Policy Learning Using Weak Supervision

Anonymous Author(s)
Affiliation
Address
email

Abstract

Most existing policy learning solutions require the learning agents to receive high-quality supervision signals, e.g., rewards in reinforcement learning (RL) or high-quality expert demonstrations in behavioral cloning (BC). These quality supervisions are either infeasible or prohibitively expensive to obtain in practice. We aim for a unified framework that leverages the available cheap weak supervisions to perform policy learning efficiently. To handle this problem, we treat the "weak supervision" as imperfect information coming from a *peer agent*, and evaluate the learning agent's policy based on a "correlated agreement" with the peer agent's policy (instead of simple agreements). Our approach explicitly punishes a policy for overfitting to the weak supervision. In addition to theoretical guarantees, extensive evaluations on tasks including RL with noisy reward, BC with weak demonstrations, and standard policy co-training (RL + BC) show that our method leads to substantial performance improvements, especially when the complexity or the noise of the learning environments is high.

1 Introduction

Recent breakthrough in policy learning (PL) opens up the possibility to apply these techniques in real-world applications such as robotics [1, 2] and self-driving [3, 4]. Nonetheless, most existing works require agents to receive high-quality supervision signals, e.g., reward or expert demonstrations, which are either infeasible or prohibitively expensive to obtain in practice. The reward may be collected through faulty sensors thus not credible [5, 6, 7]. The demonstrations by an expert in behavioral cloning (BC) are often imperfect due to limited resources and environment noise [8, 9, 10].

Learning from weak supervision signals such as noisy rewards $\tilde{r}$ (noisy versions of r) [7] or low-quality demonstrations $\widetilde{\mathcal{D}}_{\mathrm{E}}$ (noisy versions of $\mathcal{D}_E$) produced by untrustworthy expert $\tilde{\pi}_E$ [9] is one of the outstanding challenges that prevents a wider application of PL. Although some recent works have explored these topics separately in their specific domains [11, 7, 12], there lacks unified solution for robust policy learning in this imperfect situation. We first formulate a meta-framework to study RL/BC with weak supervision signals and call it *weakly supervised policy learning*. Then as a response, we propose a theoretically principled solution concept, PeerPL, to perform efficient policy learning using the available weak supervision.

Our solution is inspired by the literature of information elicitation without verification [13, 14, 15], where the problem concerns evaluating self-reported information without ground truth verification. Instead, only a group of other agents' reports (for the same task, but none of them is assumed to be perfect) are available to serve the validation. We adopt a similar idea and treat the "weak supervision" as information coming from an imperfect *peer agent*, and evaluate the learning agent's policy based on a "correlated agreement" (CA) with the weak supervision signals. Compared to standard reward/loss functions that encourage simple agreements with the supervision, our approach punishes "over-agreement" to avoid overfitting to the weak supervision, which offers us a family of solutions that do not require prior knowledge of the weakness of the supervision.

We demonstrate how the proposed PeerPL framework adapts in challenging tasks including RL with noisy rewards and BC from weak demonstrations. We provide intensive analysis of the convergence behavior and the sample complexity for our solutions. These results jointly demonstrate that our approach enables agents to learn the optimal policy efficiently using only weak supervision. Experiment results show strong evidence that PeerPL brings significant improvements over state-of-the-art solutions, especially when the complexity or the noise of the learning environments is high.

To summarize, the contributions in the paper are mainly three-folds: (1) We provide a unified formulation of the *weakly supervised policy learning* problems; (2) We propose PeerPL, a new way to perform policy evaluation for RL/BC tasks; (3) PeerPL is theoretically guaranteed to recover the optimal policy, as if the supervision are of high-quality and clean. Competitive empirical performances are observed in several policy learning tasks.

1.1 Related Work

Learning with Noisy Supervision Learning from noisy supervision is a widely explored topic. The seminal work [16] first proposed an unbiased surrogate loss function to recover the true loss from the noisy label distribution, given the knowledge of the noise rates of labels. Follow-up works offered ways to estimate the noise rates [17, 18, 19, 20, 21, 22]. A recent work [7] adapts this idea to the RL setting and proposes a statistics-based estimation algorithm for the noise rate in observed rewards. Nonetheless, the estimation of noise rates is highly non-trivial and in-efficient, especially when the state-action space is huge. Further, as in a sequential process, the error in the estimation can accumulate and amplify when deploying an RL algorithm. In contrast, our solution does not require a priori specification of the noise rates, therefore offloading the burden of estimation.

Behavioral Cloning (BC) Standard BC [23, 24] tackles the sequential decision-making problem by imitating the expert actions using supervised learning. Specifically, it aims to minimize the one-step deviation error over the expert trajectory without reasoning the sequential consequences of actions. Therefore, the agent suffers from compounding errors when there is a mismatch between demonstrations and real states encountered [24, 25]. Recent works introduce data augmentations [26] and value-based regularization [27] or inverse dynamics models [28, 29] to encourage learning long-horizon behaviors. While being simple and straightforward, BC has been widely investigated in a range of application domains [30, 31] and often yields competitive performance [32, 27]. Our framework is complementary to the current BC literature by introducing a learning strategy from weak demonstrations (e.g., noisy or from a poorly-trained agent) and provides theoretical guarantees on how to retrieve clean policy under mild assumptions [33].

Correlated Agreement Information elicitation mechanisms aim to elicit information from self-interested agents without ground-truth verification [13, 14, 15]. The only source of information to serve as verification comes from the agents' unverified and imperfect reports. Particularly, in [14, 15], a correlated agreement (CA) type of mechanism is proposed. CA evaluates the correlations between agents' reports. But in addition to encouraging a certain agreement between agents' reports, CA also punishes over-agreement when two agents always report identically. This property helps reduce the effect of noisy reports by discouraging overfitting. Recently, [34] adapts a similar idea to the supervised learning setting. We consider a more challenging weakly supervised policy learning task and study the convergence rates in sequential decision-making problems.

2 Policy Learning from Weak Supervision

We begin by introducing a general framework to unify the problem of performing PL using weak supervision. Then we provide instantiations of the proposed weakly supervised formulation with two different applications: (1) RL with noisy reward and (2) behavioral cloning (BC) using weak expert demonstrations.

2.1 Preliminary of Policy Learning

The goal of policy learning (PL) is to learn a policy π that the agent could follow to perform a series of actions in a stateful environment. For RL, the interactive environment is characterized as an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$. At each time t, the agent in state $s_t \in \mathcal{S}$ takes an action $a_t \in \mathcal{A}$ by following the policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and *potentially* receives a reward $r(s_t, a_t) \in \mathcal{R}$. Then the agent transfers to the next state s_{t+1} according to a transition probability function $\mathcal{P}$. We denote the generated trajectory $\tau = \{(s_t, a_t, r_t)\}_{t=0}^{T}$, where T is a finite or infinite horizon. RL algorithms aim to maximize the expected reward over the trajectory τ induced by the policy: $\underline{J(\pi) = \mathbb{E}_{(s_t,a_t,r_t)\sim\tau}[\sum_{t=0}^{T} \gamma^t r_t]}$, where $\gamma \in (0, 1]$ is the discount factor.

Another popular policy learning method is behavioral cloning (BC). The goal of BC is to mimic the expert policy π_E through a set of demonstrations $D_E = \{(s_i, a_i)\}_{i=1}^{N}$ drawn from a distribution $\mathcal{D}_E$ (generated according to π_E), where (s_i, a_i) is the sampled state-action pair from the expert trajectory. Typically, training a policy with standard BC corresponds to maximizing the following log-likelihood: $\underline{J(\pi) = \mathbb{E}_{(s,a)\sim\mathcal{D}_E}\left[\log \pi(a|s)\right]}$.

In both RL and BC, the learning agent receives "supervision" through either the reward r by interacting with environments or the expert policy π_E as observable demonstrations. Consider a particular policy class Π, the optimal policy is then defined as $\underline{\pi^* = \arg\max_{\pi\in\Pi} J(\pi)}$: π^* obtains the maximum expected reward over the horizon T in RL and π^* corresponds to the clean expert policy π_E in BC. In practice, one can also combine both RL and BC approaches to take advantage of both worlds [35, 36, 11, 33]. Specifically, a recent hybrid framework called policy co-training [33] will be considered in this paper.

2.2 A Meta Framework for Policy Learning with Weak Supervision

With full supervision, both RL and BC can converge to the optimal policy π^*. However, when only weak supervision is available, with an over-parameterized model such as a deep neural network, the learning agent will easily memorize the weak supervision and learn a biased policy [37]. In our meta framework, instead of converging to any biased policy, we focus on learning the optimal policy π^* with only a weak supervision sequence denoted as $\{(s_i, a_i), \widetilde{Y}_i\}_{i=1}^{N}$. The *weak supervision signal* $\widetilde{Y}$ could be the reward $\tilde{r}$ for RL or the action $\tilde{a}$ performed by an expert policy $\tilde{\pi}_E$ for BC, which are noisy versions of the corresponding high-quality supervision signals. See more details below.

RL with Noisy Reward Consider a finite MDP $\widetilde{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, F, \mathcal{P}, \gamma \rangle$ with noisy reward channels [7], where $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and the noisy reward $\tilde{r}$ is generated following a certain function $F : \mathcal{R} \to \widetilde{\mathcal{R}}$. Denote the trajectory a policy π_θ generates via interacting with $\widetilde{\mathcal{M}}$ as $\tilde{\tau}_\theta$. Assume the reward is discrete and has $|\mathcal{R}|$ levels. The noisy reward can be characterized via a matrix $\mathbf{C}^{\text{RL}}_{|\mathcal{R}|\times|\mathcal{R}|}$, where each entry $c_{j,k}$ indicates the flipping probability for generating a possibly different outcome: $c^{\text{RL}}_{j,k} = \mathbb{P}\left(\tilde{r}_t = R_k | r_t = R_j\right)$. We call r and $\tilde{r}$ the *true reward* and *noisy reward* respectively.

BC with Weak Demonstration Instead of observing the true expert demonstration generated according to π_E, denote the available weak demonstrations by $\{(s_i, \tilde{a}_i)\}_{i=1}^{N}$, where $\tilde{a}_i \sim \tilde{\pi}_E(\cdot|s_i)$ is the noisy action and each state-action pair $(s_i, \tilde{a}_i)$ is drawn from distribution $\widetilde{\mathcal{D}}_E$. In particular, we assume the noisy action $\tilde{a}_i$ is independent of the state s given the deterministic expert action $\pi_E(s)$, i.e., $\mathbb{P}(\tilde{a}_i|\pi_E(s_i)) = \mathbb{P}(\tilde{a}_i|s_i, \pi_E(s_i))$. Similar to RL, we assume the noisy actions can be characterized by a confusion matrix $\mathbf{C}^{\text{BC}}_{|\mathcal{A}|\times|\mathcal{A}|}$, where each entry $c_{j,k}$ indicates the flipping probability for taking a sub-optimal action that differs from $\pi_E(s)$: $c^{\text{BC}}_{j,k} = \mathbb{P}(\tilde{\pi}_E(s) = A_k | \pi_E(s) = A_j)$. We aim to recover π^* as if we were able to access the quality expert demonstration π_E instead of $\tilde{\pi}_E$.

Knowledge of C We emphasize that our method makes PL robust to weak supervision automatically, which is free of any knowledge characterizing the signal weakness, i.e., $\mathbf{C}^{\text{RL}}_{|\mathcal{R}|\times|\mathcal{R}|}$ nor $\mathbf{C}^{\text{BC}}_{|\mathcal{A}|\times|\mathcal{A}|}$. Comparing with methods requiring knowledge of the confusion matrices [16, 7], we get rid of the accumulated error of an imperfect confusion matrix and make it easier to implement.

Evaluation Function We have an evaluation function $\mathsf{Eva}_\pi((s_i, a_i), \widetilde{Y}_i)$ which evaluates a taken policy at state (s_i, a_i) using the weak supervision $\widetilde{Y}_i$. In the RL setting, this Eva_π is the loss for different RL algorithms, which is a function of the noisy reward $\tilde{r}$ received at (s_i, a_i). While for the BC setting, this Eva_π is the loss used to evaluate the action taken by the agent given the action taken by the expert. Furthermore, we let $J(\pi)$ denote the function that evaluates policy π under a set of state action pairs with weak supervision signals $\{(s_i, a_i), \widetilde{Y}_i\}_{i=1}^{N}$, i.e., $J(\pi) = \mathbb{E}_{(s,a)\sim\tau}[\mathsf{Eva}_\pi((s, a), \tilde{Y})]$. The above unified notations are only for better delivery of our framework and we still treat PL as a sequential decision problem.

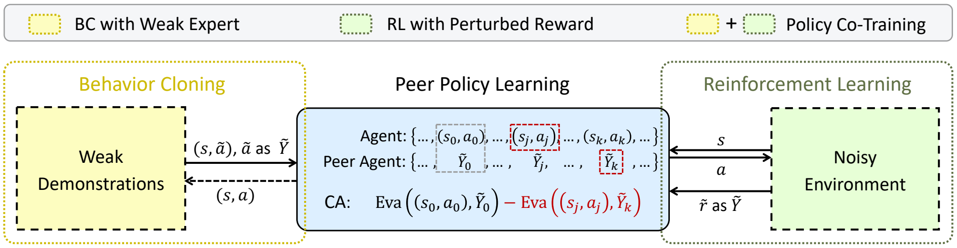

BC with Weak Expert
RL with Perturbed Reward
+ Policy Co-Training
Behavior Cloning
Peer Policy Learning
Reinforcement Learning
Weak Demonstrations
$(s,\tilde{a})$, $\tilde{a}$ as $\tilde{Y}$
(s,a)
Agent: $\{\dots,(s_0,a_0),\dots,(s_j,a_j),\dots,(s_k,a_k),\dots\}$
Peer Agent: $\{\dots,\tilde{Y}_0,\dots,\tilde{Y}_j,\dots,\tilde{Y}_k,\dots\}$
CA: $\mathrm{Eva}\big((s_0,a_0),\tilde{Y}_0\big)-\mathrm{Eva}\big((s_j,a_j),\tilde{Y}_k\big)$
s
a
$\tilde{r}$ as $\tilde{Y}$
Noisy Environment

Figure 1: Illustration of *weakly supervised policy learning* and our PeerPL solution with correlated agreement (CA). We use $\tilde{Y}$ to denote a weak supervision, be it a noisy reward, or a noisy demonstration. Eva stands for an evaluation function. "Peer Agent" corresponds to weak supervision.

3 PeerPL: Weakly Supervised PL via Correlated Agreement

To deal with weak supervision in PL, we propose a unified and theoretically principled framework PeerPL. We treat the weak supervision as information coming from a "peer agent", and then evaluate the policy using a certain type of "correlated agreement" function between the learning policy and the peer agent's information.

3.1 Overview of the Idea: Correlated Agreement with Weak supervision

We first present the general idea of our PeerPL framework using a concept named "correlated agreement" (CA). For each weak supervision $((s_i,a_i),\widetilde{Y}_i)$, we randomly (with replacement) sample two other instances indexed by j and k. Then we take the state-action pair (s_j,a_j) of sample j and the supervision signal $\widetilde{Y}_k$ of sample k, and evaluate $((s_i,a_i),\widetilde{Y}_i)$ according to the following rule:

$$\text{CA with Weak Supervision:}\quad \mathsf{Eva}_\pi\big((s_i,a_i),\widetilde{Y}_i\big)-\mathsf{Eva}_\pi\big((s_j,a_j),\widetilde{Y}_k\big).$$

This operation is illustrated in Figure 1. We further show intuitions and a toy example below.

Intuition The first term above encourages an "agreement" with the weak supervision (that a policy agrees with the corresponding supervision), while the second term punishes a "blind" and "over" agreement that happens when the agent's policy always matches with the weak supervision even on randomly paired traces (noise). The randomly paired instances j,k help us achieve this check. Note our mechanism does not require the knowledge of $\mathbf{C}^{\mathrm{RL}}_{|\mathcal{R}|\times|\mathcal{R}|}$ nor $\mathbf{C}^{\mathrm{BC}}_{|\mathcal{A}|\times|\mathcal{A}|}$, and offers a **prior-knowledge free** way to learn effectively with weak supervision.

Toy Example Consider a toy BC setting where the policy fully memorizes the weak supervision $a'_1=a'_2=a'_3=1, a'_4=0$ and outputs a sequence of actions $a_1=a_2=a_3=1, a_4=0$ at the same sequence of states. Let $\mathsf{Eva}_\pi((s_i,a_i),a'_i)=1$ if the policy agrees with the demonstration, otherwise $\mathsf{Eva}_\pi((s_i,a_i),a'_i)=0$. *Without CA:* We have $\mathbb{E}[\mathsf{Eva}_\pi((s_i,a_i),a'_i)]=1$ for agreeing with this noisy/imperfect/low-quality supervision. *With CA:* We have $\mathbb{E}[\mathsf{Eva}_\pi((s_i,a_i),a'_i)-\mathsf{Eva}_\pi((s_j,a_j),a'_k)]=1-(0.75^2+0.25^2)=0.375$, where $0.75^2+0.25^2$ is the probability of randomly paired a_j and a'_k matching each other. The above example shows that a full agreement with the weak supervision will instead be punished!

In what follows, we consolidate our implementations within each of the settings considered and provide theoretical guarantees under weak supervision.

3.2 PeerRL: Peer Reinforcement Learning

We propose the following objective function to punish the over-agreement based on CA:

$$J^{\mathrm{RL}}(\pi_\theta)=\mathbb{E}\Big[\mathsf{Eva}^{\mathrm{RL}}_\pi\big((s_i,a_i),\tilde{r}_i\big)\Big]-\xi\cdot\mathbb{E}\Big[\mathsf{Eva}^{\mathrm{RL}}_\pi\big((s_j,a_j),\tilde{r}_k\big)\Big], \tag{1}$$

$$\text{where}\quad \mathsf{Eva}^{\mathrm{RL}}_\pi\big((s,a),\tilde{r}\big)=-\ell\big(\pi_\theta,(s,a,\tilde{r})\big). \tag{2}$$

In (1), the first expectation is taken over $(s_i,a_i,\tilde{r}_i)\sim\tilde{\tau}_\theta$ and second one is taken over $(s_j,a_j,\tilde{r}_j)\sim\tilde{\tau}_\theta,(s_k,a_k,\tilde{r}_k)\sim\tilde{\tau}_\theta$, where $\tilde{\tau}$ is the trajectory specified by the noisy reward function $\tilde{r}$. Recall j,k

denote two randomly and independently sampled instances. Loss function ℓ depends on the employed RL algorithms, e.g., temporal difference error [38, 39] or the policy gradient loss [40]. The learning sequence is encoded in π. The objective $J^{\text{RL}}(\pi)$ represents the accumulated reward with agreement check. Parameter $\xi \geq 0$ balances the penalty for blind agreements induced by CA.

Peer RL In what follows, we consider the Q-Learning [41] as the underlying learning algorithm where $\ell(\pi_\theta, (s, a, \tilde{r})) = -\tilde{r}$ and demonstrate that the CA mechanism provides strong guarantees for Q-Learning with only observing the noisy reward. For clarity, we define *peer RL reward*:

$$\textsf{Peer RL Reward:} \quad \tilde{r}_{\text{peer}}(s, a) = \tilde{r}(s, a) - \xi \cdot \tilde{r}',$$

where $\tilde{r}' \overset{\pi_{\text{sample}}}{\sim} \{\tilde{r}(s, a) | s \in \mathcal{S}, a \in \mathcal{A}\}$ is a reward sampled over all state-action pairs according to a fixed policy π_{sample}. Note the sampling policy π_{sample} is independent of π and the choice of π_{sample} does not affect our theoretical results. We adopt a random sampling strategy in practice. Parameter $\xi \geq 0$ balances the noisy reward and the punishment for blind agreement (with $\tilde{r}'$). We set $\xi = 1$ (for binary case) in the following analysis and treat each (s, a) equally when sampling the r'. In experiments, we find $\tilde{r}_{\text{peer}}$ is not sensitive to the choice of ξ and keep ξ constant for each run.

Robustness to Noisy Rewards Now we show peer reward $\tilde{r}_{\text{peer}}$ offers us an affine transformation of the true reward in expectation, which is the key to guaranteeing the convergence of our Peer RL algorithm to converge to π^*. For clarity, consider the binary reward setting (r_+ and r_-) and denote the error in $\tilde{r}$ as $e_+ = \mathbb{P}(\tilde{r} = r_- | r = r_+), e_- = \mathbb{P}(\tilde{r} = r_+ | r = r_-)$ (a simplification of $\mathbf{C}^{\text{RL}}_{|\mathcal{R}| \times |\mathcal{R}|}$ in the binary setting).

Lemma 1. *Let $r \in [0, R_{\max}]$ be a bounded reward. Assume $1 - e_- - e_+ > 0$ then, we have:*

$$\mathbb{E}[\tilde{r}_{\text{peer}}] = (1 - e_- - e_+) \cdot \mathbb{E}[r_{\text{peer}}] = (1 - e_- - e_+) \cdot \mathbb{E}[r] + \textit{const}\,,$$

where $r_{peer} = r_{peer}(s, a) = r(s, a) - \xi \cdot r'$ is the peer RL reward when observing the true reward r.

Lemma 1 shows that by subtracting the peer penalty term $\tilde{r}'$ from noisy reward $\tilde{r}$, $\tilde{r}_{\text{peer}}$ recovers the clean and true reward r in expectation.

Remark It is notable that the expectation of the noisy reward $\mathbb{E}[\tilde{r}]$ writes as:

$$\mathbb{E}[\tilde{r}] = (1 - e_- - e_+)\mathbb{E}[r] + \underbrace{e_- r_+ + e_+ r_-}_{\text{const}}.$$

Nonetheless, the constant in peer reward has far less effect on the true reward r, especially when the noise rate is high. To see this:

$$\begin{aligned}\mathbb{E}[\tilde{r}] &= \eta \cdot \left(\mathbb{E}[r] + \frac{e_+}{1 - e_- - e_+} r_- + \frac{e_-}{1 - e_- - e_+} r_+ \right), \\ \mathbb{E}[\tilde{r}_{\text{peer}}] &= \eta \cdot (\mathbb{E}[r] - (1 - p_{\text{peer}}) r_- - p_{\text{peer}} r_+),\end{aligned}$$

where $\eta = 1 - e_- - e_+ > 0$, $p_{\text{peer}} \in [0, 1]$ denotes the probability that a sample policy sees a reward r_+ overall. Since the magnitude of noise terms $\frac{e_-}{1-e_- -e_+}$ and $\frac{e_+}{1-e_- -e_+}$ can potentially become much larger than $1 - p_{\text{peer}}$ and p_{peer} in a high-noise regime, $\frac{e_-}{1-e_- -e_+} r_+ + \frac{e_+}{1-e_- -e_+} r_-$ will dilute the informativeness of $\mathbb{E}[r]$. On the contrary, $\mathbb{E}[\tilde{r}_{\text{peer}}]$ contains a moderate constant noise thus maintaining more useful training signals of the true reward in practice.

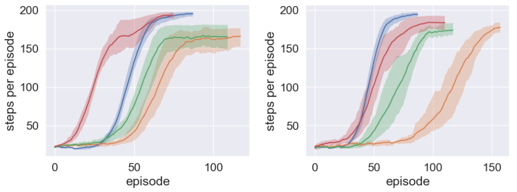
Based on Lemma 1, we further offer the following convergence guarantee:

Theorem 1. *(Convergence) Given a finite MDP with noisy reward, denoting as $\widetilde{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, F, \mathcal{P}, \gamma \rangle$, the Q-learning algorithm with peer rewards, given by:*

$$\begin{aligned} Q_{t+1}(s_t, a_t) &= (1 - \alpha_t) Q_t(s_t, a_t) + \alpha_t \left[\tilde{r}_{\text{peer}}(s_t, a_t) + \gamma \max_{a' \in \mathcal{A}} Q_t(s_{t+1}, a') \right], \\ \pi_t(s) &= \arg\max_{a \in \mathcal{A}} Q_t(s, a) \end{aligned}$$

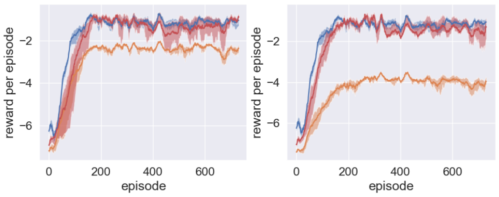
converges w.p.1 to the optimal policy $\pi^*(s)$ as long as $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$.

Theorem 1 states that the learning agent will converge to the optimal policy *w.p.1* with peer rewards without requiring any knowledge of the corruption in rewards ($\mathbf{C}^{\text{RL}}_{|\mathcal{R}| \times |\mathcal{R}|}$, as opposed to previous



(a) $e = 0.2$ (b) $e = 0.4$

Figure 2: Learning curves of DDQN on CartPole with true reward (r) 🟦, noisy reward ($\tilde{r}$) 🟧, surrogate reward [7] ($\hat{r}$) 🟩, and peer reward ($\tilde{r}_{\text{peer}}$, $\xi = 0.2$) 🟥.



(a) $e = 0.2$ (b) $e = 0.4$

Figure 3: Learning curves of DDPG [42] on Pendulum with true reward (r) 🟦, noisy reward ($\tilde{r}$) 🟧, and peer reward ($\tilde{r}_{\text{peer}}, \xi = 0.2$) 🟥. The performance of surrogate reward is omitted for clarity.

work [7] that requires such knowledge). Moreover, we found that to guarantee the convergence to π^*, the number of samples needed for our approach is no more than $\mathcal{O}(1/(1 - e_- - e_+)^2)$ times of the one needed when the RL agent observes true rewards perfectly (see Appendix A).

Extension Even though we only present an analysis for the binary case for Q-Learning, our approach is rather generic and is ready to be plugged into modern DRL algorithms. We provide **multi-reward extensions**, implementations with DQN [38] and policy gradient [40] in Appendix A.

3.3 PeerBC: Peer Behavioral Cloning

Similarly, we present our CA solution in the setting of behavioral cloning (PeerBC). In BC, the supervision is given by the weak expert's noisy trajectory. At each iteration, the agent learns under weak supervision $\tilde{a}$, and the training samples are generated from the distribution $\widetilde{\mathcal{D}}_E$ determined by the weak expert. The $\mathsf{Eva}_\pi^{\mathrm{BC}}$ function in BC evaluates the agent policy π_θ and the weak trajectory $\{(s_i, \tilde{a}_i)\}_{i=1}^N$ using $\ell(\pi_\theta, (s_i, \tilde{a}_i))$, where ℓ is an arbitrary classification loss. Taking the cross-entropy for instance, the objective of PeerBC is:

$$J^{\mathrm{BC}}(\pi_\theta) = \mathbb{E}\Big[\mathsf{Eva}_\pi^{\mathrm{BC}}\big((s_i, a_i), \tilde{a}_i\big)\Big] - \xi \cdot \mathbb{E}\Big[\mathsf{Eva}_\pi^{\mathrm{BC}}\big((s_j, a_j), \tilde{a}_k\big)\Big], \tag{3}$$

$$\text{where} \quad \mathsf{Eva}_\pi^{\mathrm{BC}}\big(s, a), \tilde{a}\big) = -\ell\big(\pi_\theta, (s, \tilde{a})\big) = \log \pi_\theta(\tilde{a}|s). \tag{4}$$

In (3), the first expectation is taken over $(s_i, \tilde{a}_i) \sim \widetilde{\mathcal{D}}_E, a_i \sim \pi(\cdot|s_i)$ and the second is taken over $(s_j, \tilde{a}_j) \sim \widetilde{\mathcal{D}}_E, a_j \sim \pi(\cdot|s_j), (s_k, \tilde{a}_k) \sim \widetilde{\mathcal{D}}_E, a_k \sim \pi(\cdot|s_k)$. Again, the second $\mathsf{Eva}_\pi^{\mathrm{BC}}$ term in J^{BC} serves the purpose of punishing over-agreement with the weak demonstration. Similarly, $\xi \geq 0$ is a parameter to balance the penalty for blind agreements.

Robustness to Noisy Demonstrations We prove that the policy learned by PeerBC converges to the expert policy when observing a sufficient amount of weak demonstrations. We focus on the binary action setting for theoretical analyses, where the action space is given by $\mathcal{A} = \{A_+, A_-\}$ and the weakness or noise in the weak expert $\tilde{\pi}_E$ is quantified by $e_+ = \mathbb{P}(\tilde{\pi}_E(s) = A_- | \pi_E(s) = A_+)$ and $e_- = \mathbb{P}(\tilde{\pi}_E(s) = A_+ | \pi_E(s) = A_-)$. Let $\pi_{\widetilde{\mathcal{D}}_E}$ be the optimal policy for maximizing the objective in (3) with imperfect demonstrations $\widetilde{D}_E$ (a particular set of with N i.i.d. imperfect demonstrations). Note $\ell(\cdot)$ is specified as the 0-1 loss: $\mathbb{1}(\pi(s), a) = 1$ when $\pi(s) \neq a$, otherwise $\mathbb{1}(\pi(s), a) = 0$. We have the following upper bound on the error rate.

Theorem 2. *Denote by $R_{\widetilde{\mathcal{D}}_E} := \mathbb{P}_{(s,a)\sim\mathcal{D}_E}(\pi_{\widetilde{\mathcal{D}}_E}(s) \neq a)$ the error rate for PeerBC. With probability at least $1 - \delta$, it is upper-bounded as:*

$$R_{\widetilde{\mathcal{D}}_E} \leq \frac{1 + \xi}{1 - e_- - e_+} \sqrt{\frac{2 \log 2/\delta}{N}}. \tag{5}$$

Theorem 2 states that as long as weak demonstrations are observed sufficiently, i.e., N is sufficiently large, the policy learned by PeerBC is able to converge to the clean expert policy $\pi_E(s)$ with a convergence rate of $\mathcal{O}\big(1/\sqrt{N}\big)$.

Peer Policy Co-Training Our discussion of BC allows us to study a more challenging co-training task [33]. Given a finite MDP $\mathcal{M}$, there are two agents that receive partial observations and

Table 1: Numerical performance of DDQN on CartPole with true reward (r), noisy reward ($\tilde{r}$), surrogate reward $\hat{r}$ [7], and peer reward $\tilde{r}_{\text{peer}}(\xi = 0.2)$. $\mathcal{R}_{avg}$ denotes average reward per episode after convergence, the higher (↑) the better; N_{epi} denotes total episodes involved in 10,000 steps, the lower (↓) the better.

		$e = 0.1$		$e = 0.2$		$e = 0.3$		$e = 0.4$	
		$\mathcal{R}_{avg} \uparrow$	$N_{epi} \downarrow$	$\mathcal{R}_{avg} \uparrow$	$N_{epi} \downarrow$	$\mathcal{R}_{avg} \uparrow$	$N_{epi} \downarrow$	$\mathcal{R}_{avg} \uparrow$	$N_{epi} \downarrow$
DDQN	r	195.6 ± 3.1	101.2 ± 3.2	195.6 ± 3.1	101.2 ± 3.2	195.6 ± 3.1	101.2 ± 3.2	195.2 ± 3.0	101.2 ± 3.3
	$\tilde{r}$	185.2 ± 15.6	114.6 ± 6.0	168.8 ± 13.6	123.9 ± 9.6	177.1 ± 11.2	133.2 ± 9.1	185.5 ± 10.9	163.1 ± 11.0
	$\hat{r}$	183.9 ± 10.4	110.6 ± 6.7	165.1 ± 18.2	113.9 ± 9.6	$\mathbf{192.2 \pm 10.9}$	115.5 ± 4.3	179.2 ± 6.6	125.8 ± 9.6
	$\tilde{r}_{\text{peer}}$	$\mathbf{198.5 \pm 2.3}$	$\mathbf{86.2 \pm 5.0}$	$\mathbf{195.5 \pm 9.1}$	$\mathbf{85.3 \pm 5.4}$	174.1 ± 32.5	$\mathbf{88.8 \pm 6.3}$	$\mathbf{191.8 \pm 8.5}$	$\mathbf{106.9 \pm 9.2}$

we let π^A and π^B denote the policies for agent A and B. Moreover, two agents are trained jointly to learn with rewards and noisy demonstrations from each other (e.g., at preliminary training phase). Symmetrically, we consider the case where agent A learns with the demonstrations from B on sampled trajectories, and π_B effectively serves as a noisy version of expert policy $\pi_E = \arg\max_{\pi \in \Pi} \mathbb{E}_{(s,a)\sim\mathcal{D}_E}[\log \pi(a|s)]$.

Algorithm 1 Peer policy co-training (PeerCT)

```
Require: Views $A$, $B$, MDPs $\mathcal{M}^A$, $\mathcal{M}^B$, policies $\pi_A, \pi_B$, mapping
    functions $f_{A\to B}, f_{B\to A}$ that maps states from one view to the
    other view, CA coefficient $\xi$, step size $\beta$ for policy update.
repeat
    Run $\pi^A$ to generate trajectories $\tau^A = \{(s_i^A, a_i^A, r_i^A)\}_{i=1}^N$.
    Run $\pi^B$ to generate trajectories $\tau^B = \{(s_j^B, a_j^B, r_j^B)\}_{j=1}^M$.
    Agents label the trajectories for each other
        $\tau'^A \leftarrow \Big\{\big(s_i^A, \pi^B\big(f_{B\leftarrow A}(s_i^A)\big)\big)\Big\}_{i=1}^N$,
        $\tau'^B \leftarrow \Big\{\big(s_j^B, \pi^A\big(f_{A\leftarrow B}(s_j^B)\big)\big)\Big\}_{j=1}^M$.
    Update policies: $\pi^{\{A,B\}} \leftarrow \pi^{\{A,B\}} + \beta \cdot \nabla J^{\mathrm{CT}}(\pi^{\{A,B\}})$
until convergence
```

For simplicity of demonstration, we focus on recovering the clean expert policy by only adapting the BC evaluation term (ignoring the effect of RL rewards, see Eqn. (6)). Denote by $\tau_\theta^A = \{(s_i^A, a_i^A, r_i^A)\}_{i=1}^N$ the trajectory that π^A generated via interacting with the partial world $\mathcal{M}^A$. Then π^B substitutes each action a_i^A with its selection $a_i'^B \sim \pi^B(\cdot|f_{A\to B}(s_i^A))$ as the weak supervision. Similar to PeerRL/PeerBC, the objective function of peer co-learning (PeerCT) becomes

$$J^{\mathrm{CT}}(\pi_\theta) = \mathbb{E}\Big[\mathsf{Eva}_\pi^{\mathrm{RL}}\big((s_i^A, a_i^A), r_i^A\big) + \mathsf{Eva}_\pi^{\mathrm{BC}}\big((s_i^A, a_i^A), a_i'^B\big)\Big] - \xi \cdot \mathbb{E}\Big[\mathsf{Eva}_\pi^{\mathrm{BC}}\big((s_j^A, a_j^A), a_k'^B\big)\Big], \quad (6)$$

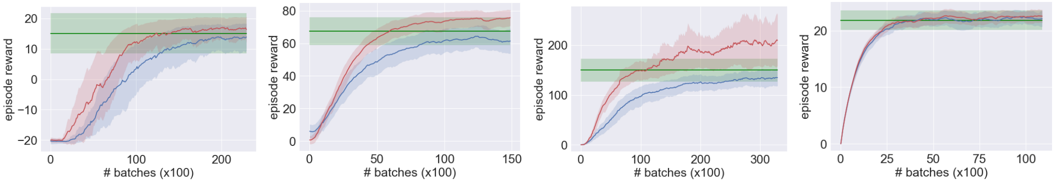
where the first expectation is taken over $(s_i^A, a_i^A, r_i^A) \sim \tau_\theta^A$, and $a_i'^B \sim \pi^B(\cdot|f_{A\to B}(s_i^A))$, and the second is taken over $(s_j^A, a_j^A, r_j^A) \sim \tau_\theta^A$, $(s_k^A, a_k^A, r_k^A) \sim \tau_\theta^A$, and $a_k'^B \sim \pi^B(\cdot|f_{A\to B}(s_k^A))$, ℓ is the loss function defined in Eqn. (4) to measure the policy difference, and $\mathsf{Eva}_\pi^{\mathrm{RL}}$, $\mathsf{Eva}_\pi^{\mathrm{BC}}$ are defined in Eqn. (2) and (4) respectively. The full algorithm PeerCT is provided in Algorithm 1. We omit detailed discussions on the convergence of PeerCT - it can be viewed as a straight-forward extension of Theorem 2 in the context of co-training.

4 Experiments

We evaluate our solution in three challenging weakly supervised PL problems. Experiments on control games and Atari show that, without any prior knowledge of the noise, our approach is able to leverage weak supervision more effectively.

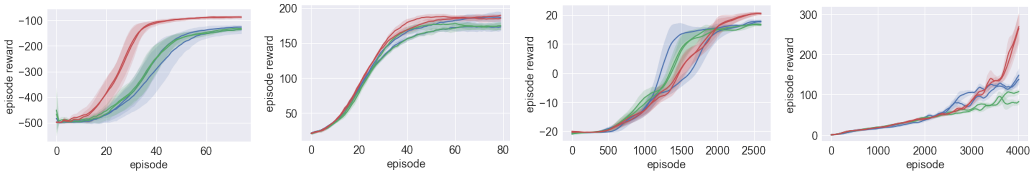
4.1 PeerRL with Noisy Reward

***CartPole*:** We first evaluate our method in RL with noisy reward setting. Following [7], we consider the binary reward $\{-1, 1\}$ for Cartpole where the symmetric noise is synthesized with different error rates $e = e_- = e_+$. We choose DQN [38] and DDQN [39] algorithms and train the models for 10,000 steps. We repeat each experiment 10 times with different random seeds and leave extra results in Appendix D.



(a) Pong (b) Boxing (c) Enduro (d) Freeway

Figure 4: Learning curves of BC on Atari. Standard BC 🟦, PeerBC (ours) 🟥, expert 🟩.



(a) Acrobot (b) CartPole (c) Pong (d) Breakout

Figure 5: Policy co-training on control/Atari. Single view 🟦, [33] 🟩, PeerCT (ours) 🟥.

Figure 2 shows the learning curves for DDQN with different approaches in noisy environments ($\xi = 0.2$) [1]. Since the number of training steps is fixed, the faster the algorithm converges, the fewer total episodes the agent will involve thus the learning curve is on the left side. As a consequence, the proposed peer reward outperforms other baselines significantly even in a high-noise regime (e.g., $e = 0.4$). Table 1 provides quantitative results on the average reward $\mathcal{R}_{avg}$ and total episodes N_{epi}. We find the agents with peer reward lead to a larger $\mathcal{R}_{avg}$ (less generalization error) and a smaller N_{epi} (faster convergence) consistently, which again verifies the effectiveness of our approach.

Pendulum: We further conduct experiments on a more challenging continuous control task *Pendulum*, where the goal is to keep a frictionless pendulum standing up. Since the rewards in pendulum are continuous: $r \in (-16.3, 0.0]$, we discretized it into 17 intervals: $(-17, -16], (-16, -15], \cdots, (-1, 0]$, with its value approximated using its maximum point. We experiment DDPG [42] and uniform noise in this environment. In Figure 3, the RL agents with the proposed CA objective successfully converge to the optimal policy under different amounts of noise. On the contrary, the agents with noisy rewards suffer from biased noise, especially in a high-noise regime. Prior work [7] has similar learning curves like peer reward on Pendulum and are omitted for clarity. However, we highlight that peer reward does not require any knowledge of noise rates or complicated estimation algorithms compared to [7].

Analysis of the benefits in PeerRL More surprisingly, we observed that the agents on CartPole with peer reward even lead to faster convergence than the ones observing true reward perfectly when the noise rate e is small. This indicates the possibility of other benefits to further promote peer reward, other than the noise reduction one we primarily focused on. We hypothesize this is because (1) peer reward scales the reward signals appropriately, which potentially reduces the variance and makes it easier to learn from; (2) the peer penalty term encourages explorations in RL implicitly (but then it removes the bias in the noisy supervision in expectation); (3) the human-specific "true reward" is also imperfect which leads to a weak supervision scenario. More discussions and analysis are deferred to Appendix C.5. However, we emphasize that the advantage of recovering from noisy reward signal is non-negligible, especially in a high-noise regime (e.g., $e = 0.4$ in Figure 2 and 3).

4.2 PeerBC from Weak Demonstrations

Atari: In BC setting, we evaluate our approach on four vision-based Atari games. For each environment, we train an imperfect RL model with PPO [43] algorithm. Here, "imperfect" means the training is terminated before convergence when the performance is about $70\% \sim 90\%$ as good as the fully converged model. We then collect the imperfect demonstrations using the expert model and generate 100 trajectories for each environment. The results are reported under three random seeds.

Figure 4 presents the comparisons for standard BC and PeerBC, from which we observe that our approach outperforms standard BC and even the expert it learns from. Note that during the whole

[1] We analysed the sensitivity of ξ and found the algorithm performs reasonable when $\xi \in (0.1, 0.4)$. More insights and experiments with varied ξ is deferred to Appendix D.

Table 2: BC from weak demonstrations. PeerBC successfully recovers better policies than expert.

Environment		Pong	Boxing	Enduro	Freeway	Lift (↑)
Expert		15.1 ± 6.6	67.5 ± 8.5	150.1 ± 23.0	21.9 ± 1.7	-
Standard BC		14.7 ± 3.2	56.2 ± 7.7	138.9 ± 14.1	22.0 ± 1.3	-6.6%
PeerBC	$\xi = 0.2$	$\mathbf{18.8 \pm 0.6}$	67.2 ± 8.4	177.9 ± 29.3	$\mathbf{22.5 \pm 0.6}$	$+11.3\%$
	$\xi = 0.5$	16.6 ± 4.0	$\mathbf{75.6 \pm 5.4}$	$\mathbf{230.9 \pm 73.0}$	22.4 ± 1.3	$\mathbf{+19.5\%}$
	$\xi = 1.0$	16.7 ± 4.3	69.7 ± 4.7	230.4 ± 61.6	8.9 ± 4.9	$+2.0\%$
Fully converged PPO		20.9 ± 0.3	89.3 ± 5.4	389.6 ± 216.9	33.3 ± 0.8	-

Table 3: Comparison with single view training and CoPiEr [33] on standard policy co-training.

Environment		Acrobot	CartPole	Pong	Breakout
Single View	A	-136.6 ± 15.6	172.8 ± 5.5	17.8 ± 0.6	148.0 ± 16.5
	B	-126.4 ± 8.0	186.7 ± 8.1	17.7 ± 0.5	137.8 ± 12.5
CoPiEr	A	-136.2 ± 5.2	174.1 ± 5.1	16.8 ± 0.5	107.5 ± 5.8
	B	-131.5 ± 4.5	174.3 ± 5.4	16.5 ± 0.2	82.7 ± 6.9
PeerCT	A	$\mathbf{-87.0 \pm 3.9}$	$\mathbf{188.8 \pm 2.7}$	$\mathbf{20.5 \pm 0.4}$	263.6 ± 36.0
	B	-87.1 ± 6.3	184.7 ± 3.9	20.4 ± 0.5	$\mathbf{268.6 \pm 33.6}$

training process, the agent never learns by interacting directly with the environment but only have access to the expert trajectories. Therefore, we owe this performance gain to PeerBC's strong ability for learning from weak supervision. The peer term we add not only provably eliminates the effects of noise but also extracts useful strategy from the demonstrations. See results in Table 2. Our approach consistently outperforms the expert and standard BC. As a reference, we also compare two other baselines GAIL [44] and SQIL [27] and provide the sensitivity analysis of ξ in Appendix D.

Analysis of benefits in PeerBC Similarly, the performance improvement of PeerBC might be also coupled with multiple possible factors. (1) The imperfect expert model might be a noisy version of the fully-converged agent since there are less visited states on which the selected actions of the model contains noise. (2) The improvements might be brought up by biasing against high-entropy policies thus PeerBC is useful when the true policy itself is deterministic. We provide more discussions about the second factor in Appendix C.6.

4.3 PeerCT for Standard Policy Co-training

***Continuous Control/Atari*:** Finally, we verify the effectiveness of the PeerCT algorithm in policy co-training setting [33]. This setting is more challenging since the states are partially observable and each agent needs to imitate another agent's behavior that is highly biased and imperfect. Note that we adopt the exact same setting as [33] **without any synthetic noise** included. This implies the potential of our approach to deal with natural noise in real-world applications. Following [33], we mask the first two dimensions respectively in the state vector to create two views for co-training in classic control games (Acrobot and CartPole). Similarly, the agent either removes all even index coordinates (view-A) in the state vector or removing all odd index ones (view-B) on Atari games. As shown in Table 3 and Figure 5, PeerCT algorithm outperforms training from single view, and CoPiEr algorithm consistently on both control games ($\xi = 0.5$ in Figure 5a, 5b) and Atari games ($\xi = 0.2$ in Figure 5c, 5d). In most cases, our approach leads to a faster convergence and lower generalization error compared to CoPiEr, which again verify that our ways of leveraging information from peer agent enables recovery of useful knowledge from highly imperfect supervision.

5 Conclusion

We have proposed PeerPL, a weakly supervised policy learning framework to unify a series of RL/BC problems with low-quality supervision signals. In PeerPL, instead of blindly memorizing the weak supervision, we evaluate a learning policy's correlated agreements with the weak supervision. We demonstrate how our method adapts in RL/BC and the hybrid co-training tasks and provide analysis of the convergence rate and sample complexity. Current theorems focus on the specific discrete noise model. Future work may extend it to more general noise scenarios and evaluate our method on real RL/BC systems, such as robotics and self-driving.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, abs/1910.07113, 2019.

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[4] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional behavior cloning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[5] Tom Everitt, Victoria Krakovna, Laurent Orseau, and Shane Legg. Reinforcement learning with a corrupted reward channel. In *IJCAI*, pages 4705–4713, 2017.

[6] Joshua Romoff, Alexandre Piché, Peter Henderson, Vincent François-Lavet, and Joelle Pineau. Reward estimation for variance reduction in deep reinforcement learning. In *ICLR (Workshop)*. OpenReview.net, 2018.

[7] Jingkang Wang, Yang Liu, and Bo Li. Reinforcement learning with perturbed rewards. In *AAAI*, 2020.

[8] Michael Laskey, Jonathan Lee, Roy Fox, Anca D. Dragan, and Ken Goldberg. DART: noise injection for robust behavior cloning. In *CoRL*, volume 78 of *Proceedings of Machine Learning Research*, pages 143–156. PMLR, 2017.

[9] Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Behavior cloning from imperfect demonstration. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 6818–6827. PMLR, 2019.

[10] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. SQIL: behavior cloning via reinforcement learning with sparse rewards. In *ICLR*. OpenReview.net, 2020.

[11] Xiaoxiao Guo, Shiyu Chang, Mo Yu, Gerald Tesauro, and Murray Campbell. Hybrid reinforcement learning with expert state sequences. In *AAAI*, pages 3739–3746. AAAI Press, 2019.

[12] Lisa Lee, Benjamin Eysenbach, Ruslan Salakhutdinov, Shixiang, Gu, and Chelsea Finn. Weakly-supervised reinforcement learning for controllable behavior, 2020.

[13] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359 –1373, 2005.

[14] Anirban Dasgupta and Arpita Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, pages 319–330, 2013.

[15] Victor Shnayder, Arpit Agarwal, Rafael M. Frongillo, and David C. Parkes. Informed truthfulness in multi-task peer prediction. In *EC*, pages 179–196. ACM, 2016.

[16] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.

[17] Clayton Scott, Gilles Blanchard, Gregory Handy, Sara Pozzi, and Marek Flaska. Classification with asymmetric label noise: Consistency and maximal denoising. In *COLT*, pages 489–511, 2013.

[18] Clayton Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *AISTATS*, 2015.

[19] Sainbayar Sukhbaatar and Rob Fergus. Learning from noisy labels with deep neural networks. *arXiv preprint arXiv:1406.2080*, 2(3):4, 2014.

[20] Brendan van Rooyen and Robert C Williamson. Learning in the presence of corruption. *arXiv preprint arXiv:1504.00091*, 2015.

[21] Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *ICML*, pages 125–134, 2015.

[22] Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to anchor points when learning with noisy labels. *arXiv preprint arXiv:2102.05291*, 2021.

[23] Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

[24] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.

[25] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[26] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

[27] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. SQIL: Behavior Cloning via Reinforcement Learning with Sparse Rewards. 2019.

[28] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *IJCAI*, pages 4950–4957. ijcai.org, 2018.

[29] Juarez Monteiro, Nathan Gavenski, Roger Granada, Felipe Meneguzzi, and Rodrigo Coelho Barros. Augmented behavioral cloning from observation. *CoRR*, abs/2004.13529, 2020.

[30] Alessandro Giusti, Jerome Guzzi, Dan C. Ciresan, Fang Lin He, Juan P. Rodriguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jurgen Schmidhuber, Gianni Di Caro, Davide Scaramuzza, and Luca M. Gambardella. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2016.

[31] Niels Justesen and Sebastian Risi. Learning macromanagement in starcraft from replays using deep learning. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 162–169. IEEE, 2017.

[32] Wael Farag and Zakaria Saleh. Behavior cloning for autonomous driving using convolutional neural networks. *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2018*, 2018.

[33] Jialin Song, Ravi Lanka, Yisong Yue, and Masahiro Ono. Co-training for policy learning. In *UAI*, page 441. AUAI Press, 2019.

[34] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. *ICML*, abs/1910.03231, 2020.

[35] Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E. Taylor, and Ann Nowé. Reinforcement learning from demonstration through shaping. In *IJCAI*, pages 3352–3358. AAAI Press, 2015.

[36] Todd Hester, Matej Vecerík, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. In *AAAI*, pages 3223–3230. AAAI Press, 2018.

[37] Yang Liu. The importance of understanding instance-level noisy labels. *arXiv preprint arXiv:2102.05336*, 2021.

[38] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[39] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In *ICML*, volume 48, pages 1995–2003, 2016.

[40] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, pages 1057–1063. The MIT Press, 1999.

[41] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. In *Machine Learning*, pages 279–292, 1992.

[42] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015.

[43] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[44] Jonathan Ho and Stefano Ermon. Generative adversarial behavior cloning. In *Advances in Neural Information Processing Systems*, pages 4572–4580, 2016.

[45] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, pages 2094–2100, 2016.

[46] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, pages 278–287. Morgan Kaufmann, 1999.

[47] John Asmuth, Michael L. Littman, and Robert Zinkov. Potential-based shaping in model-based reinforcement learning. In *AAAI*, pages 604–609. AAAI Press, 2008.

[48] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior (commemorative edition)*. Princeton university press, 2007.

[49] Tommi S. Jaakkola, Michael I. Jordan, and Satinder P. Singh. Convergence of stochastic iterative dynamic programming algorithms. In *NIPS*, pages 703–710, 1993.

[50] John N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine Learning*, 16(3):185–202, 1994.

[51] Michael J. Kearns and Satinder P. Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In *NIPS*, pages 996–1002, 1998.

[52] Michael J. Kearns and Satinder P. Singh. Bias-variance error bounds for temporal difference updates. In *COLT*, pages 142–147, 2000.

[53] Michael J. Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. In *IJCAI*, pages 1324–1231, 1999.

[54] Sham Machandranath Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University of London, 2003.

[55] Kiante Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized behavior cloning. In *Eighth International Conference on Learning Representations (ICLR)*, April 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] see future work in Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our works aim to improve the robustness of policy learning algorithms which is relevant to applications concerning fairness and training data biases. We acknowledge that the use of AI technology may bring us an unexpected impact. While we are not aware of any negative social impact, we caution that our theoretical guarantees are mostly for the scenario with a large number of samples. Using our method when the number of weak supervision is very limiting might lead to unstable performance and unintended consequences, especially when the supervision are highly noisy.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide our code and pre-trained models for reproducibility.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]