A PROOFS

A.1 Proof of Lemma 1

We firstly restate the main assumption of our theoretical result:

Assumption 1 (Mixture state-action density)**.** *The state-action density of the learning policy π is a mixture of the state-action densities of the expert and non-expert policies with a mixing coefficient $0 \leqslant \kappa(\pi) \leqslant 1$:*

$$\rho_\pi(\mathbf{x}) = \kappa(\pi)\rho_{\mathrm{E}}(\mathbf{x}) + (1 - \kappa(\pi))\rho_{\mathrm{N}}(\mathbf{x}), \tag{22}$$

where $\rho_\pi(\mathbf{x})$, $\rho_{\mathrm{E}}(\mathbf{x})$, and $\rho_{\mathrm{N}}(\mathbf{x})$ are the state-action densities of the learning policy, the expert policy, and the non-expert policy, respectively.

Under this assumption and the assumption that $\rho'(\mathbf{s}, \mathbf{a}) = \alpha\rho_{\mathrm{E}}(\mathbf{s}, \mathbf{a}) + (1 - \alpha)\rho_{\mathrm{N}}(\mathbf{s}, \mathbf{a})$, we obtain Lemma 1.

Lemma 1. *Letting $\ell_{\mathrm{sym}}(\cdot)$ be a symmetric loss that satisfies $\ell_{\mathrm{sym}}(g(\mathbf{x})) + \ell_{\mathrm{sym}}(-g(\mathbf{x})) = c$, $\forall \mathbf{x} \in \mathcal{X}$ and a constant $c \in \mathbb{R}$, the following equality holds.*

$$\mathcal{R}(g; \rho', \rho_\pi^\lambda, \ell_{\mathrm{sym}}) = (\alpha - \kappa(\pi)(1 - \lambda))\mathcal{R}(g; \rho_{\mathrm{E}}, \rho_{\mathrm{N}}, \ell_{\mathrm{sym}}) + \frac{1 - \alpha + \kappa(\pi)(1 - \lambda)}{2}c. \tag{23}$$

Proof. Firstly, we define $\tilde{\kappa}(\pi, \lambda) = \kappa(\pi)(1 - \lambda)$ and $\delta^\ell(\mathbf{x}) = \ell(g(\mathbf{x})) + \ell(-g(\mathbf{x}))$. Then, we substitute $\rho'(\mathbf{x}) := \alpha\rho_{\mathrm{E}}(\mathbf{x}) + (1 - \alpha)\rho_{\mathrm{N}}(\mathbf{x})$ and $\rho_\pi(\mathbf{x}) = \kappa(\pi)\rho_{\mathrm{E}}(\mathbf{x}) + (1 - \kappa(\pi))\rho_{\mathrm{N}}(\mathbf{x})$ into the risk $\mathcal{R}(g; \rho', \rho_\pi^\lambda, \ell)$.

$$\begin{aligned}
2\mathcal{R}(g; \rho', \rho_\pi^\lambda, \ell) &= \mathbb{E}_{\rho'}\left[\ell(g(\mathbf{x}))\right] + \mathbb{E}_{\rho_\pi^\lambda}\left[\ell(-g(\mathbf{x}))\right] \\
&= \mathbb{E}_{\rho'}\left[\ell(g(\mathbf{x}))\right] + (1 - \lambda)\mathbb{E}_{\rho_\pi}\left[\ell(-g(\mathbf{x}))\right] + \lambda\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] \\
&= \alpha\mathbb{E}_{\rho_{\mathrm{E}}}\left[\ell(g(\mathbf{x}))\right] + (1 - \alpha)\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(g(\mathbf{x}))\right] \\
&\quad + (\kappa(\pi)(1 - \lambda))\mathbb{E}_{\rho_{\mathrm{E}}}\left[\ell(-g(\mathbf{x}))\right] + (1 - \kappa(\pi))(1 - \lambda)\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] + \lambda\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] \\
&= \alpha\mathbb{E}_{\rho_{\mathrm{E}}}\left[\ell(g(\mathbf{x}))\right] + (1 - \alpha)\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(g(\mathbf{x}))\right] + \tilde{\kappa}(\pi, \lambda)\mathbb{E}_{\rho_{\mathrm{E}}}\left[\ell(-g(\mathbf{x}))\right] + (1 - \tilde{\kappa}(\pi, \lambda))\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] \\
&= \alpha\mathbb{E}_{\rho_{\mathrm{E}}}\left[\ell(g(\mathbf{x}))\right] + (1 - \alpha)\mathbb{E}_{\rho_{\mathrm{N}}}\left[\delta^\ell(\mathbf{x}) - \ell(-g(\mathbf{x}))\right] \\
&\quad + \tilde{\kappa}(\pi, \lambda)\mathbb{E}_{\rho_{\mathrm{E}}}\left[\delta^\ell(\mathbf{x}) - \ell(g(\mathbf{x}))\right] + (1 - \tilde{\kappa}(\pi, \lambda))\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] \\
&= \alpha\mathbb{E}_{\rho_{\mathrm{E}}}\left[\ell(g(\mathbf{x}))\right] + (1 - \alpha)\mathbb{E}_{\rho_{\mathrm{N}}}\left[\delta^\ell(\mathbf{x})\right] - \mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] + \alpha\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] \\
&\quad + \tilde{\kappa}(\pi, \lambda)\mathbb{E}_{\rho_{\mathrm{E}}}\left[\delta^\ell(\mathbf{x})\right] - \tilde{\kappa}(\pi, \lambda)\mathbb{E}_{\rho_{\mathrm{E}}}\left[\ell(g(\mathbf{x}))\right] + \mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] - \tilde{\kappa}(\pi, \lambda)\mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right] \\
&= (\alpha - \tilde{\kappa}(\pi, \lambda))\left(\mathbb{E}_{\rho_{\mathrm{E}}}\left[\ell(g(\mathbf{x}))\right] + \mathbb{E}_{\rho_{\mathrm{N}}}\left[\ell(-g(\mathbf{x}))\right]\right) + (1 - \alpha)\mathbb{E}_{\rho_{\mathrm{N}}}\left[\delta^\ell(\mathbf{x})\right] + \tilde{\kappa}(\pi, \lambda)\mathbb{E}_{\rho_{\mathrm{E}}}\left[\delta^\ell(\mathbf{x})\right] \\
&= 2(\alpha - \tilde{\kappa}(\pi, \lambda))\mathcal{R}(g; \rho_{\mathrm{E}}, \rho_{\mathrm{N}}, \ell) + (1 - \alpha)\mathbb{E}_{\rho_{\mathrm{N}}}\left[\delta^\ell(\mathbf{x})\right] + \tilde{\kappa}(\pi, \lambda)\mathbb{E}_{\rho_{\mathrm{E}}}\left[\delta^\ell(\mathbf{x})\right].
\end{aligned} \tag{24}$$

For symmetric loss, we have $\delta^{\ell_{\mathrm{sym}}}(\mathbf{x}) = \ell_{\mathrm{sym}}(g(\mathbf{x})) + \ell_{\mathrm{sym}}(-g(\mathbf{x})) = c$ for a constant $c \in \mathbb{R}$. With this, we can express the left hand-side of Eq. (23) as follows:

$$\begin{aligned}
\mathcal{R}(g; \rho', \rho_\pi^\lambda, \ell_{\mathrm{sym}}) &= (\alpha - \kappa(\pi)(1 - \lambda))\mathcal{R}(g; \rho_{\mathrm{E}}, \rho_{\mathrm{N}}, \ell_{\mathrm{sym}}) + \frac{(1 - \alpha)}{2}\mathbb{E}_{\rho_{\mathrm{N}}}\left[\delta^{\ell_{\mathrm{sym}}}(\mathbf{x})\right] + \frac{\kappa(\pi)(1 - \lambda)}{2}\mathbb{E}_{\rho_{\mathrm{E}}}\left[\delta^{\ell_{\mathrm{sym}}}(\mathbf{x})\right] \\
&= (\alpha - \kappa(\pi)(1 - \lambda))\mathcal{R}(g; \rho_{\mathrm{E}}, \rho_{\mathrm{N}}, \ell_{\mathrm{sym}}) + \frac{(1 - \alpha)}{2}\mathbb{E}_{\rho_{\mathrm{N}}}\left[c\right] + \frac{\kappa(\pi)(1 - \lambda)}{2}\mathbb{E}_{\rho_{\mathrm{E}}}\left[c\right] \\
&= (\alpha - \kappa(\pi)(1 - \lambda))\mathcal{R}(g; \rho_{\mathrm{E}}, \rho_{\mathrm{N}}, \ell_{\mathrm{sym}}) + \frac{1 - \alpha + \kappa(\pi)(1 - \lambda)}{2}c.
\end{aligned} \tag{25}$$

This equality concludes the proof of Lemma 1. Note that this proof follows Charoenphakdee et al. (2019). □

A.2 Proof of Theorem 1

Lemma 1 indicates that, when $\alpha - \kappa(\pi)(1 - \lambda) > 0$, we have

$$\begin{aligned}
g^\star &= \operatorname*{argmin}_g \mathcal{R}(g; \rho', \rho_\pi^\lambda, \ell_{\mathrm{sym}}) \\
&= \operatorname*{argmin}_g \mathcal{R}(g; \rho_{\mathrm{E}}, \rho_{\mathrm{N}}, \ell_{\mathrm{sym}}).
\end{aligned} \tag{26}$$

With this, we obtain Theorem 1 which we restate and prove below.

Theorem 1. *Given the optimal classifier $g^\star$ in Eq. (26), the solution of $\max_\pi \mathcal{R}(g^\star; \rho', \rho_\pi^\lambda, \ell_{\rm sym})$ is equivalent to the expert policy.*

Proof. By using the definitions of the risk and $\rho_\pi^\lambda(\mathbf{x})$, $\mathcal{R}(g^\star; \rho', \rho_\pi^\lambda, \ell_{\rm sym})$ can be expressed as

$$\mathcal{R}(g^\star; \rho', \rho_\pi^\lambda, \ell_{\rm sym}) = \frac{1}{2}\mathbb{E}_{\rho'}[\ell_{\rm sym}(g^\star(\mathbf{x}))] + \frac{\lambda}{2}\mathbb{E}_{\rho_{\rm N}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] + \frac{1-\lambda}{2}\mathbb{E}_{\rho_\pi}[\ell_{\rm sym}(-g^\star(\mathbf{x}))]\,. \tag{27}$$

Since the first and second terms are constant w.r.t. π, the solution of $\max_\pi \mathcal{R}(g^\star; \rho', \rho_\pi^\lambda, \ell_{\rm sym})$ is equivalent to the solution of $\max_\pi \mathbb{E}_{\rho_\pi}[\ell_{\rm sym}(-g^\star(\mathbf{x}))]$, where we omit the positive constant factor $(1-\lambda)/2$. Under Assumption 1 which assumes $\rho_\pi(\mathbf{x}) = \kappa(\pi)\rho_{\rm E}(\mathbf{x}) + (1-\kappa(\pi))\rho_{\rm N}(\mathbf{x})$, we can further express the objective function as

$$\begin{aligned}\mathbb{E}_{\rho_\pi}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] &= \kappa(\pi)\mathbb{E}_{\rho_{\rm E}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] + (1-\kappa(\pi))\mathbb{E}_{\rho_{\rm N}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] \\ &= \kappa(\pi)\Big(\mathbb{E}_{\rho_{\rm E}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] - \mathbb{E}_{\rho_{\rm N}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))]\Big) + \mathbb{E}_{\rho_{\rm N}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))]\,. \end{aligned} \tag{28}$$

The last term is a constant w.r.t. π and can be safely ignored. The right hand-side is maximized by increasing $\kappa(\pi)$ to 1 when the inequality $\mathbb{E}_{\rho_{\rm E}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] - \mathbb{E}_{\rho_{\rm N}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] > 0$ holds. Since $g^\star$ is also the optimal classifier of $\mathcal{R}(g; \rho_{\rm E}, \rho_{\rm N}, \ell_{\rm sym})$, the inequality $\mathbb{E}_{\rho_{\rm E}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] - \mathbb{E}_{\rho_{\rm N}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))] > 0$ holds. Specifically, the expected loss of classifying expert data as non-expert: $\mathbb{E}_{\rho_{\rm E}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))]$, is larger to the expected loss of classifying non-expert data as non-expert: $\mathbb{E}_{\rho_{\rm N}}[\ell_{\rm sym}(-g^\star(\mathbf{x}))]$. Thus, the objective can only be maximized by increasing $\kappa(\pi)$ to 1. Because $\kappa(\pi) = 1$ if and only if $\rho_\pi(\mathbf{x}) = \rho_{\rm E}(\mathbf{x})$, we conclude that the solution of $\max_\pi \mathcal{R}(g^\star; \rho', \rho_\pi^\lambda, \ell_{\rm sym})$ is equivalent to $\pi_{\rm E}$. □

B DATASETS AND IMPLEMENTATION

We conduct experiments on continuous-control benchmarks simulated by PyBullet simulator (Coumans and Bai, 2019). We consider four locomotion tasks, namely `HalfCheetah`, `Hopper`, `Walker2d`, and `Ant`, where the goal is to control the agent to move forward to the right. We use true states of the agents and do not use visual observation. To obtain demonstration datasets, we collect expert and non-expert state-action samples by using 6 policy snapshots trained by the trust-region policy gradient method (ACKTR) (Wu et al., 2017), where each snapshot is obtained using different training samples. The cumulative rewards achieved by the six snapshots are given in Table 2, where snapshot `#1` is used as the expert policy. Visualization of trajectories obtained by the expert policy in these tasks is provided in Figure 7. Sourcecode of our datasets and implementation for reproducing the results is publicly available at `https://github.com/voot-t/ril_co`.

All methods use policy networks with 2 hidden-layers of 64 hyperbolic tangent units. We use similar networks with 100 hyperbolic tangent units for classifiers in RIL-Co and discriminators in other methods. The policy networks are trained by ACKTR, where we use a public implementation (Kostrikov, 2018). In each iteration, the policy collects a total of $B = 640$ transition samples using 32 parallel agents, and we use these transition samples as the dataset $\mathcal{B}$ in Algorithm 1. Throughout the learning process, the total number of transition samples collected by the learning policy is 20 million. For training the classifier and discriminator, we use Adam (Kingma and Ba, 2015) with learning rate 10^{-3} and the gradient penalty regularizer with the regularization parameter of 10 (Gulrajani et al., 2017). The mini-batch size for classifier/discriminator training is 128.

For co-pseudo-labeling in Algorithm 1 of RIL-Co, we initialize by splitting the demonstration dataset $\mathcal{D}$ into two disjoint subset $\mathcal{D}_1$ and $\mathcal{D}_2$. In each training iteration, we draw batch samples $\mathcal{U} = \{\mathbf{x}_u\}_{u=1}^U \sim \mathcal{D}_2$ and $\mathcal{V} = \{\mathbf{x}_v\}_{v=1}^V \sim \mathcal{D}_1$ from the split datasets with $U = V = 640$. To obtain pseudo-labeled datasets $\mathcal{P}_1$ for training classifier g_1, we firstly compute the classification scores $g_2(\mathbf{x}_u)$ using samples in $\mathcal{U}$. Then, we choose $K = 128$ samples with the least negative values of $g_2(\mathbf{x}_u)$ in an ascending order as $\mathcal{P}_1$. We choose samples in this way to incorporate a heuristic that prioritizes choosing negative samples which are predicted with high confidence to be negative by the classifiers, i.e., these samples are far away from the decision boundary. Without this heuristic, obtaining good approximated samples requires using a large batch size which is computationally expensive. The procedure to obtain $\mathcal{P}_2$ is similar, but we use $\mathcal{V}$ instead of $\mathcal{U}$ and $g_1(\mathbf{x}_v)$ instead of $g_2(\mathbf{x}_u)$. The implementation of RIL-P variants in our ablation study is similar, except that we have only one neural networks and we do not split the dataset into disjoint subsets.

For VILD, we the log-sigmoid reward variant and perform important sampling based on the estimated noise, as described by Tangkaratt et al. (2020). For behavior clonig (BC), we use a deterministic policy neural network

Table 2: Cumulative rewards achieved by six policy snapshots used for generating demonstrations. Snapshot 1 is used as the expert policy. Ant (Gaussian) denotes a scenario in the experiment with the Gaussian noise dataset in Section 4.3, where Gaussian noise with different variance is added to expert actions.

Task	Snapshot #1	Snapshot #2	Snapshot #3	Snapshot #4	Snapshot #5	Snapshot #6
`HalfCheetah`	2500	1300	1000	700	-1100	-1000
`Hopper`	2300	1100	1000	900	600	0
`Walker2D`	2700	800	600	700	100	0
`Ant`	3500	1400	1000	700	400	0
`Ant` (Gaussian)	3500	1500	1000	800	500	400

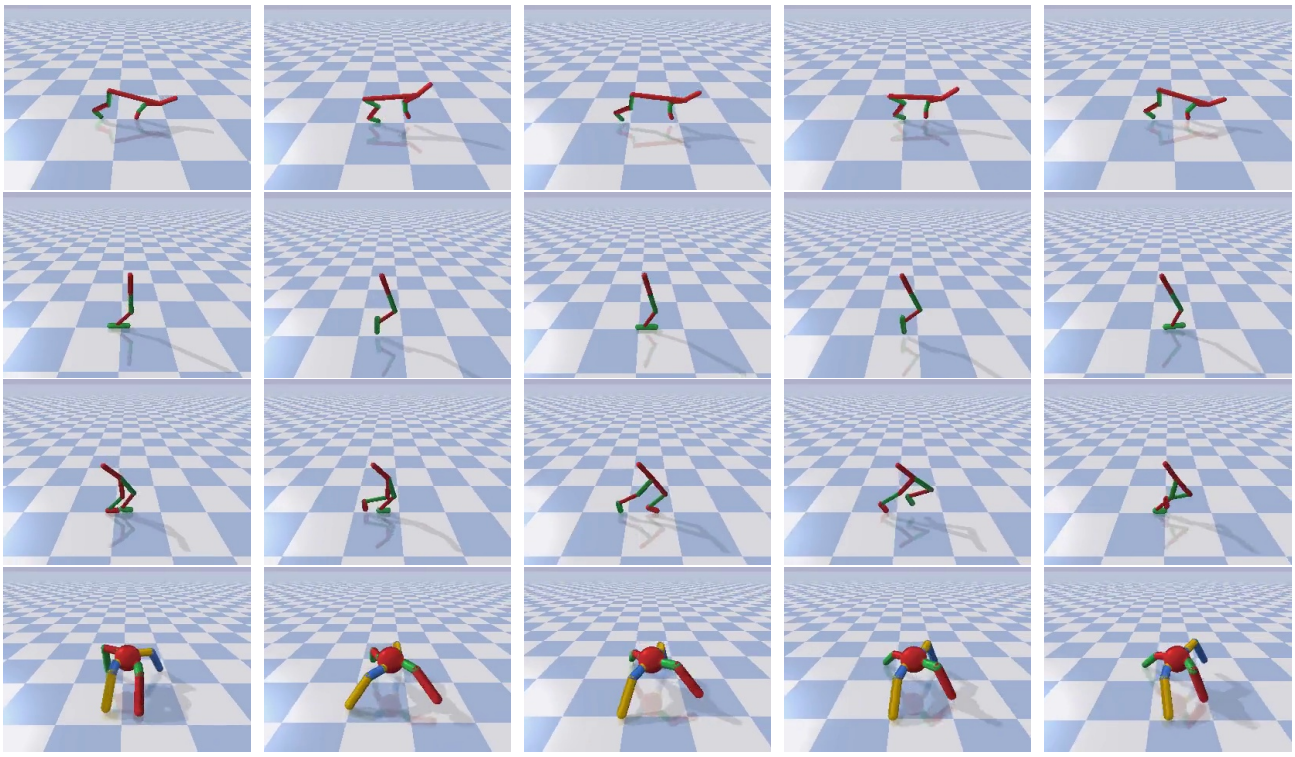


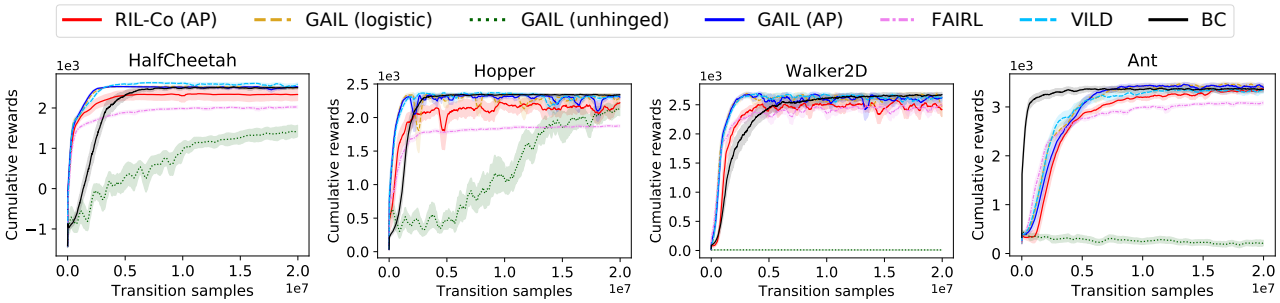
Figure 7: Visualization of the first 100 time steps of expert demonstrations. Time step increases from the leftmost figure ($t = 0$) to the rightmost figure ($t = 100$). Videos are provided with the sourcecode.

and train it by minimizing the mean-squared-error with Adam and learning rate 10^{-3}. We do not apply a regularization technique for BC. For the other methods, we follow the original implementation as close as possible, where we make sure that these methods perform well overall on datasets without noise.

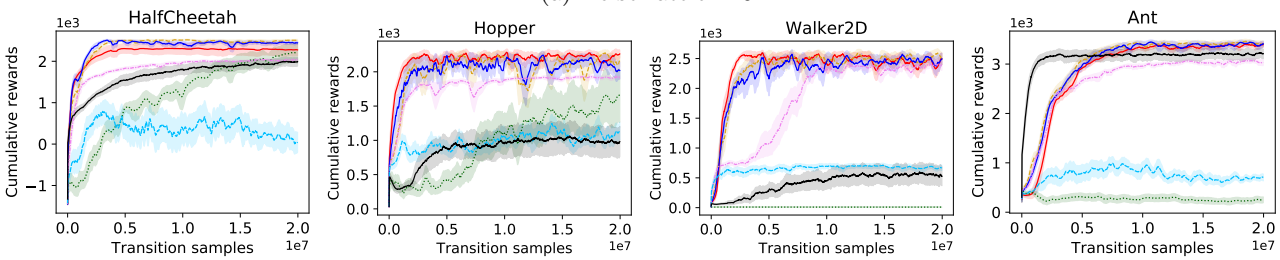
C ADDITIONAL RESULTS

Here, we present learning curves obtained by each method in the experiments. Figure 8 depicts learning curves (performance against the number of transition samples collected by the learning policy) for the results in Section 4.1. Since BC does not use the learning policy to collect transition samples, the horizontal axes for BC denote the number of training iterations. It can be seen that RIL-Co achieves better performances and uses less transition samples compared to other methods in the high-noise scenarios where $\delta \in \{0.2, 0.3, 0.4\}$. Meanwhile in the low-noise scenarios, all methods except GAIL with the unhinged loss and VILD perform comparable to each other in terms of the performance and sample efficiency.

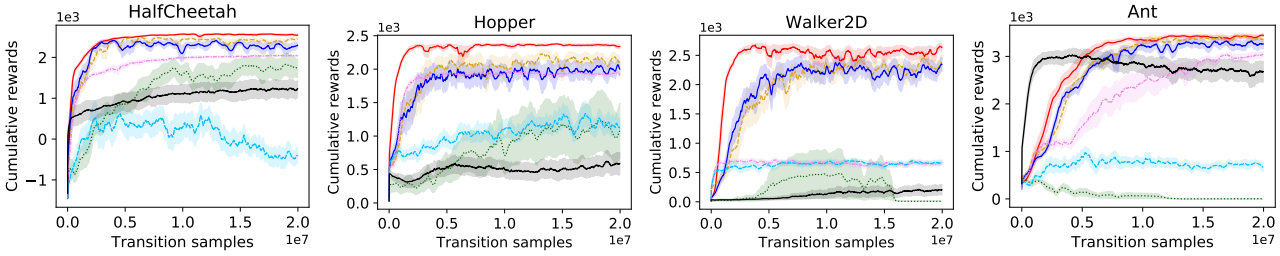
Figure 9 shows learning curves of the ablation study in Section 4.2. RIL-Co with the AP loss clearly outperforms the comparison methods in terms of both sample efficiency and final performance. The final performance in Figures 2 and 5 are obtained by averaging the performance in the last 1000 iterations of the learning curves.



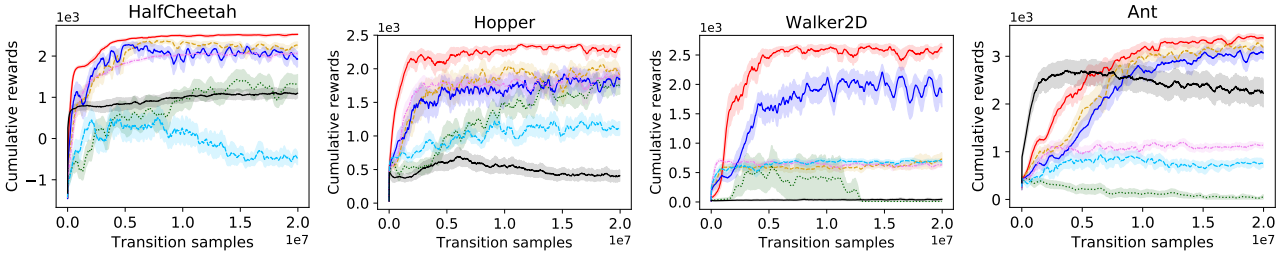
(a) Noise rate $\delta = 0$



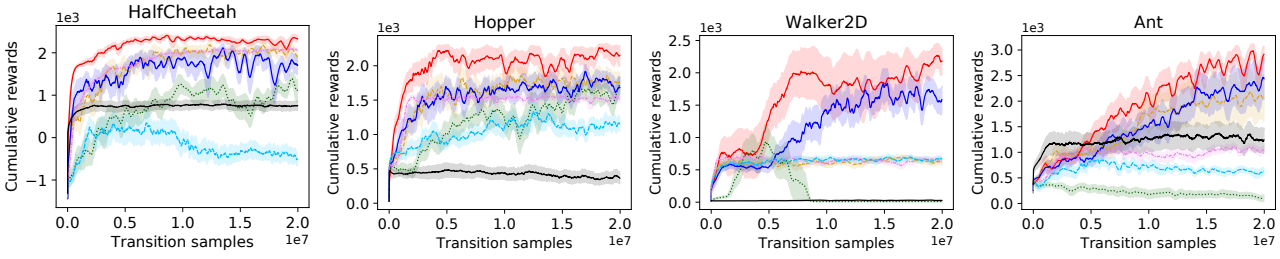
(b) Noise rate $\delta = 0.1$



(c) Noise rate $\delta = 0.2$

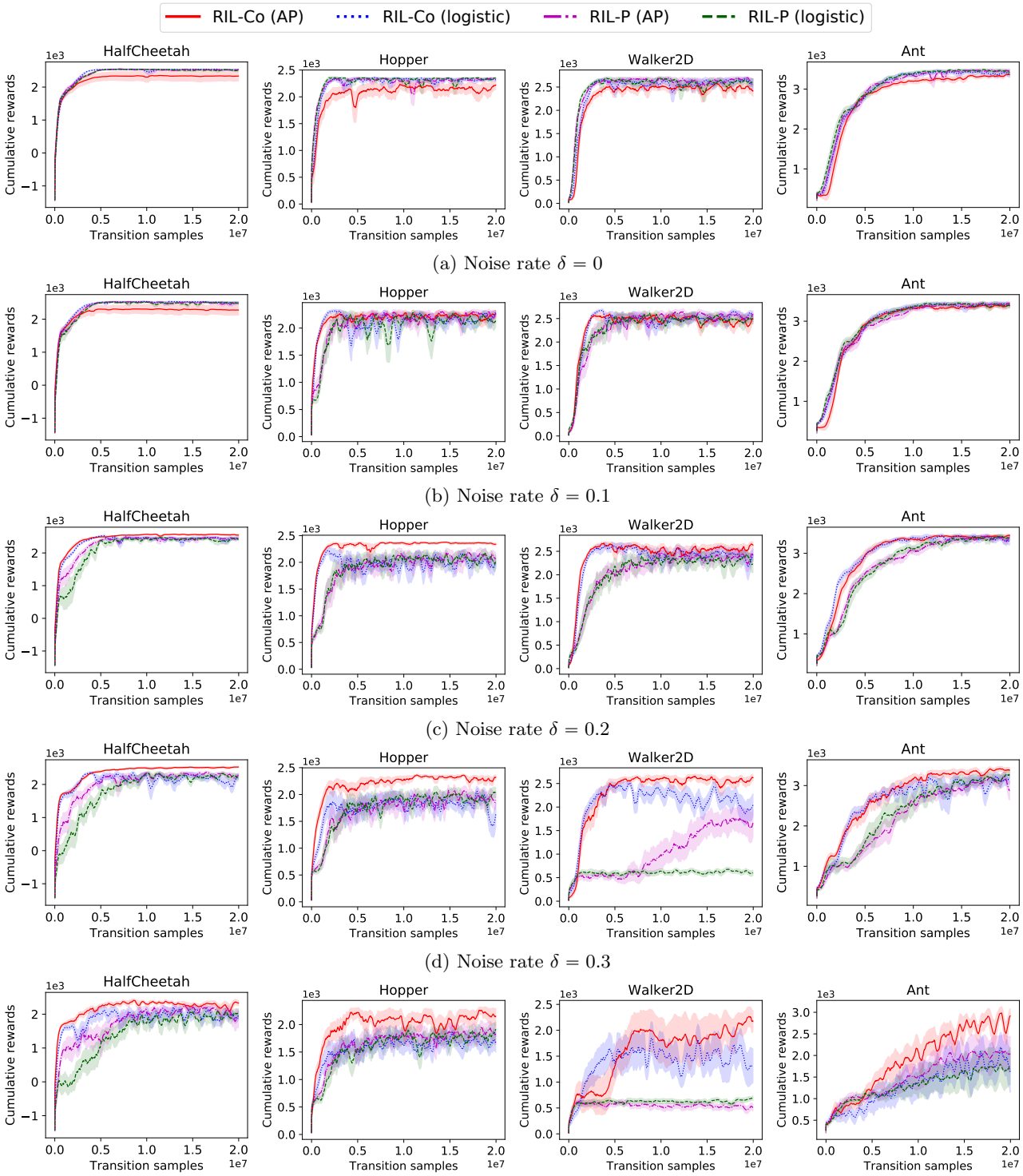


(d) Noise rate $\delta = 0.3$



(e) Noise rate $\delta = 0.4$

Figure 8: Performance against the number of transition samples in continuous-control benchmarks. For BC, the horizontal axes denote the number of training iterations. Clearly, RIL-Co with the AP loss is more robust than comparison methods when the noise rate increases.



(a) Noise rate $\delta = 0$

(b) Noise rate $\delta = 0.1$

(c) Noise rate $\delta = 0.2$

(d) Noise rate $\delta = 0.3$

(e) Noise rate $\delta = 0.4$

Figure 9: Performance against the number of transition samples in the ablation study. RIL-Co with the AP loss is more robust than its variants that use non-symmetric losses and naive pseudo-labeling.