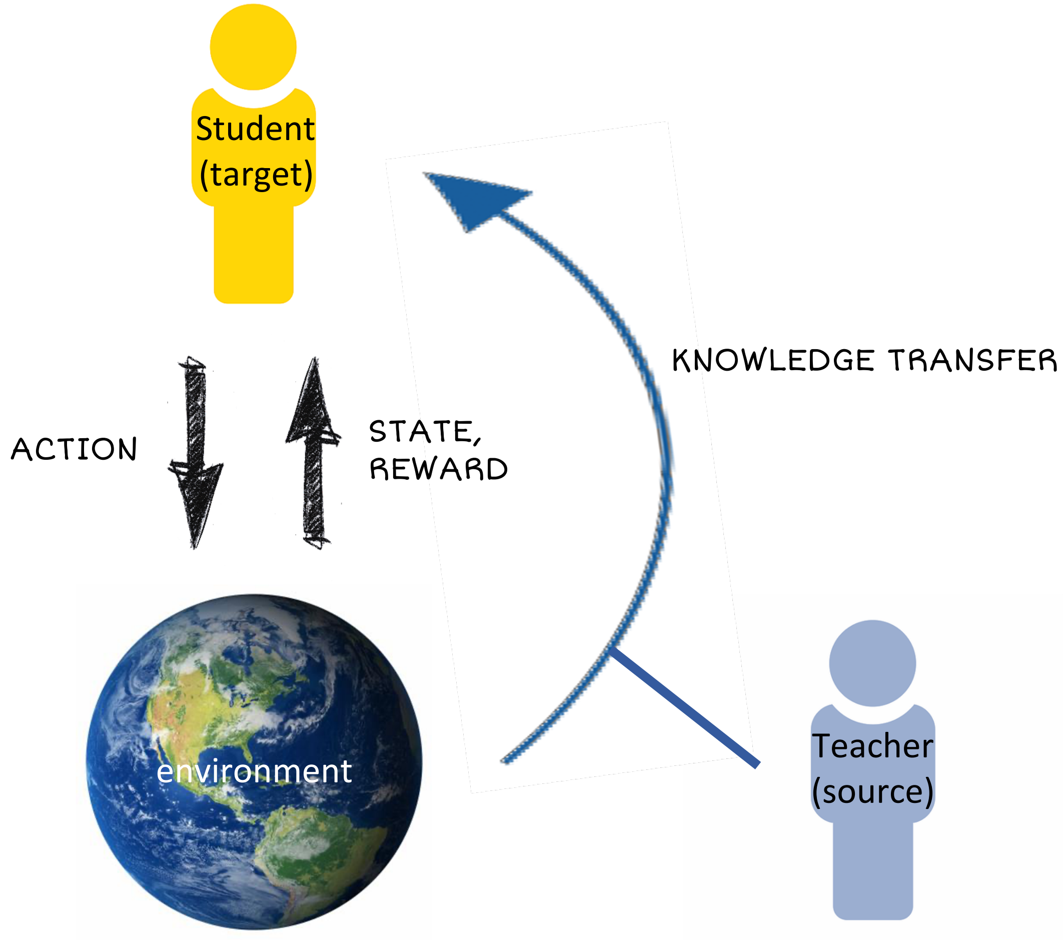
REPAINT: Knowledge Transfer in Deep Reinforcement Learning

Yunzhe Tao, Sahika Genc, Jonathan Chung, Tao Sun, Sunil Mallya

Amazon Web Services

Transfer Learning in RL



Parameter transfer
- [Taylor ’08; Mehta ’08; Rajendran ’15; Gupta ’17; etc.]

Representation transfer
- [Konidaris ’12; Parisotto ’15; Schaul ’15; Duan ’16; Yin ’17; Borsa ’18; Zhang ’18; Schmitt ’18; Ma ’18; Barreto ’19; etc.]

Instance transfer
- [Lazaric ’08; Taylor ’08; Tirinzoni ’18; etc.]

Our paper:

REPresentation **A**nd **IN**stance **T**ransfer (**REPAINT**)

- On-policy representation transfer
- Off-policy instance transfer
- Handles generic cases of source/target task similarity

On-policy Representation Transfer

- Kickstarting Deep RL [Schmitt et al. 2018]
- Allow a student network (target policy) to exploit access to expert teachers:

$$L_{\mathrm{aux}}(\theta) = H(\pi_{\mathrm{teacher}}(a|s)||\pi_\theta(a|s))$$

cross-entropy

- The objective (k is the iteration number):

$$L_{\mathrm{rep}}^{k}(\theta) = L_{\mathrm{RL}}(\theta) - \beta_k L_{\mathrm{aux}}(\theta)$$

RL objective

vanishing as k increases

Off-policy Instance Transfer

- Policy distillation works well only when source/target tasks are *similar*
- Idea: select "*good*" samples and update policy using those "*good*" samples
- "good" ≈ high **advantage** estimates

$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$

- Advantage-based experience selection:
  1. Collect trajectories $\{s_i, a_i, s_{i+1}\}$ following **teacher policy** $\boldsymbol{\pi}_{\text{teacher}}$
  2. Compute rewards using **current reward function**: $\{s_i, a_i, s_{i+1}, \tilde{r}_i\}$
  3. Compute advantage estimates $\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_T$
  4. $\boxed{\text{Remove } \hat{A}_t \text{ and } \{s_t, a_t, s_{t+1}, \tilde{r}_t\} \text{ if } \hat{A}_t < \zeta}$
  5. Update policy using selected samples

Alternative: select top X% samples

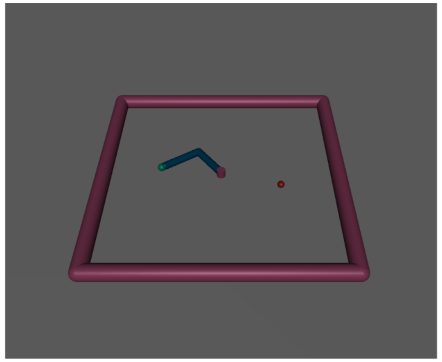
REPAINT Algorithm with Actor-Critic RL

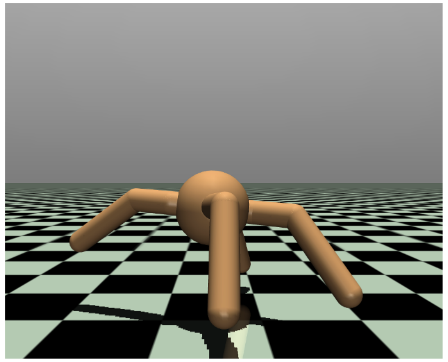
1. Collect $\mathcal{S} = \{s, a, s', r\}$ following $\pi_{\theta_{\text{old}}}(\cdot)$
2. Collect $\tilde{\mathcal{S}} = \{\tilde{s}, \tilde{a}, \widetilde{s'}, \tilde{r}\}$ following $\pi_{\text{teacher}}(\cdot)$
3. Update critic using $\mathcal{S}$
4. Perform advantage-based experience selection on $\tilde{\mathcal{S}}$
5. Update actor by:

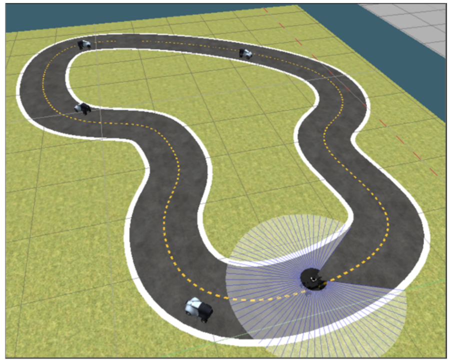
$$\theta \leftarrow \theta + \alpha_1 \nabla_\theta L_{\text{rep}}^k(\theta) + \alpha_2 \nabla_\theta L_{\text{ins}}(\theta)$$

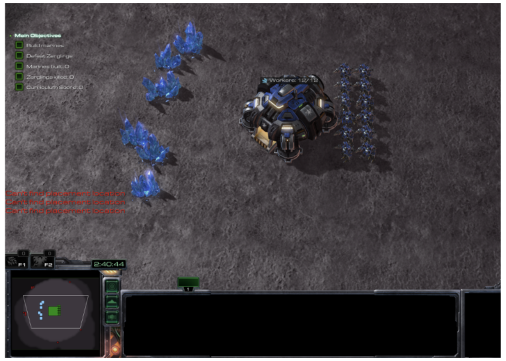
using $\mathcal{S}$ using $\tilde{\mathcal{S}}$ off-policy RL objective

Summary of Experimental Results









Env.	Teacher type	Target score	K_{Baseline}	K_{KS} (pct. reduced)	K_{IT} (pct. reduced)	K_{REPAINT} (pct. reduced)	Best scores KS	Best scores IT	Best scores REPAINT
Reacher	similar different	-7.4	173	51 (71%) 73 (58%)	97 (44%) 127 (27%)	42 (76%) 51 (71%)	-5.3 -6.9	-5.9 -6.4	-5.4 -5.2
Ant	similar	3685	997	363 (64%)	623 (38%)	334 (66%)	5464	5172	5540
Single-car	different different	394 345	18 22	Not achieved Not achieved	Not achieved Not achieved	13 (28%) 15 (32%)	331 300	388 319	396 354
Multi-car	sub-task diff/sub-task	1481 2.7	100 77	34 (66%) 66 (14%)	75 (25%) 53 (31%)	29 (71%) 25 (68%)	1542 4.9	1610 4.2	1623 6.1
StarCraft II	sub-task	112	95	92 (3%)	24 (75%)	6 (94%)	125	312	276

training time reduction