T_k})}\nabla_\theta \log \pi_\theta(a_{T_k}|s_{T_k})$$
$$+\alpha_{2,k}\frac{\pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})}{\pi_{\text{teacher}}(\widetilde{a}_{T_k}|\widetilde{s}_{T_k})}\hat{Q}_{\pi_\theta}(\widetilde{s}_{T_k}, \widetilde{a}_{T_k})\nabla \log \pi_\theta(\widetilde{a}_{T_k}|\widetilde{s}_{T_k}). \quad \text{(E.59)}$$

This results in having an additional term in $Z^k_{\text{rep}}(\theta)$ in the proof of Lemma E.9.

$$Z^k_{\text{rep}}(\theta) = \alpha_{1,k}(\xi^T_k - \xi_*)\varphi(s_{T_k}, a_{T_k})\nabla \log \pi_\theta(a_{T_k}|s_{T_k})$$
$$+ \alpha_{1,k}\beta_k \frac{\pi_{\text{teacher}}(a_{T_k}|s_{T_k})}{\pi_\theta(a_{T_k}|s_{T_k})}\nabla_\theta \log \pi_\theta(a_{T_k}|s_{T_k}). \quad \text{(E.60)}$$

Since

$$||\frac{\pi_{\text{teacher}}}{\pi_\theta}||.||\nabla \log \pi_\theta(a_{T_k}|s_{T_k})|| \leq \frac{B_\Theta}{d_{\text{min}}}, \quad \text{(E.61)}$$

it will only change the parameter in the proof of Lemma E.9 but not affect the final conclusion.