Replacing Rewards with Examples: Example-Based Policy Search via Recursive Classification

arxiv.org/pdf/2103.12656.pdf

tl;dr: transform example-based learning into

actor-critic style value-based learning

Disclaimer

- I have decided not to use any formulas in this presentation, otherwise it would get too involved. I ask you to trust my language ability when I summarize the paper

- But feel free to ask me further theoretical questions, since I think their “recursive classification” idea is novel

Background

- Q. Has this been attempted before?
- A. Yes! Behavioral cloning. Except BC still tries to learn a reward

- Q. What about goal-conditioned learning?
- A. More general, but based on C-Learning (Eysenbach et al 2021)

- Q. ValueDICE (Kostrikov et al 2019) and SQIL (Reddy et al 2020)
- A. Probabilistic formulation that does not rely on BC theory

Important Points

- 1. Only give success examples (i.e. states) and not trajectories (i.e. state-action pairs)
- 2. The algorithm learns a classifier that differentiates between "good" s-a pairs and s-a pairs from any "bad" random policy
- 3. Very similar to actor-critic methods, except the neural net learns to output probabilities rather than values
  - Specifically, I believe they directly modified the loss function of SAC
- 4. As a result, RCE satisfies (basically) the same Bellman Equations as reward-based updates

Algorithm

Algorithm 1 Recursive Classification of Examples

Input: success examples $\mathcal{S}^*$
Initialize policy $\pi_\phi(\mathbf{a_t} \mid \mathbf{s_t})$, classifier $C^\pi_\theta(\mathbf{s_t}, \mathbf{a_t})$,
  replay buffer $\mathcal{D}$
while not converged **do**
  Collect a new trajectory: $\mathcal{D} \leftarrow \mathcal{D} \cup \{\tau \sim \pi_\phi\}$
  Sample success examples: $\{\mathbf{s_t}^{(1)} \sim \mathcal{S}^*, \mathbf{a_t}^{(1)} \sim \pi_\phi(\mathbf{a_t}|\mathbf{s_t}^{(1)})\}$
  Sample transitions: $\{(\mathbf{s_t}^{(2)}, \mathbf{a_t}^{(2)}, \mathbf{s_{t+1}}) \sim \mathcal{D},$
    $\mathbf{a_{t+1}} \sim \pi_\phi(\mathbf{a_{t+1}} \mid \mathbf{s_{t+1}})\}$
  $w \leftarrow \frac{C^\pi_\theta(\mathbf{s_{t+1}}, \mathbf{a_{t+1}})}{1 - C^\pi_\theta(\mathbf{s_{t+1}}, \mathbf{a_{t+1}})}$ ▷ Eq. 9
  $\mathcal{L}(\theta) \leftarrow (1-\gamma)\mathcal{CE}(C_\theta(\mathbf{s_{t+1}}^{(1)}, \mathbf{a_t}^{(1)}); y = 1)$
    $(1+\gamma w)\mathcal{CE}(C_\theta(\mathbf{s_t}^{(2)}, \mathbf{a_t}^{(2)}); y = \frac{\gamma w}{1+\gamma w})$
  Update classifier: $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}(\theta)$ ▷ Eq. 8
  Update policy: $\phi \leftarrow \phi + \eta \nabla_\phi \mathbb{E}_{\pi_\phi(\mathbf{a_t}|\mathbf{s_t})}[C_\theta(\mathbf{s_t}, \mathbf{a_t})]$
return π_ϕ

Experiments (1)

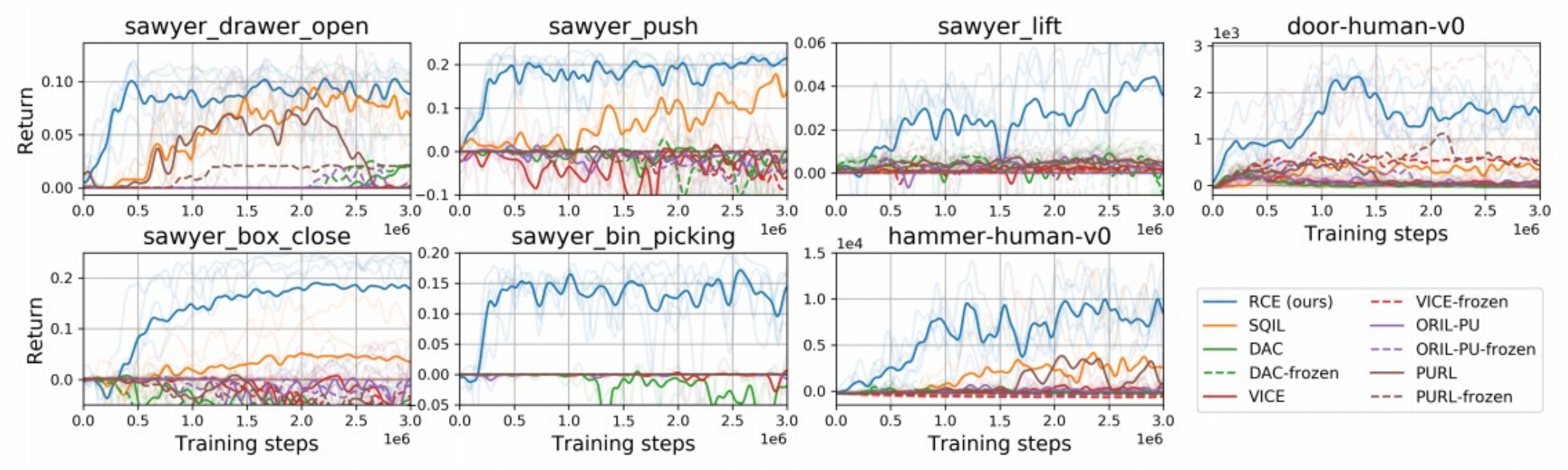

Figure 2. **Recursive Classification of Examples for learning manipulation tasks**: We apply RCE to a range of manipulation tasks, each accompanied with a dataset of success examples. For example, on the `sawyer_lift` task, we provide success examples where the object has been lifted above the table. We use the cumulative task return (↑ is better) solely for evaluation. We observe that our method (blue line) outperforms prior methods across all tasks.

Experiments (2)

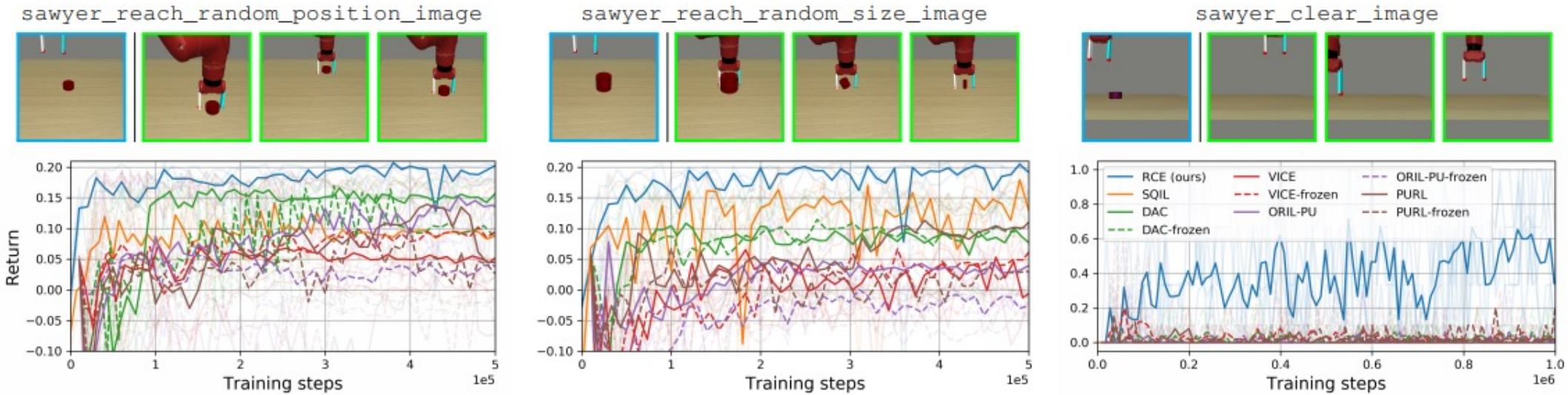

Figure 4. **Example-based control from images**: We evaluate RCE on three manipulation tasks using image-observations. *(Top)* We show examples of the initial state and success examples for each task. *(Bottom)* RCE (blue line) outperforms prior methods, especially on the more challenging clearing task. For the `random_size` task (*center*), this entails reaching for new objects that have different sizes from any seen in the success examples.

Future Directions

- 1. The authors observed phenomenons very similar to Q-function overestimation in their experiments. Since the classifier shares certain commonality with value-based update, this is a promising future direction of research on improving RCE

- 2. Even though this paper talks about a very interesting assumption about how the success examples are collected, I feel like they did not exploit the theoretical properties enough.