Policy Learning Using Weak Supervision

Jingkang Wang[*1,2] **Hongyi Guo**[*3] **Zhaowei Zhu**[*4] **Yang Liu**[4]
University of Toronto[1], Vector Institute[2], Northwestern University[3], UC Santa Cruz[4]
wangjk@cs.toronto.edu hongyiguo2025@u.northwestern.edu
zwzhu@ucsc.edu yangliu@ucsc.edu

Abstract

Most existing policy learning solutions require the learning agents to receive high-quality supervision signals such as well-designed rewards in reinforcement learning (RL) or high-quality expert demonstrations in behavioral cloning (BC). These quality supervisions are usually 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 rewards, BC with weak demonstrations, and standard policy co-training 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 breakthroughs in policy learning (PL) open up the possibility to apply reinforcement learning (RL) or behavioral cloning (BC) in real-world applications such as robotics [1, 2] and self-driving [3, 4]. Most existing works require agents to receive high-quality supervision signals, e.g., reward or expert demonstrations, which are either infeasible or expensive to obtain in practice [5, 6].

The outputs of reward functions in RL are subject to multiple kinds of randomness. For example, the reward collected from sensors on a robot may be biased and have inherent noise due to physical conditions such as temperature and lighting [7, 8, 9]. For the human-defined reward, different human instructors might provide drastically different feedback that leads to biased rewards [10]. Besides, the demonstrations by an expert in behavioral cloning (BC) are often imperfect due to limited resources and environment noise [11, 12, 13]. Therefore, learning from weak supervision signals such as noisy rewards [7] or low-quality demonstrations produced by untrustworthy expert [12, 14] is one of the outstanding challenges that prevents a wider application of PL.

Although some works have explored these topics separately in their specific domains [7, 15, 14, 16], there lacks a unified solution for robust policy learning in imperfect situations. Moreover, the noise model as well as the corruption level in supervision signals is often required. To handle these challenges, we first formulate a meta-framework to study RL/BC with weak supervision and call it *weakly supervised policy learning*. Then we propose a theoretically principled solution, PeerPL, to perform efficient policy learning using the available weak supervision without requiring noise rates.

Our solution is inspired by peer loss [17], a recently proposed loss function for learning with noisy labels but does not require the specification of noise rates. In peer loss, the noisy labels are treated as a *peer agent*'s supervision. This loss function explicitly punishes the classifier from simply agreeing with the noisy labels, but would instead reward it for a "correlated agreement" (CA). We adopt a

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

similar idea and treat the "weak supervision" as the noisy 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 and evaluation 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 corruption level in supervision signals.

To summarize, the contributions in the paper are: (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, and demonstrate how it adapts in challenging tasks including RL with noisy rewards and BC from weak demonstrations; (3) PeerPL is theoretically guaranteed to recover the optimal policy, as if the supervision are of high-quality and clean. (4) Experiment results show strong evidence that PeerPL brings significant improvements over state-of-the-art solutions. Code is online available at: https://github.com/wangjksjtu/PeerPL.

1.1 Related Work

Learning with Noisy Supervision Learning from noisy supervision is a widely explored topic. The seminal work [18] 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 level from model predictions [19, 20, 21, 22, 23, 24, 25, 26, 27] or label consensuses of nearby representations [28]. Recent works also studied this problem in sequential settings including federated bandit [29] and RL [7]. The former work assumes the noise can be offset by averaging rewards from multiple agents. [7] designs a statistics-based estimation algorithm for noise rates in observed rewards, which can be inefficient especially when the state-action space is huge. Moreover, the error in the estimation can accumulate and amplify in sequential problems. Inspired by recent advances of *peer loss* [17, 30, 31], our solution is able to recover true supervision signals without requiring a priori specification of the noise rates.

Behavioral Cloning (BC) Standard BC [32, 33] 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 about the sequential consequences of actions. Therefore, the agent suffers from compounding errors when there is a mismatch between demonstrations and real states encountered [33, 34, 35]. Recent works introduce data augmentations [36] and value-based regularization [37] or inverse dynamics models [38, 39] to encourage learning long-horizon behaviors. While being simple and straightforward, BC has been widely investigated in a range of application domains [40, 41] and often yields competitive performance [42, 37]. 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 [43].

Correlated Agreement In [44, 45], a correlated agreement (CA) type of mechanism is proposed to evaluate the correlations between agents' reports. In addition to encouraging a certain agreement between agents' reports, CA also punishes over-agreement when two agents always report identically. Recently, [17, 30, 25] adapt a similar idea to noisy label learning thus offloading the burdens of estimating noise rates. We consider a more challenging sequential decision-making problem and study the convergence rates under noisy supervision signals.

2 Policy Learning from Weak Supervision

We begin by reviewing conventional reinforcement learning and behavioral cloning with clean supervision signals. Then we introduce the weak supervision problem in policy learning and define two concrete instantiations: (1) RL with noisy reward and (2) BC using weak expert demonstrations.

2.1 Overview 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 *reinforcement learning*, 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

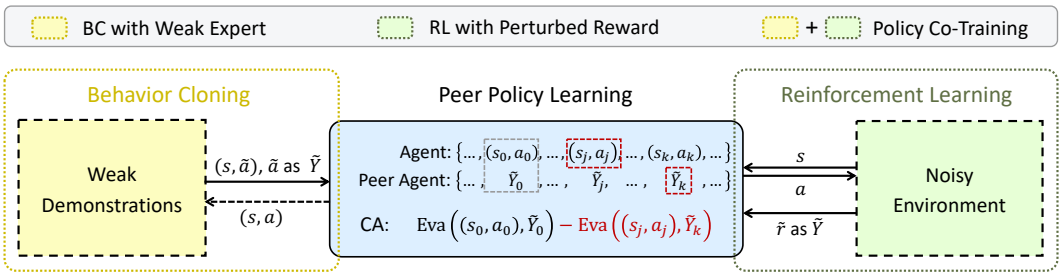


Figure 1: Illustration of *weakly supervised policy learning* and our PeerPL solution with correlated agreement (CA). We use $\widetilde{Y}$ to denote a weak supervision, be it a noisy reward, or a noisy demonstration. Eva stands for an evaluation function. "Peer Agent" corresponds to weak supervision.

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: $J^{\text{clean}}(\pi) = \mathbb{E}_{(s_t,a_t,r_t)\sim\tau}[\sum_{t=0}^{T} \gamma^t r_t]$, where $\gamma \in (0, 1]$ is the discount factor.

Another popular policy learning method is *behavioral cloning*. Let $\pi(\cdot|s)$ denotes the distribution over actions formed by π, and $\pi(a|s)$ be the probability of choosing action a given state s and policy π. 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 $\mathcal{D}_E$, where (s_i, a_i) is the sampled state-action pair from the expert trajectory and $a_i \sim \pi_E(\cdot|s_i)$ Then training a policy with standard BC corresponds to maximizing the following log-likelihood: $J^{\text{clean}}(\pi) = \mathbb{E}_{(s,a)\sim\mathcal{D}_E}[\log \pi(a|s)]$.

In both RL and BC, the learning agent receives supervision through either the *(clean) 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 $\pi^* = \arg\max_{\pi\in\Pi} J^{\text{clean}}(\pi)$: 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 learning paradigm [46, 47, 15, 43]. Specifically, a recent hybrid framework called policy co-training [43] will be considered in this paper.

2.2 Weak Supervision in Policy Learning

The *weak supervision signal* $\widetilde{Y}$ could be noisy reward $\tilde{r}$ for RL or noisy action $\tilde{a}$ from an imperfect 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 unknown 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: $c^{\text{RL}}_{j,k} = \mathbb{P}(\tilde{r}_t = R_k | r_t = R_j)$. We call r and $\tilde{r}$ the *true reward* and *noisy reward*.

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 $\tilde{a}_i$ is is the noisy expert action drawn according to a random variable $\tilde{a}_i = \tilde{\pi}_E(s_i) \sim \tilde{\pi}_E(\cdot|s_i)$, each state-action pair $(s_i, \tilde{a}_i)$ is sampled from distribution $\widetilde{\mathcal{D}}_E$. Note there may exist two randomness factors in getting $\tilde{a}_i$: uncertainty in true policy π_E and noise from imperfect policy $\tilde{\pi}_E$. In particular, we do not consider the former randomness in theoretical analyses: given the output distribution $\pi_E(\cdot|s_i)$, only one deterministic action $\pi_E(s_i)$ is taken by expert. This is because with uncertainty in true expert actions, it is hard to distinguish a clean case with true expert actions from the weak supervision case without addition knowledge. Similar assumptions are also adopted in [23, 28]. The noisy action is modeled by a unknown 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 $\pi_E(s)$: $c^{\text{BC}}_{j,k} = \mathbb{P}(\tilde{\pi}_E(s) = A_k | \pi_E(s) = A_j)$, A_k and A_j denote the k-th and the j-th action from the action space $\mathcal{A}$. In the above definition, we assume the noisy action $\tilde{a}_i$ is independent of the state s given the deterministic expert action $\pi_E(s)$,

i.e., $\mathbb{P}(\tilde{a}_i|\pi_E(s_i)) = \mathbb{P}(\tilde{a}_i|s_i, \pi_E(s_i))$. 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 Recall $\mathbf{C}$: $\mathbf{C}^{\mathrm{RL}}_{|\mathcal{R}|\times|\mathcal{R}|}$ or $\mathbf{C}^{\mathrm{BC}}_{|\mathcal{A}|\times|\mathcal{A}|}$ is unknown in practice. While recent works estimate this matrix [26, 23, 28] in supervised classification problems, it is still challenging to generalize them to a sequential setting [7]. When $\mathbf{C}$ is not perfectly estimated, the estimation error of $\mathbf{C}$ may lead to unexpected state-action pairs then the error of reward estimates will be accumulated in sequential learning. Besides, estimating $\mathbf{C}$ involves extra computation burden. In contrast, our method gets rid of the above issues since it is free of any knowledge of $\mathbf{C}$ and leads to more robust policy learning algorithms.

Learning Goal 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 [48]. 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 $\{(s_t, a_t), \widetilde{Y}_t\}_{t=1}^T$ (RL) or $\{(s_i, a_i), \widetilde{Y}_i\}_{i=1}^N$ (BC).

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 A Unified Evaluation Function

We use an evaluation function $\mathsf{Eva}_\pi((s_i, a_i), \widetilde{Y}_i)$ to evaluate a taken policy π at agent state (s_i, a_i) using the weak supervision $\widetilde{Y}_i$. For RL, Eva_π is the instance-wise measure (negative loss) for different RL algorithms, which is a function of the noisy reward $\tilde{r}$ received at (s_i, a_i). In the BC setting, Eva_π is the loss to evaluate the action a_i taken by the agent given the expert's demonstration $\tilde{a}_i$. Note that the larger the Eva_π is at state (s_i, a_i), the better it follows the supervision $\widetilde{Y}_i$. Specifically, we have

$$\mathsf{Eva}^{\mathrm{RL}}_\pi\big((s,a),\tilde{r}\big) = -\ell\big(\pi,(s,a,\tilde{r})\big) \ \text{(RL)} \quad \text{and} \quad \mathsf{Eva}^{\mathrm{BC}}_\pi\big((s,a),\tilde{a}\big) = \log\pi(\tilde{a}|s) \ \text{(BC)},$$

where the RL loss function ℓ can be temporal difference error [49, 50] or the policy gradient loss [51]. Furthermore, we let $J(\pi)$ denote the function that evaluates policy π under a set of state action pairs with weak supervision sequence $\{(s_i, a_i), \widetilde{Y}_i\}_{i=1}^N$, i.e.,

$$J(\pi) = \mathbb{E}_{(s,a)\sim\tau}[\mathsf{Eva}_\pi((s,a),\widetilde{Y})].$$

Then the goal of weakly supervised policy learning is to recover the optimal policy π^* as if we receive clean supervision Y. Note that directly maximizing $J(\pi)$ might result in sub-optimal performance due to the weak supervisions. The above unified notations are only for better delivery of our framework and we still treat PL as a sequential decision problem.

3.2 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 weakly supervised sample $((s_i, a_i), \widetilde{Y}_i)$, we randomly sample (with replacement) two other peer samples 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 $((s_i, a_i), \widetilde{Y}_i)$ as follows:

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

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 and outputs the same sequence of actions given the same sequence of states, i.e.,

$$\text{Weak-supervision: } \tilde{a}_1 = \tilde{a}_2 = \tilde{a}_3 = 1, \tilde{a}_4 = 0; \quad \text{Outputs: } a_1 = a_2 = a_3 = 1, a_4 = 0.$$

Let $\mathsf{Eva}_\pi((s_i, a_i), \tilde{a}_i) = 1$ if the policy output agrees with the weak demonstration ($a_i = \tilde{a}_i$), and 0 otherwise. When the policy fully memorizes weak supervisions, we have:

$$\begin{aligned} \underline{\textit{Without CA:}} \quad & \mathbb{E}[\mathsf{Eva}_\pi((s_i, a_i), \tilde{a}_i)] = 1, \\ \underline{\textit{With CA:}} \quad & \mathbb{E}[\mathsf{Eva}_\pi((s_i, a_i), \tilde{a}_i) - \mathsf{Eva}_\pi((s_j, a_j), \tilde{a}_k)] = 0.375, \end{aligned}$$

where $0.375 = 1 - (0.75^2 + 0.25^2)$ is obtained by considering the probability of randomly paired a_j and $\tilde{a}_k$ matching each other. The above example shows that a full agreement with the weak supervision will instead be punished.

In what follows, we showcase two concrete implementations: PeerRL (peer reinforcement learning) and PeerBC (peer behavioral cloning). We provide algorithms and theoretical guarantees under weak supervisions.

4 PeerRL: Peer Reinforcement Learning

We propose the following objective function to punish the over-agreement of parametric policy π_θ 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 $(s_i, a_i, \tilde{r}_i) \sim \tilde{\tau}$ and second one is taken over $(s_j, a_j, \tilde{r}_j) \sim \tilde{\tau}, (s_k, a_k, \tilde{r}_k) \sim \tilde{\tau}$, 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 [49, 50] or the policy gradient loss [51]. The learning sequence is encoded in π. The objective $J^{\mathrm{RL}}(\pi)$ represents the accumulated peer RL reward. Parameter $\xi \geq 0$ balances the penalty for blind agreements induced by CA.

4.1 Peer Reward

In what follows, we consider the Q-Learning [52] as the underlying learning algorithm where $\ell(\pi_\theta, (s, a, \tilde{r})) = -\tilde{r}(s, a)$ 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*:

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

where $\tilde{r}' \overset{\pi_{\mathrm{sample}}}{\sim} \{\tilde{r}(s, a) | s \in \mathcal{S}, a \in \mathcal{A}\}$ 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 $\tilde{r}'$. In experiments, we find $\tilde{r}_{\mathrm{peer}}$ is not sensitive to the choice of ξ and keep ξ constant for each run.

Robustness to Noisy Rewards Now we show peer reward $\tilde{r}_{\mathrm{peer}}$ offers us an affine transformation of the true reward in expectation, which guarantees that our PeerRL algorithm converges to π^*. 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 $\mathbf{C}^{\mathrm{RL}}_{|\mathcal{R}|\times|\mathcal{R}|}$ in the binary setting).

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

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

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

Lemma 1 shows that by subtracting the peer penalty term $\tilde{r}'$ from noisy reward $\tilde{r}(s,a)$, $\tilde{r}_{\text{peer}}(s,a)$ recovers the clean and true reward $r(s,a)$ in expectation. Based on Lemma 1, we prove in Theorem A1 that the Q-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 work [7] that requires such knowledge). Moreover, we prove in Theorem A2 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 [49] and policy gradient [51] in Appendix A.

4.2 Why does Peer Reward Work?

Compared with noisy reward, proposed peer variant is a less biased estimation of true reward *(Benefit-1)*. On the other hand, PeerRL helps break the unstable "tie" states, which might encourage the agent to explore in the early stage [53] *(Benefit-2)*.

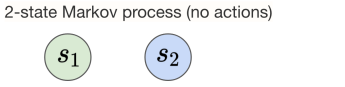
Benefit-1: PeerRL reduces the bias We highlight that the biased noise model considered is rather generic, departing from the previous noise assumption such as zero-mean Gaussian noise [8, 9]. In zero-mean noise models, the major focus is on variance reduction so adding the random term $\tilde{r}'$ increases the variance thus resulting in worse estimation. However, in the discrete biased noise model [18], bias correction also plays an important role especially the noise rate is high [7].

Similar to peer reward (Lemma 1), the expectation of the noisy reward writes as: $\mathbb{E}[\tilde{r}(s,a)] = (1-e_- - e_+)\mathbb{E}[r(s,a)] + e_- r_+ + e_+ r_- = (1-e_- - e_+)\mathbb{E}[r(s,a)] + \textit{const}$. But the constant in peer reward has less effect on the true reward r, especially when the noise rate is high. To see this:

$$\begin{aligned}\text{noisy reward:}\quad & \mathbb{E}[\tilde{r}(s,a)] = \eta \cdot \left(\mathbb{E}[r(s,a)] + \tfrac{e_+}{1-e_- - e_+} r_- + \tfrac{e_-}{1-e_- - e_+} r_+\right),\\ \text{peer reward:}\quad & \mathbb{E}[\tilde{r}_{\text{peer}}(s,a)] = \eta \cdot (\mathbb{E}[r(s,a)] - (1-p_{\text{peer}}) r_- - p_{\text{peer}} r_+),\end{aligned}$$

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 $\mathbb{E}[r(s,a)]$. On the contrary, $\mathbb{E}[\tilde{r}_{\text{peer}}(s,a)]$ contains a moderate constant noise thus maintaining more useful training signals of the true reward in practice. In summary, although peer reward (similar to the surrogate reward in previous literature [7]) increases the variance (no free-lunch), it will lead to a better estimation of the true reward due to lower bias.

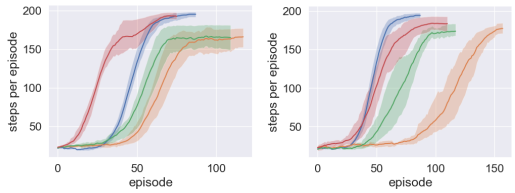
Benefit-2: PeerRL helps break ties For RL, "tie" states indicate that the rewards for different states are the same, which are less informative as they neither serve as positive nor negative examples. Due to the discrete nature of the noise model, adding a randomly sampled penalty term helps break the tie states and treats them as either positive examples or negative examples such that it can encourage exploration in the early stage, which has similar intuitions to some RL exploration works [53]. It has also been demonstrated that reducing the uncertainty, a.k.a. pushing confident predictions, makes the learning robust to weak-supervisions in supervised learning [17, 54]. On the other hand, it is known that positive examples are sparse yet important in RL. To leverage these useful experiences sufficiently, experience replay [55, 56] is invented to store and up-sample the positive examples for faster convergence. Tie breaking potentially provides an alternative way to access more positive examples. To illustrate *tie-breaking* phenomenon when using peer reward, we consider a two-state Markov process (no actions) with bounded Gaussian noise and see how well we could infer which state was better by correcting the reward signals. We collect two observations for each state and conduct 10^4 trials to calculate the success rate of inferring which state has larger returns ("correct" in the Table). As we can see, PeerRL exploits the "discreteness" of the reward thus breaking ties to obtain more examples with good-quality supervision. More examples on varied noise models (bounded continuous noise, discrete noise) are deferred to Appendix B.



$r_1 \sim \text{clamp}[\mathcal{N}(0.6,1), \text{min}=0, \text{max}=1]$

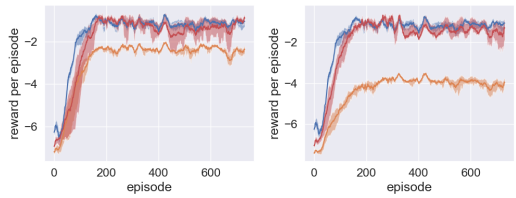
$r_2 \sim \text{clamp}[\mathcal{N}(0.4,1), \text{min}=0, \text{max}=1]$

	Correct	Tie	Incorrect
baseline	54.6%	5.6%	39.8%
PeerRL	**58.0%**	**0.3%**	**41.7%**



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

Figure 2: Learning curves of DDQN on CartPole-v0 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 [57] on Pendulum with true reward (r), noisy reward ($\tilde{r}$), and peer reward ($\tilde{r}_{\text{peer}}$, $\xi = 0.2$).

5 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 $\mathsf{Eva}_\pi^{\text{BC}}$ function in BC evaluates the agent policy π_θ, parametrized by 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:

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

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

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 $\mathsf{Eva}_\pi^{\text{BC}}$ 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.

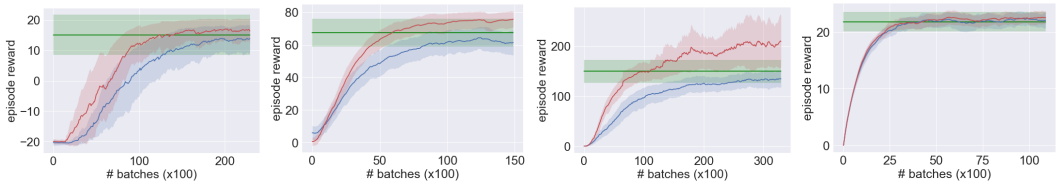
Robustness to Noisy Demonstrations We prove that the policy learned by PeerBC converges to the expert policy when observing a sufficient amount of weak demonstrations. We focus on the binary action setting for theoretical analyses, where the action space is given by $\mathcal{A} = \{A_+, A_-\}$ and the 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 $\pi_{\widetilde{D}_E}$ 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 1. *Denote by $R_{\widetilde{D}_E} := \mathbb{P}_{(s,a)\sim\mathcal{D}_E}(\pi_{\widetilde{D}_E}(s) \neq a)$ the error rate for PeerBC. When $e_+ + e_- < 1$, with probability at least $1 - \delta$, it is upper-bounded as: $R_{\widetilde{D}_E} \leq \frac{1+\xi}{1-e_--e_+}\sqrt{\frac{2\log 2/\delta}{N}}$.*

Theorem 1 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 [43]. Given a finite MDP $\mathcal{M}$, there are two agents that receive partial observations and 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 the 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.

Following [43], we assume a mapping function $f_{A\to B}$ exists that transforms states under view A into B. Denote by $\tau^A = \{(s_i^A, a_i^A, r_i^A)\}_{i=1}^N$ the trajectory that π_A generates via interacting with the partial world $\mathcal{M}^A$. Then π_B replaces each action a_i^A with its selection $\tilde{a}_i^B = \pi_B(f_{A\to B}(s_i^A))$ as the weak supervision. To recover the clean expert policy, we adapt the BC peer evaluation term to the



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

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

co-learning objective function:

$$J^{\mathrm{CT}}(\pi_\theta) = \mathbb{E}\Big[\mathsf{Eva}_\pi^{\mathrm{RL}}\big((s_i^A, a_i^A), r_i^A\big) + \mathsf{Eva}_\pi^{\mathrm{BC}}\big((s_i^A, a_i^A), \tilde{a}_i^B\big)\Big] - \xi \cdot \mathbb{E}\Big[\mathsf{Eva}_\pi^{\mathrm{BC}}\big((s_j^A, a_j^A), \tilde{a}_k^B\big)\Big], \quad (5)$$

where the first expectation is taken over $(s_i^A, a_i^A, r_i^A) \sim \tau^A$, and $\tilde{a}_i^B = \pi_B(f_{A\to B}(s_i^A))$, and the second is taken over $(s_j^A, a_j^A, r_j^A) \sim \tau^A, (s_k^A, a_k^A, r_k^A) \sim \tau^A$, and $\tilde{a}_k^B = \pi_B(f_{A\to B}(s_k^A))$, 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 1 in the context of co-training.

6 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.

Algorithm 1 Peer policy co-training (PeerCT)

Require: Views A, B, MDPs $\mathcal{M}^A$, $\mathcal{M}^B$, policies π_A, π_B, mapping functions $f_{A\to B}, f_{B\to A}$ that maps states from one view to the other view, CA coefficient ξ, step size β for policy update.

1: **repeat**
2: Run π^A to generate trajectories $\tau^A = \{(s_i^A, a_i^A, r_i^A)\}_{i=1}^N$.
3: Run π^B to generate trajectories $\tau^B = \{(s_j^B, a_j^B, r_j^B)\}_{j=1}^M$.
4: Agents label the trajectories for each other

$$\tilde{\tau}^A \leftarrow \big\{(s_i^A, \pi_B\big(f_{B\leftarrow A}(s_i^A)\big)\big\}_{i=1}^N,$$
$$\tilde{\tau}^B \leftarrow \big\{(s_j^B, \pi_A\big(f_{A\leftarrow B}(s_j^B)\big)\big\}_{j=1}^M.$$

5: Update policies: $\pi_{\{A,B\}} \leftarrow \pi_{\{A,B\}} + \beta \cdot \nabla J^{\mathrm{CT}}(\pi_{\{A,B\}})$
6: **until** convergence

Experiment Setup & Baselines We evaluate PeerPL on a wide variety of control and Atari games. For RL with noisy reward, we add synthetic noise to reward signals and compare with previous work [7], where an unbiased estimator of true reward is constructed by approximating the confusion matrix. For BC from weak demonstrations, we adopt not fully converged PPO agents as the weak experts and unroll the trajectories. We also consider a standard policy co-training setting [43] without any synthetic noise added and compare PeerCT with single-view training paradigm and CoPiEr [43].

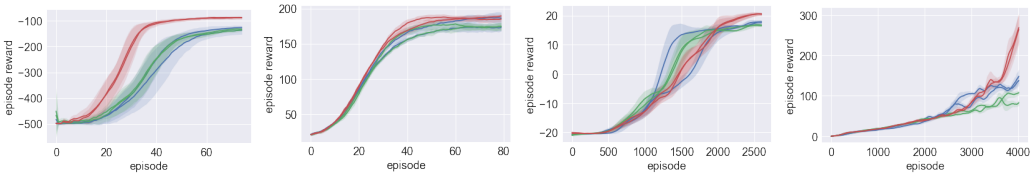
6.1 PeerRL with Noisy Reward

***CartPole-v0*:** 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 [49] and DDQN [50] 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. 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.

[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 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. Note $0 \leq e < 0.5$.

		$e = 0.1$		$e = 0.2$		$e = 0.3$		$e = 0.4$	
		$\mathcal{R}_{avg}$ ↑	N_{epi} ↓	$\mathcal{R}_{avg}$ ↑	N_{epi} ↓	$\mathcal{R}_{avg}$ ↑	N_{epi} ↓	$\mathcal{R}_{avg}$ ↑	N_{epi} ↓
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	$\mathbf{88.8 \pm 6.3}$	$\mathbf{191.8 \pm 8.5}$	$\mathbf{106.9 \pm 9.2}$



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

Figure 5: Policy co-training on control/Atari. Single view ■, [43] ■, PeerCT (ours) ■.

***Pendulum*:** We further conduct experiments on a 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 test DDPG [57] with uniform noise in this environment following [7]. 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.

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) the peer penalty term breaks the tie states (Benefit-2 in Section 4.1) and encourages explorations in RL; (2) PeerRL scales the reward signals appropriately for easier learning; (3) the human-specific "true reward" might be also imperfect which leads to a weak supervision scenario. 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).

6.2 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 [58] 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 shows that our approach outperforms standard BC and even the expert it learns from. Note that during the whole 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. As shown in Table 2, our approach consistently outperforms the expert and standard BC. We provide the sensitivity analysis of ξ in Appendix D.

Comparison with imitation learning baselines We further extend the empirical study to imitation learning (IL) algorithms on *CartPole-v1*. To collect weak demonstrations, we train a PPO agent for 50k iterations that are not fully converged. As shown in Figure 6, standard IL algorithms such as BC, AIRL [37], or GAIL [59] cannot handle noisy demonstrations well and lead to sub-optimal performance. Our PeerBC brings 18% improvement over standard BC by penalizing blind agreements with the weak demonstrations. We remark that performance of PeerBC is worse than DAgger due to notorious distribution shift issue. To further improve performance, we train PeerBC in the DAgger fashion (Peer-DAgger) by querying the imperfect expert to augment the training sets. Not surprisingly,

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 [43] 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	$\mathbf{268.6 \pm 33.6}$

Peer-DAgger surpasses DAgger by a large margin, which indicates that our framework has wide applicability and successfully recovers the true supervision signals. Adapting PeerPL idea to more IL algorithms such as GAIL [59] and DART [35] together with rigorous analysis is left as future works.

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 D.5.

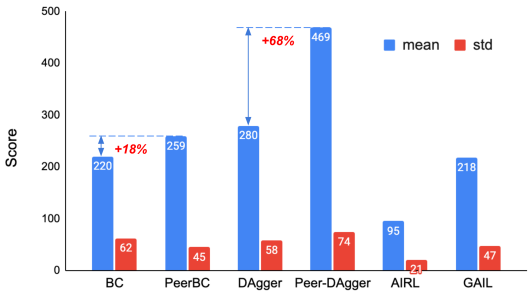


Figure 6: Comparison of imitation learning approaches on *CartPole-v1* with imperfect expert.

6.3 PeerCT for Standard Policy Co-training

***Continuous Control/Atari*:** Finally, we verify the effectiveness of the PeerCT algorithm in policy co-training setting [43]. 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 [43] **without any synthetic noise** included. This implies the potential of our approach to deal with natural noise in real-world applications. Following [43], 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, showing that our ways of leveraging information from peer agent enables recovery of useful knowledge from highly imperfect supervision.

7 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.

Acknowledgement

We sincerely thank the anonymous reviewers for their insightful suggestions. Our final version benefited substantially from the discussions with Reviewer 58fG. In particular, the tie-breaking analysis together with the code snippet is designed and contributed by Reviewer 58fG. This work is partially supported by the National Science Foundation (NSF) under grant IIS-2007951 and the Office of Naval Research under grant N00014-20-1-2240. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

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] Vibhu Agarwal, Tanya Podchiyska, Juan M Banda, Veena Goel, Tiffany I Leung, Evan P Minty, Timothy E Sweeney, Elsie Gyang, and Nigam H Shah. Learning statistical models of phenotypes using noisy labeled training data. *Journal of the American Medical Informatics Association*, 23(6):1166–1173, 2016.

[6] Yang Gao, Huazhe Xu, Ji Lin, Fisher Yu, Sergey Levine, and Trevor Darrell. Reinforcement learning from imperfect demonstrations. *arXiv preprint arXiv:1802.05313*, 2018.

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

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

[9] 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.

[10] Robert Loftin, Bei Peng, James MacGlashan, Michael L Littman, Matthew E Taylor, Jeff Huang, and David L Roberts. Learning something from nothing: Leveraging implicit human feedback strategies. In *The 23rd IEEE international symposium on robot and human interactive communication*, pages 607–612. IEEE, 2014.

[11] 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.

[12] Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pages 6818–6827. PMLR, 2019.

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

[14] Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2020.

[15] 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.

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

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

[18] 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.

[19] 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.

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

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

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

[23] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[24] 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.

[25] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10113–10123, 2021.

[26] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

[27] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. *arXiv preprint arXiv:2102.02400*, 2021.

[28] 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.

[29] Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. Federated bandit: A gossiping approach. In *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, pages 3–4, 2021.

[30] Jiaheng Wei and Yang Liu. When optimizing f-divergence is robust with label noise. In *International Conference on Learning Representations*, 2021.

[31] Yang Liu. Understanding instance-level label noise: Disparate impacts and treatments. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 6725–6735. PMLR, 2021.

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

[33] 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.

[34] 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.

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

[36] 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.

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

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

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

[40] 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.

[41] 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.

[42] 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.

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

[44] 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.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[50] 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.

[51] 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.

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

[53] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[54] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2021.

[55] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *ICLR (Poster)*, 2016.

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

[57] 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.

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

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

[60] 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.

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

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

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

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

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

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

[67] 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.

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

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 7.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] see broader impacts.
   (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]

Broader Impacts

Weak supervision often encodes biases and noise. Our works aim to improve the robustness of policy learning algorithms which is relevant to applications concerning fairness and training data biases. Our solutions are expected to be of interests to machine learning practitioners and researchers who are interested in applications and theory in RL. 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 supervisions is very limiting might lead to unstable performance and unintended consequences, especially when the supervisions are highly noisy.