REWARD-FREE POLICY LEARNING THROUGH ACTIVE HUMAN INVOLVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the success of reinforcement learning (RL) in many control tasks, the behaviors of the learned agents are largely limited by the hand-crafted reward function in the environment, which might not truthfully reflect human intents and preferences. This work proposes a reward-free policy learning method called Proxy Value Propagation that conveys human intents explicitly to the learning policy through active involvement. We adopt an interactive learning setting where human subjects can actively intervene and demonstrate to the agent. Our key insight is that a latent value function can be learned from active human involvement, which in return guides the learning policy to emulate human behaviors. The proposed method first relabels and propagates the proxy values of human demonstrations to other states, and then optimizes the policies to comply with the human intents expressed through the proxy value function. The proposed method can be incorporated into many existing RL algorithms with minimum modifications. Experiments on various tasks and human control devices demonstrate the generality and efficiency of our method. Theoretic guarantee on the learning safety is also provided. Demo video and code are available in the supplementary material.

1 INTRODUCTION

Reinforcement learning (RL) has been successfully applied in many domains, ranging from board game Go (Silver et al., 2016), strategy game StarCraft II (Samvelyan et al., 2019), autonomous driving (Kendall et al., 2019), and even nuclear fusion (Degrave et al., 2022). Existing RL methods assume the manually designed reward function can fully express human intents and goals, however, the resulting agents might exhibit biased, misguided, or undesired behaviors due to faulty reward functions (Leike et al., 2018; Russell, 2019; Krakovna et al., 2020). Moreover, the poor sample efficiency as well as the safety concern due to the trial-and-error exploration prevent the real-world deployment of RL.

Human-in-the-loop policy learning (HL) methods are promising complements to RL and reward engineering, which relies on human subject to oversee the learning process of the robots and autonomous agents. HL methods can learn complex and safety-aware behaviors that are intractable to be encoded in the handcrafted reward function. Different forms of human involvement have been studied over the years. Human subjects can advise actions upon the requests of the robots (Mandel et al., 2017) or provide preference-based feedback to assess the relative value of the collected trajectories (Wirth et al., 2017; Christiano et al., 2017; Reddy et al., 2018; Warnell et al., 2018; Palan et al., 2019; Guan et al., 2021). These methods learn from passive human involvement, where the humans do not provide real-time feedback and intervention during data collection, reducing the efficiency of the human-robot systems. On the other hand, an increasing body of works focuses on active human involvement, where human subjects are authorized to actively intervene and provide demonstrations during the execution of robots (Kelly et al., 2019; Spencer et al., 2020; Mandlekar et al., 2020; Li et al., 2022b). With online corrective demonstrations from human subjects, the safety and learning efficiency of the system can be greatly improved.

In this work, we explore transforming a standard RL task into a reward-free setting where the agent can learn from active human involvement. Our intuition is that the agent should either replicate the behaviors of human subjects in the circumstances where humans are once involved and provide demonstrations, or should apply the actions that can move toward those human-involved states where

high-quality demonstrations are available. We find that this intuition can be instantiated into many existing RL algorithms, by **casting active human involvement as human subjects depicting the landscape of a latent value function**. We design a method called *Proxy Value Propagation (PVP)* that can be incorporated into common value-based RL methods. PVP assigns high Q values to human actions and low values to agent actions that are intervened by human subjects. RL policy thus tends to replicate human actions due to the value-maximizing nature. The proxy values will be propagated to prior states through TD learning, informing that human demonstrations are available in these human-involved states. Experiments show that PVP can be successfully applied to both continuous-action and discrete-action tasks, and achieve higher learning efficiency compared to baselines in solving grid-world navigation, Atari video game, and driving tasks. It is also compatible with various forms of human input devices, including gamepad, driving wheel, and keyboard. We summarize our main contributions as follows:

1) We show that active human involvement is an effective supervision to train the agent and theoretically prove the safety guarantee in training and testing time under this setting.
2) We propose a simple yet effective method Proxy Value Propagation that can be easily integrated in existing RL algorithms to learn from active human involvement. Our method can be generalized across various task settings and human control devices.
3) The experiments show that the proposed PVP method enables safety guarantees and high training efficiency. Compared to previous HL baselines, PVP achieves better performance with less cost of human involvement.

2 Related Work

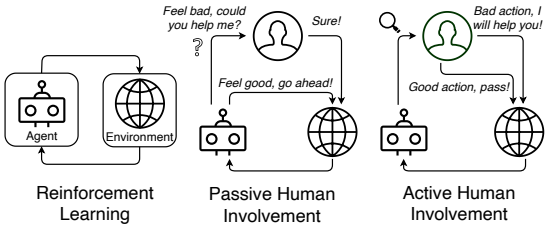


Figure 1: Different policy learning approaches.

As shown in the left panel of Fig. 1, reinforcement learning (RL) method learns from the interactions between agent and environment. The source of supervision comes from the reward function. However, Russell (2019) challenges the existing RL methods that it is hard to comprehensively represent human preferences into a scalar reward function (Dafoe et al., 2020). The manually designed reward function, which might be misaligned with human intent and preference, often leads to undesired agent behaviors (Leike et al., 2018; Krakovna et al., 2020). As a promising complement to RL, Human-in-the-loop Learning (HL) can overcome costly reward engineering and convey human intent and preference to the learning process directly through human involvement.

HL with Passive Human Involvement. As shown in the middle panel of Fig. 1, many works learn from passive human involvement. The neural policy operates the robot and the human subjects can provide demonstration directly upon request of the learning agents (Mandel et al., 2017; Menda et al., 2019; Jonnavittula & Losey, 2021). This exposes human subjects to potential risks since they do not fully control the system. Alternatively, human subjects can evaluate the collected trajectories after the agent-environment interaction (Christiano et al., 2017; Guan et al., 2021; Reddy et al., 2018; Warnell et al., 2018; Sadigh et al., 2017; Palan et al., 2019). These methods can be applied to the tasks that human can not conduct, such as moving a six-legged Ant robot via providing exact torque at each joint (Christiano et al., 2017). However, for those tasks that human can demonstrate, these methods are unable to have real-time feedback from human subject during agent-environment interaction.

HL with Active Human Involvement. As shown in the right panel of Fig. 1, human subjects can proactively participate based on their own judgment. There are many works that utilize the idea of learning from advice. TAMER framework (Knox & Stone, 2012) learns policy from human-provided evaluative feedback (human-generated reward). The Task-Instruction-Contingency-Shaping (TISC) method (Najar et al., 2020) also uses evaluative feedback for accelerating the learning. Evaluative feedback is a boolean criticizing correct/wrong. This is similar to the intervention in our framework. However, in TICS, humans provide high-level instructions, e.g. pointing to the left/right, while in PVP humans provide intervention and low-level demonstrations. TICS directly modifies the policy action distribution based on human feedback, while PVP uses a proxy value to indirectly encourage agents to follow human preference. Other works allow human subjects to decide to terminate the episode if a near-accidental situation happens (Zhang & Cho, 2017; Abel et al., 2017; Saunders

et al., 2018; Xu et al., 2022). Recent studies explore active human involvement methods through intervention and demonstration in the human-robot shared autonomy (Menda et al., 2019; Kelly et al., 2019; Spencer et al., 2020; Li et al., 2022b; Jonnavittula & Losey, 2021; Xu et al., 2022). However, previous methods do not fully utilize the power of human involvement. Kelly et al. (2019) do not leverage data collected by agents, while Mandlekar et al. (2020) do not suppress undesired actions likely intervened by human. Meanwhile, Spencer et al. (2020) and Mandlekar et al. (2020) focus on optimizing actions step-wise without considering the temporal consistency between steps. These drawbacks harm the learning efficiency and thus incur more human involvement. Moreover, previous methods lack experiments to evaluate the generalizability to different task settings and human control devices.

Another challenge for RL methods is to learn from the data collected from human-robot shared control since the data is drastically off-policy with significant distribution shift. One solution is to frame active human involvement as a form of interactive imitation learning, where the human demonstration is used to train the agent and the data generated by the learning agent is discarded (Mandlekar et al., 2020; Kelly et al., 2019). The information in the agent exploration, *e.g.* the latent model of state transitions (Levine et al., 2020), is lost. Alternative is to use the intervention signals to train an intervention predictor to block dangerous actions and moderate the stress of human subjects (Kelly et al., 2019; Abel et al., 2017; Saunders et al., 2018). Reward learning can be used to extract knowledge from human demonstration and intervention (Ibarz et al., 2018; Wang et al., 2018). Offline RL method is used to learn from human-robot mixed data (Peng et al., 2021; Li et al., 2022b). In this work, we build upon the existing RL framework and relabel the data collected from shared control with proxy values, connecting RL with active human involvement.

3 Problem Formulation

Policy learning aims at finding a policy to solve the sequential decision-making task, which is usually modeled by a Markov decision process (MDP). MDP is defined by the tuple $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, d_0 \rangle$ consisting of a state space $\mathcal{S}$, an action space $\mathcal{A}$, a state transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, a reward function $R : \mathcal{S} \times \mathcal{A} \to [R_{\min}, R_{\max}]$, a discount factor $\gamma \in (0, 1)$, and an initial state distribution $d_0 : \mathcal{S} \to [0, 1]$. The goal of conventional reinforcement learning is to learn a *novice policy* $\pi_n(a|s) : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that can maximize the expected cumulative return: $\pi_n = \arg\max_{\pi_n} \mathbb{E}_{\tau \sim P_{\pi_n}}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)]$, wherein $\tau = (s_0, a_0, ..., s_T, a_T)$ is the trajectory sampled from trajectory distribution P_{π_n} induced by π_n, d_0 and $\mathcal{P}$. Here π_n defines a stochastic policy, while deterministic policy can be denoted as $\mu_n(s) : \mathcal{S} \to \mathcal{A}$ and its action distribution can be described by a Dirac delta distribution $\pi_n(a|s) = \delta(a - \mu_n(s))$.

The reward function imposes an assumption that the reward can fully reflect the intentions of the users and incentivize the desired behaviors of the agents. However, the assumption may not always hold and the learned agent may obtain biased behaviors or figure out the loophole to finish the task (Russell, 2019; Leike et al., 2018). Revisiting the primal goal when developing learning systems, we find the reward is not a necessity since what we really want to achieve is the realization of human preference in the learned behaviors and, as suggested by Russell (2019), **the ultimate source of information about human preferences is human behavior**.

We thus study the reward-free setting and incorporate real human subjects into the training loop. During training, a human subject accompanies the novice policy and can intervene the agent by taking over the control to demonstrate correct behaviors. As discussed in Sec. 4.3, we show that such active human involvement can ensure the safety of human-robot system. We formulate this setting by assuming human subject has a *human policy* $\pi_h(a_h|s) : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which outputs human action $a_h \in \mathcal{A}$ at each time step. We model the human subjects' intervention behaviors by *intervention policy* $I(\cdot|s, a_n)$. In earlier methods such as DAgger (Ross & Bagnell, 2010), the intervention policy is a Bernoulli distribution and the control authority switches back and forth between the novice and the expert. Later studies allow the human subjects to intervene and take full control (Wang et al., 2018; Xu et al., 2022; Saunders et al., 2018; Li et al., 2022b). In this case the intervention policy can be considered as a deterministic policy denoted by $I(s, a_n) : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$ where $a_n \sim \pi_n(\cdot|s)$ is agent's action. With the intervention policy, we can summarize the *behavior policy* π_b that generates

actions applied to the environment as:

$$\pi_b(a|s) = (1 - I(s,a))\pi_n(a|s) + \pi_h(a|s) \int_{a' \in \mathcal{A}} I(s,a')\pi_n(a'|s)da' \quad \text{(Stochastic novice)} \quad (1)$$

$$\pi_b(a|s) = (1 - I(s, \mu_n(s)))\delta(a - \mu_n(s)) + I(s, \mu_n(s))\pi_h(a|s) \quad \text{(Deterministic novice)} \quad (2)$$

With such a model of active human involvement, we can now formulate our goal. Recall that our primal goal is to find the novice agents whose behaviors are compliant to human preferences. To quantitatively measure the extent where agents accomplish this, we consider two metrics and thus form a twofold problem.

On one hand, task-specified metrics are available in our experiments since we need to evaluate the performance of the learned agent when they are running independently without human involvement in testing time. These metrics can be the success rate in navigation or the scores in games. Though agents do not have access to the metrics, human subjects are aware of how these metrics are computed and the correspondence between the metrics and the success of the tasks. Therefore, we can use these metrics as the measurements to evaluate the realization of human preference:

Problem 1 (Preference Compliance)**.** Find a novice policy that maximizes the underlying task-specified metric U: $\pi_n^* = \arg\max_{\pi_n} \mathbb{E}_{\tau \sim P_{\pi_n}}[U(\tau)]$.

Note that the task-specified metric is not accessible to the learning agent and thus can not be optimized directly. Instead, human intervention $I(s,a)$ and demonstration $a_h \sim \pi_h(\cdot|s)$ are the only sources of supervision in our current reward-free setting.

In active human involvement, human subjects can intervene at any time at their discretion. The most common cases of human involvement are the near-accidental situations. It is also possible for the human subjects to intervene if the agents perform poorly. Inversely, if the human subjects do not involve, then current states and agent's actions are deemed to be human-compatible. Therefore, we summarize another problem that need to be resolved by our method:

Problem 2 (Involvement Minimization)**.** Find a novice policy that minimizes the human involvement: $\pi_n = \arg\min_{\pi_n} \mathbb{E}_{\tau \sim P_{\pi_b}}[I(s,a)]$.

In next section, we will discuss our insights and how we build a concise, general, and efficient learning method to address the aforementioned problems.

4 METHOD

We propose the *Proxy Value Propagation (PVP)* method to tackle the problems discussed in the previous section. PVP turns a value-based RL method into an efficient human-in-the-loop policy learning method that learns from active human involvement. PVP is compatible with various task settings such as continuous and discrete action spaces and various human control devices. In this section, we first recap the basic workflow of value-based RL method. Based on that, we introduce the motivation and the design of PVP. Finally, we describe the implementation details.

Value-based Reinforcement Learning. In the conventional RL method, the state-action value and state value of policy π are defined as $Q(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$ and $V(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} Q(s,a)$, respectively. The optimal policy is expected to maximize the cumulative return $J(\pi) = \mathbb{E}_{s \sim d_0} V(s)$. A neural network is commonly used to estimate the value function with Bellman backup: $Q(s,a) \leftarrow R(s,a) + \gamma \max_{a'} Q(s',a')$, where s' is the next state. To learn the value network, stochastic gradient descent on the temporal difference (TD) loss is conducted:

$$J^{\text{TD}}(Q) = \mathop{\mathbb{E}}_{(s,a,s')} |Q(s,a) - (R(s,a) + \gamma \max_{a'} Q(s',a'))|^2. \quad (3)$$

If we discard the instant reward, the Q value is optimized to fit $Q(s,a) \leftarrow \gamma \max_{a'} Q(s',a')$. Based on the learned value function, the policy π_θ parameterized by θ can be learned by optimizing

$$\theta = \arg\max_{\theta} \mathop{\mathbb{E}}_{\tau \sim P_{\pi_\theta}} [Q(s,a)]. \quad (4)$$

4.1 PROXY VALUE PROPAGATION

The major challenge to efficiently learn from active human involvement is the scarcity of information. To address the issue, we introduce a key observation and the goals behind the design of our method. The observation is that when a human subject intervenes in the human-robot shared control process, it leads to a clear signal that human subject is not satisfied with the current performance of the agent, either due to its unsafe actions or poorly performing behaviors. Based on this, we can summarize that *the learned policy should (1) replicate the behaviors demonstrated by the human subject and (2) avoid performing actions that are once intervened by the human subject.*

The requirement can be fulfilled by the value-based RL framework. A deeper look at Eq. 4 suggests an optimal deterministic policy will always choose actions with the largest Q values. Therefore we can manipulate the Q values to induce desired behaviors: If human subject intervenes at some states, the human actions $a_h \sim \pi_h$ should always have higher values than other actions in these states. At the same time, the agent actions $a_n \sim \pi_n$ should always have lower values than others since they are rejected by human subject.

To implement this idea, we overwrite the Q value of the human action a_h during human intervention to be $+1$ and the novice action a_n to be -1.

$$J^{\text{PV}}(Q) = \mathop{\mathbb{E}}_{(s,a_n,a_h)}[|Q(s,a_h)-1|^2 + |Q(s,a_n)-(-1)|^2]I(s,a_n). \tag{5}$$

To propagate the information of those desired actions provided by human subjects, we combine TD loss to update Q network:

$$J(Q) = J^{\text{PV}}(Q) + J^{\text{TD}}(Q). \tag{6}$$

4.2 IMPLEMENTATION DETAILS

Base RL Methods. Our method can be implemented for both continuous and discrete action spaces. In continuous action space, we extend TD3 (Fujimoto et al., 2018) with the balanced buffer and PV loss. In discrete action space, we use DQN (Mnih et al., 2015). In order to verify our idea in a minimalist and concise framework, though compatible, we do not apply advanced techniques to DQN such as prioritized replay buffer (Schaul et al., 2015), double Q learning (Van Hasselt et al., 2016) and ensemble method (Osband et al., 2016). While TD3 uses deterministic policy, DQN adopts epsilon-greedy exploration that makes the policy stochastic. We remove the action noise in DQN and simply follow the argmax rule to select actions: $\mu^*(s) = \arg\max_a Q(s,a)$. The primary reason is that according to the feedback of human subjects, stochastic agents make human subjects experience excessive fatigue since it is hard to monitor and correct agents' noisy actions.

Balanced Buffers. The intervention gradually becomes sparse as the agent learns to reduce human intervention. However, those sparse intervention signals contain even more important information on how to behave under critical situations. Previous method (Li et al., 2022b) stores agent data and human data into one buffer and samples them uniformly. The human demonstrations are overwhelmed by the amount of agent-generated trajectories, leading to inefficient learning of critical human behaviors. For example, the driving policy sometimes fails to master acceleration at the beginning of an episode, even though the human subject has already demonstrated the expected maneuver multiple times. This is because the demonstration of initial acceleration only lasts a short period of time and thus is scarce in the buffer.

To address this issue, we utilize two replay buffers to store transitions with or without human intervention. The human buffer $\mathcal{B}_h = \{(s, a_n, a_h, s')\}$ stores data during human involvement and the novice buffer $\mathcal{B}_n = \{(s, a_n, s')\}$ stores the transitions during agent's independent exploration. In each training iteration, we sample two equally-sized batches b_n, b_h from $\mathcal{B}_n$, $\mathcal{B}_h$ respectively, each has $N/2$ samples. We compute PV loss J^{PV} use b_h and concatenate b_n and b_h to form a batch with N transitions to compute the TD loss J^{TD}. N is the hyper-parameter batch size. With such design, our method can balance the data from human and from agent. In the initial acceleration example above, as training goes, the agent gradually masters driving skills, *e.g.* driving on a straight road or turning. After that, human subject will only demonstrate the initial acceleration that the agent does not yet learn. Those demonstrations will gradually dominate the human buffer until agent masters them.

(a) MetaDrive (b) CARLA (c) MiniGrid (d) Atari

Figure 2: Training environments and human interfaces.

4.3 Human Intent Compliance

Here we provide a conceptual framework to describe the compliance of human intention. We introduce a simple theorem showing that, in the setting of active human involvement, the compliance of human intention can be grounded by the human policy performance, the accuracy of human intervention and the overall intervention rate.

Theorem 1 (Upper bound of the training risk)**.** *Let $C : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$ be the ground-truth indicator of the intention violation i.e. safety violation, saying whether the action is undesired. Denote the upper bound of human action error rate as $\mathbb{E}_{a\sim\pi_h(\cdot|s)} C(s, a) \leq \epsilon$, the upper bound of human intervention error rate as $\mathbb{E}_{s\sim P_{\pi_b}, a\sim\pi_n(\cdot|s)}(1 - I(s, a))C(s, a) \leq \kappa$ and the intervention rate as $\psi = \mathbb{E}_{s\sim P_{\pi_b}} I(s, a_n)$. The discounted intent violation $S_{\pi_b}(s_0) = \mathbb{E}_{\tau\sim P_{\pi_b}} \sum_{t=0} \gamma^t C(s_t, a_t)$ of the behavior policy π_b is bounded by*

$$S_{\pi_b} \leq \frac{1}{1-\gamma}(\kappa + \epsilon\psi). \tag{7}$$

The proof and detailed explanation is provided in Appendix D.

5 Experiments

5.1 Experimental Setting

Tasks. We conduct experiments on various control tasks with different observation and action spaces. For continuous action space, we use two driving environments, MetaDrive safety benchmark (Li et al., 2022a) and CARLA Town01 environment (Dosovitskiy et al., 2017). In both tasks, the agent needs to steer the target vehicle with low-level acceleration, brake, and steering, to reach the destination. We use sensory state vector in MetaDrive and the bird-eye view image in CARLA as observation. Note that for MetaDrive task, there exists a split of training and test environments and we present the performance of the learned agent in a held-out test environment. For discrete action space, we use MiniGrid Two Room task (Chevalier-Boisvert et al., 2018) and Atari Skiing game (Bellemare et al., 2013). The Two Room is a task requiring heavy exploration since the agent needs to move toward a door and open the door before reaching the destination. The observation of MiniGrid is the semantic map of agent's local neighborhood. The Atari game is difficult since the agent needs to learn to output meaningful action based on the RGB observation. MLP is used for MetaDrive task and CNN is used for others as the feature extractors. Please refer to Sec. E in Appendix for more information.

Human Interfaces. To examine the generalizability of our method, we leverage multiple control devices: Xbox Wireless Controller (Gamepad), keyboard and Logitech G29 Racing Wheel. We denote the MetaDrive tasks with three devices as MetaDrive-Gamepad/Keyboard/Wheel. CARLA task uses Wheel only. In MiniGrid and Atari tasks, we use keyboard to provide discrete control signals. In all tasks, an key in the device is configured to indicate emergency stop. If any discomfort happens, human subjects can pause or stop the experiment immediately. Ethics statement is provided in Appendix A. As shown in Fig. 2, human subjects can intervene and takeover through control devices and the visualization of environments is shown on the screen for monitoring.

Experimental Details. We implement most of the code with Stable-Baselines3 (Raffin et al., 2021). Training results of various baselines in MetaDrive task are obtained from the open-source code by (Li et al., 2022b). The pure RL baselines are repeated 5 times with different random seeds, while other human-in-the-loop methods are repeated fewer times due to limited human resources. In the training

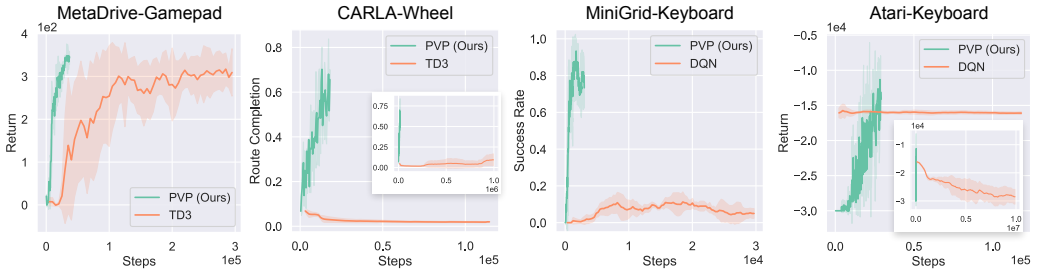


Figure 3: Our method achieves superior sample efficiency compared to the RL counterpart.

Table 1: Comparison of different approaches in MetaDrive-Keyboard. The overall intervention rate is given besides the human data usage.

Method	Training			Testing		
	Human Data Usage	Total Data Usage	Total Safety Cost	Episodic Return	Episodic Safety Cost	Success Rate
SAC	-	1M	2.76K ± 0.95K	386.77 ±35.1	0.73 ±1.18	0.82 ±0.18
PPO	-	1M	24.34K ±3.56K	335.39 ±12.41	3.41 ±1.11	0.69±0.08
TD3	-	1M	1.74K ± 0.62K	318.12 ±21.9	0.47 ±0.08	0.70 ±0.09
SAC-Lag	-	1M	1.84K ± 0.49K	351.96 ±101.88	0.72 ±0.49	0.73 ±0.29
PPO-Lag	-	1M	11.64K ± 4.16K	299.99 ±49.46	1.18 ±0.83	0.51 ±0.17
CPO	-	1M	4.36K ±2.22K	194.06 ±108.86	1.71 ±1.02	0.21 ±0.29
Human Demo.	30K	-	39	347.523	0.39	0.97
BC	30K (1.0)	30K	-	113.323	2.171	0.073
HG-Dagger	39.0K (0.76)	51K	56	116.393	1.979	0.045
IWR	35.8K (0.79)	45K	52	226.221	1.457	0.465
HACO	19.2K (0.48)	40K	130	143.287	1.645	0.139
PVP w/o TD Learning	13.5K (0.34)	40.5K	70	252.447	1.277	0.220
PVP w/ Reward	12.8K (0.32)	40K	30	319.383	0.767	0.755
PVP (Ours)	14.6K (0.37)	40K	76	353.636	0.898	0.857

of the human-in-the-loop method, we need real human participation and we do not use any simulated user input. During testing, there is no any form of human involvement. We run the trained agents in the environment for multiple runs and report the average task-specified metrics in the tables. For each run, since we have many checkpoints during training, we will report the metrics of the checkpoint that achieves the highest $\mathbb{E}_{\tau \sim P_{\pi_n}} U(\tau)$ as the performance of this run in the tables. One human subject participates in each experiment. Experiments with humans are conducted on a local computer with an Nvidia GeForce RTX 3080. We provide the standard deviation if the experiments are repeated multiple runs in tables and figures. Other hyper-parameters are given in the Appendix G.

Baselines. We test four native RL baselines: PPO (Schulman et al., 2017), SAC (Haarnoja et al., 2018), TD3 (Fujimoto et al., 2018) and DQN (Mnih et al., 2015). RL baselines can access environmental reward. In MetaDrive Safety Benchmark, -1 penalty will be added to reward when a crash happens. Three safe RL baselines Constraint Policy Optimization (CPO) (Achiam et al., 2017), PPO-Lagrangian (Stooke et al., 2020), SAC-Lagrangian (Ha et al., 2020) are evaluated. In all baselines above, the reward function and cost function (for Safety Benchmark) are defined by the environment and can be accessed by learning algorithms.

On the other hand, using human-generated dataset in MetaDrive, provided by (Li et al., 2022b), we evaluate IL methods Behavior Cloning, GAIL (Ho & Ermon, 2016) and offline RL method CQL (Kumar et al., 2020). We also run many human-the-loop methods that learn from active human involvement. Human-Gated DAgger (HG-DAgger) (Kelly et al., 2019), Intervention Weighted Regression (IWR) (Mandlekar et al., 2020) and Human-AI Copilot Optimization (HACO) (Li et al., 2022b) are tested.

Demo Video and **Code** are included in the supplementary materials.

Table 2: Results of different approaches in CARLA.

Method	Data Usage	Episodic Return	Route Completion	Success Rate
PPO	1M	81.57 ± 4.935	0.24 ± 0.013	0.0 ± 0.0
TD3	1M	43.46 ± 12.83	0.11 ± 0.05	0.0 ± 0.0
HACO	23K	120.53	0.25	0.11
PVP w/o Balanced Buffers	28.8K	263.82	0.51	0.2
PVP w/ Reward	23.2K	580.125	0.793	0.533
PVP w/ Stochastic Policy	18K	200.18	0.39	0.13
PVP (Ours)	18K	449.65 ± 52.04	0.76 ± 0.07	0.49 ± 0.04

5.2 Baseline Comparison

Comparing with RL Counterparts. Fig. 3 shows the curves of test-time performance of the agents during training. That is, each point in the curves represents the task-specified measurement of the agent saved at that stage of training, performing actions independently without human involvement. In MetaDrive task, our method achieves 350 return in 37K steps. This takes about one hour in the real-world HL experiment. TD3 baseline fails to achieve comparable results even after 300K steps of training. In CARLA Town01, agents learn to drive within 18K steps (30 minutes) with our method, while TD3 can not solve the task. In MiniGrid tasks, our method successfully solves the tasks while vanilla DQN fails, showing that PVP can learn a solution without complex exploration strategies. We also show experiments on an easier and a harder MiniGrid environments in Appendix F.3, where PVP greatly improves the learning efficiency. In Atari task, our method does not demonstrate impressive improvement. We hypothesize this is caused by the difficulty in representation learning. As a comparison, Ibarz et al. (2018) use 6,800 human-labeled episodes, containing 50M environment interactions, to learn a reward function and solve the Atari game. Active human involvement setting generates insufficient data for learning a good representation.

Comparing with Human-in-the-loop Baselines. To ensure a fair comparison, we conduct human-in-the-loop experiments with the same group of human subjects to avoid the performance difference caused by personal difference. To quantitatively assess the experience of human subjects, we report the *safety violation* and *cognitive cost*. Safety violation is measured by the number of crashes happening during the training in MetaDrive. This value reflects how many potential damages the human subject might suffer in training. The cognitive cost is measured by the total amount of human involvement (*Human Data Usage*) and the ratio of human data over total sampled data (*Overall Intervention Rate*). It shows how much effort humans need to pay to teach and protect the agents. As shown in Table 1, all HL methods we test achieve extremely low safety violations in training compared to RL methods, empirically supporting the safety guarantee of the active human involvement in Sec. 4.3. Compared to other human-in-the-loop methods, our method costs the lowest human efforts in terms of human data usage and overall intervention rate, while greatly outperforms baselines in testing performance. Since we are in MetaDrive task with training and test set split, the results suggest PVP can learn high quality agents with generalizability.

Table 2 presents the final performance in CARLA Town01. Compared to RL baselines, our method achieves a decent success rate and route completion rate even though we only utilize 18K environmental interactions. In Fig. 5, we compare the *intervention frequency* of PVP and HACO. It is the number of the human involvement segments in each episode divided by the episode length. Intervention frequency is not equivalent to intervention rate in that it measures how frequently the human subjects involve, reflecting the mental stress human subjects bear during shared control. Fig. 5 suggests HACO requires more human demonstration fragments than PVP. The drawback stems from that HACO utilizes SAC method with stochastic policy for better exploration. The actions produced by the novice policy varies drastically due to randomness, making human subjects stressful to actively intervene. A detailed visualization of the trajectories generated in human-robot shared control is available in Appendix F.2, clearly illustrating PVP method can generate smoother trajectories.

5.3 Comparicng PVP with HACO

Different Input Devices. Table 3 presents the experiment results with different input devices in MetaDrive benchmark. In both settings, the agents trained by PVP outperform those by HACO, showing the generalizability of PVP on different input devices.

Table 3: The impact of different human input devices in MetaDrive benchmark.

Input Device	Method	Training			Testing		
		Human Data Usage	Total Data Usage	Total Safety Cost	Episodic Return	Episodic Safety Cost	Success Rate
Wheel	HACO	21.2K (0.53)	40K	42	250.039	1.453	0.355
	PVP	10.3K (0.26)	40K	12	336.657	1.543	0.808
Gamepad	HACO	28.4K (0.71)	40K	55	71.37	1.97	0.0
	PVP	7.4K (0.19)	40K	21	356.99	1.31	0.920
Keyboard	HACO	19.2K (0.48)	40K	130	143.28	1.645	0.139
	PVP	14.6K (0.37)	40K	76	353.636	0.898	0.857

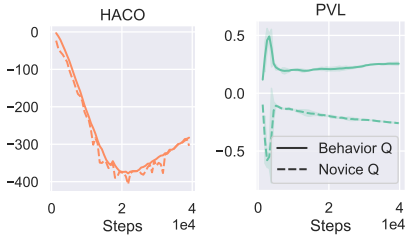


Figure 4: Evolution of values.

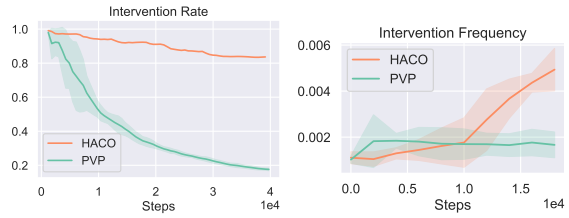


Figure 5: Intervention rate and intervention frequency in CARLA.

Comparing PVP with HACO. In Table 3, we observe that HACO (Li et al., 2022b) has performance discrepancy with different input devices. When using Gamepad, human subjects tend to push and pull the stick to the limits, producing extreme values. As we will discuss later, extreme actions are particularly harmful to HACO. When using keyboard, the human subjects press arrow keys to indicate increasing/decreasing current steering/acceleration values for an increment. Therefore there will be fewer extreme values happening than using a Gamepad, which explains why the baseline HACO performs better with the keyboard compared to Gamepad.

To explain why HACO is vulnerable to extreme action values, we compare the proxy Q values in our method and in HACO baseline in Fig. 4 and find that HACO has much larger magnitude in its proxy values compared to PVP. This is because HACO updates the state-action values based on CQL objective (Kumar et al., 2020): $\min Q(s, a_n) - Q(s, a_h)$. Since extreme values happen frequently when human intervenes through Gamepad, the proxy values in those human actions are reinforced without bound, making the novice policy rapidly learn those actions. In contrast, the proxy values learned via PVP are much more moderate. The novice Q has distinct negative value. In contrast, the values of behavior actions, the actions that satisfy human, have positive values. This result reveals the problem of unbounded values in the previous method. Our method instead resolves this issue and has bounded proxy values, leading to stable training.

5.4 Ablation Study

We conduct ablation studies to show the importance of TD learning and balanced buffers. As shown in Table 1, disabling TD learning via setting $J^{\text{TD}}(Q) = 0$ significantly damages the performance of PVP, suggesting that propagating information from human-involved states to other states is critical to the success of PVP. We find that reward has no significant impact on the learning performance. In Table 2, we find that disabling balanced buffers makes the training unstable and leads to poor performance. We also implement PVP based on Soft Actor-critic (Haarnoja et al., 2018) so that the novice policy is now a stochastic policy. The human subjects report that the novice agents in PVP w/ Stochastic Policy oscillate frequently, making them hard to respond when the agents suddenly drive toward the sideroad. We can find that PVP w/ Stochastic Policy performs worse compared to the deterministic version of PVP.

6 CONCLUSION

Learning through active human involvement is a promising approach to provide safe and efficient policy learning. In this work, we propose *Proxy Value Propagation (PVP)* that can effectively learn from active human involvement. PVP is applicable to existing value-based RL methods and induces reward-free policy learning. Experiments show the superior performance and low human cost of the proposed method in a wide range of tasks, action spaces, and human control devices.

Limitations. First, the Atari results show that our method has a defect in learning representations of RGB frames. Introducing self-supervised objective is a way to improve representation learning (Srinivas et al., 2020) especially when the data is insufficient as in our method. Second, we only apply our method to two value-based baselines. Many advanced techniques including exploration encouraging (Osband et al., 2016), prioritized replay buffer (Schaul et al., 2015) and double Q learning (Van Hasselt et al., 2016) can be added upon our method. Third, our method is not applicable to the tasks where humans can not provide demonstration, e.g. controlling a six-legged robot via torque at each joint.

REFERENCES

David Abel, John Salvatier, Andreas Stuhlmüller, and Owain Evans. Agent-agnostic human-in-the-loop reinforcement learning. *ArXiv preprint*, abs/1701.04079, 2017. URL https://arxiv.org/abs/1701.04079.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017. URL http://proceedings.mlr.press/v70/achiam17a.html.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4299–4307, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1582–1591. PMLR, 2018. URL http://proceedings.mlr.press/v80/fujimoto18a.html.

Lin Guan, Mudit Verma, Sihang Guo, Ruohan Zhang, and Subbarao Kambhampati. Widening the pipeline in human-guided reinforcement learning with explanation and context-aware data augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

Sehoon Ha, Peng Xu, Zhenyu Tan, Sergey Levine, and Jie Tan. Learning to walk in the real world with minimal human effort, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4565–4573, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/cc7e2b878868cbae992d1fb743995d8f-Abstract.html.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8022–8034, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/8cbe9ce23f42628c98f80fa0fac8b19a-Abstract.html.

Ananth Jonnavittula and Dylan P Losey. Learning to share autonomy across repeated interaction. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1851–1858. IEEE, 2021.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083. IEEE, 2019.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8248–8254. IEEE, 2019.

W Bradley Knox and Peter Stone. Reinforcement learning from human reward: Discounting in episodic tasks. In *2012 IEEE RO-MAN: The 21st IEEE international symposium on robot and human interactive communication*, pp. 878–885. IEEE, 2012.

Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. Specification gaming: the flip side of ai ingenuity. *DeepMind Blog*, 2020.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv preprint*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 2022a.

Quanyi Li, Zhenghao Peng, and Bolei Zhou. Efficient learning of safe driving policy via human-ai copilot optimization. In *International Conference on Learning Representations*, 2022b.

Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popovic. Where to add actions in human-in-the-loop reinforcement learning. In Satinder P. Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 2322–2328. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/15031.

Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-loop imitation learning using remote teleoperation. *ArXiv preprint*, abs/2012.06733, 2020. URL https://arxiv.org/abs/2012.06733.

Kunal Menda, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5041–5048. IEEE, 2019.

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):529–533, 2015.

Anis Najar, Olivier Sigaud, and Mohamed Chetouani. Interactively shaping robot behaviour with unlabeled human instructions. *Autonomous Agents and Multi-Agent Systems*, 34(2):1–35, 2020.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4026–4034, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8d8818c8e140c64c743113f563cf750f-Abstract.html.

Malayandi Palan, Gleb Shevchuk, Nicholas Charles Landolfi, and Dorsa Sadigh. Learning reward functions by integrating human demonstrations and preferences. In *Robotics: Science and Systems*, 2019.

Zhenghao Peng, Quanyi Li, Chunxiao Liu, and Bolei Zhou. Safe driving via expert guided policy optimization. In *5th Annual Conference on Robot Learning*, 2021.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Siddharth Reddy, Anca D Dragan, and Sergey Levine. Shared autonomy via deep reinforcement learning. *Robotics: Science and Systems*, 2018.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668. JMLR Workshop and Conference Proceedings, 2010.

Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. *UC Berkeley*, 2017.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *ArXiv preprint*, abs/1902.04043, 2019. URL https://arxiv.org/abs/1902.04043.

William Saunders, Girish Sastry, Andreas Stuhlmueller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv preprint*, abs/1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Jonathan Spencer, Sanjiban Choudhury, Matthew Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Siddhartha Srinivasa. Learning from interventions. In *Robotics: Science and Systems (RSS)*, 2020.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID lagrangian methods. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9133–9143. PMLR, 2020. URL http://proceedings.mlr.press/v119/stooke20a.html.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Fan Wang, Bo Zhou, Ke Chen, Tingxiang Fan, Xi Zhang, Jiangyong Li, Hao Tian, and Jia Pan. Intervention aided reinforcement learning for safe and practical policy optimization in navigation. In *Conference on Robot Learning*, pp. 410–421. PMLR, 2018.

Garrett Warnell, Nicholas R. Waytowich, Vernon Lawhern, and Peter Stone. Deep TAMER: interactive agent shaping in high-dimensional state spaces. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 1545–1554. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16200.

Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Yunkun Xu, Zhenyu Liu, Guifang Duan, Jiangcheng Zhu, Xiaolong Bai, and Jianrong Tan. Look before you leap: Safe model-based reinforcement learning with human intervention. In *Conference on Robot Learning*, pp. 332–341. PMLR, 2022.

Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end simulated driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

A ETHICS STATEMENT

Human subjects get paid to participate in the experiments. They can pause or stop the experiment if any discomfort happens. No human subjects are injured because all tasks we test are in virtual simulation. Each experiment will not last longer than one hour and subjects will rest at least three hours after one experiment. During training and data processing, no personal information is revealed in the collected dataset or the trained agents.

B DEMO VIDEO

Please find our demo video in the supplementary material. This video shows the footage of human experiments and the comparisons between agents learned by the baselines and the proposed method. The video contains three sections:

1) The first section shows how we learn driving policy in CARLA task within 20 minutes. We also compare the behavior of agents learned from PVP and TD3 baseline.
2) In the second section, we show the footage of MetaDrive human experiment where the human subject uses a gamepad as input device and demonstrate the behavior comparison between PVP and TD3 baseline.
3) In the third section, we show the applicability of our method in discrete control tasks. The behavior comparison between PVP and DQN baseline in MiniGrid Empty Room and Four Room and Atari Skiing tasks are provided.

C EXPECTED POLICY BEHAVIOR

In this section, we analyze the learning dynamics and provide insights on the learned policy. For simplicity, here we can define a proxy reward function:

$$R(s,a) = \begin{cases} +1 & \text{if } a \text{ is given by human during intervention} \\ -1 & \text{if } a \text{ is given by agent during intervention} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Assumption 1. *Assume the Q network can perfectly fit regression objective. Therefore it approximate both PVP and TD target equivalently. We can assume:*

$$Q(s,a) = \frac{R(s,a) + \gamma \max_{a'} Q(s',a')}{2}. \tag{9}$$

Finding 1 (Resembling Q Learning)**.** Combining the PV loss with TD loss is equivalent to the traditional Q learning objective with the reward function redirected to $R(s,a)/2$ and discount factor to $\gamma/2$.

The convergence of our method is promised by Q learning theory.

Denoting the last step is T and supposing action a_T leads to a terminal state, we have:

$$Q(s_T,a_T) = \begin{cases} +1 & \text{if } a_T = a_h \text{ and } I(s,a) = \text{True}, \\ -1 & \text{if } a_T = a_n \text{ and } I(s,a) = \text{True}, \\ 0 & \text{otherwise}. \end{cases} \tag{10}$$

Considering Assumption 1 and Eq. 10, for all steps $t < T$ that are prior to the step T, we can easily write the bound of Q values:

Theorem 2. *The proxy Q value is bounded:*

$$-1 \le Q(s_t,a_t) < \frac{1}{2-\gamma} \le 1. \tag{11}$$

Proof. It is easy to find $\min_a Q(s,a) = (-1-\gamma)/2 \geq -1$. For supremum, considering the contraction of Q:

$$\begin{aligned}\sup_a Q(s_t,a) &= \frac{1}{2} + \frac{\gamma}{2}\max_{a'} Q(s_{t+1},a') = \frac{1}{2} + \frac{\gamma}{2}(\frac{1}{2} + \frac{\gamma}{2}\max_{a'} Q(s_{t+2},a')) = ... \\ &= \frac{1}{2}(1 + \frac{\gamma}{2} + \frac{\gamma}{2}^2 + ...) = \frac{1}{2-\gamma} \leq 1.\end{aligned} \tag{12}$$

□

Finding 2 (Avoiding Value Explosion)**.** The Q value is bounded so that we can eliminate the value explosion issue. In contrast, previous work (Li et al., 2022b) uses unbounded value target so the Q value is vulnerable to overestimation and goes to infinity.

Now we discuss the behavior of the final proxy Q function. For arbitrary state action pair:

$$Q^*(s_t,a_t) = \begin{cases} \dfrac{1+\gamma \max_{a'} Q(s',a')}{2} \approx 1 & \text{if } a_t \text{ was previous human action,} \\ \dfrac{-1+\gamma \max_{a'} Q(s',a')}{2} \approx 0 & \text{if } a_t \text{ was previously overwritten by human,} \\ \dfrac{\gamma \max_{a'} Q(s',a')}{2} \approx \dfrac{1}{2} & \text{otherwise.} \end{cases} \tag{13}$$

Here we assume there is no attractor in state space. That is, there always exists a trajectory from current state to reach the state where human once intervened. This makes $\max_{a'} Q(s',a') \approx 1$ due to contraction. The optimal proxy value function sculptures such a Q value landscape that all previous human actions have highest $+1$ value. Those actions that lead to other states where human once involved also have higher values because the $+1$ value of the human-involved state will be propagated to those states.

Finding 3 (Implicit Intervention Minimization)**.** Since the agent learns policy that maximizes Q function, the proxy value propagation mechanism encourages agent to be human-imitating, that is to reproduce human actions or recover to the states where human once taught what to do. Our proxy Q value also penalizes actions that cause human intervention, which implicitly achieves intervention minimization.

D Human Intent Compliance

Here we provide a conceptual framework to describe the compliance of human intention. First, we introduce a ground-truth indicator $C : \mathcal{S} \times \mathcal{A} \to \{0,1\}$ of the intention violation, denoting whether the action is undesired. C is not revealed to learning algorithm.

$$C(s,a) = \begin{cases} 1, & \text{if } a \text{ violates human intention} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

We will derive the upper bound of the discounted occurrence of intent violation, a measure of training time human intent compliance:

$$S_{\pi_b} = S_{\pi_b}(s_0) = \mathop{\mathbb{E}}_{\tau \sim P_{\pi_b}} \sum_t \gamma^t C(s_t,a_t), \tag{15}$$

where P_{π_b} denotes the probability distribution of trajectories deduced by the behavior policy π_b.

During training, a human subject shares control with the learning agent. The agent's policy is a deterministic policy $\mu_n(s)$, the human's policy is a stochastic policy $\pi_h(a|s)$. The human subject intervenes $I(s,a) =$ True under certain state and agent's action a_n. The mixed behavior policy π_b that produces the real actions to environment is denoted as:

$$\pi_b(a|s) = (1 - I(s,\mu_n(s)))\delta(a - \mu_n(s)) + I(s,\mu_n(s))\pi_h(a|s), \tag{16}$$

where we use Dirac delta distribution to represent the deterministic novice policy.

Two important assumptions on the human subject are introduced:

Assumption 2 (Error rate of human policy)**.** *During human-AI shared control, the probability that the human subject produces an undesired action is bounded by a small value* $\epsilon < 1$*:*

$$\mathop{\mathbb{E}}_{s\sim P_{\pi_b}, a\sim\pi_h(\cdot|s)} C(s,a) \leq \epsilon. \tag{17}$$

Assumption 3 (Error rate of intervention policy)**.** *During human-AI shared control, the probability that the human subject does not intervene when the action is undesired is bounded by a small value* $\kappa < 1$*:*

$$\mathop{\mathbb{E}}_{s\sim P_{\pi_n}} (1 - I(s,\mu(s)))C(s,\mu(s)) \leq \kappa. \tag{18}$$

We introduce the following theorem and give the proof as follows.

Theorem 3 (Upper bound of intent violation)**.** *The discounted occurrence of intent violation* S_{π_b} *of the behavior policy* π_b *is bounded by the error rate of the human action* ϵ*, the error rate of the human intervention* κ *and the intervention rate* $\psi = \mathbb{E}_{s\sim P_{\pi_b}} I(s,a_n)$*:*

$$S_{\pi_b} \leq \frac{1}{1-\gamma}(\kappa + \epsilon\psi). \tag{19}$$

Proof. Consider Eq. 16, we have:

$$\begin{aligned}\mathop{\mathbb{E}}_{s\sim P_{\pi_b}, a\sim\pi_b(\cdot|s)} C(s,a) &= \mathop{\mathbb{E}}_{s\sim P_{\pi_b}} \{[1 - I(s,\mu_n(s))]C(s,\mu_n(s)) + I(s,\mu_n(s)) \mathop{\mathbb{E}}_{a\sim\pi_h(\cdot|s)} C(s,a)\} \\ &\leq \kappa + \epsilon \mathop{\mathbb{E}}_{s\sim P_{\pi_b}} I(s,\mu_n(s)) = \kappa + \epsilon\psi\end{aligned} \tag{20}$$

The upper bound of S_{π_b}:

$$S_{\pi_b} = \mathop{\mathbb{E}}_{\tau\sim P_{\pi_b}} \sum_{t=0} \gamma^t C(s_t,a_t) \leq \sum_{t=0} \gamma^t(\kappa + \epsilon\psi) = \frac{1}{1-\gamma}(\kappa + \epsilon\psi) \tag{21}$$

□

E Environment Details

Table 4: Summary of the environments we used for experiments.

Environment	Human Input Device	Observation Format	Action Space	Training & Test Set Split
MetaDrive	Gamepad, Keyboard, Wheel	State Vector	Continuous	Yes
CARLA	Wheel	Bird-eye View Image	Continuous	No
MiniGrid	Keyboard	Semantic Map	Discrete	No
Atari	Keyboard	RGB Image	Discrete	No

To avoid the potential risks of employing human subjects in physical experiments, we benchmark different approaches in four virtual simulated environments. We conduct experiments on various tasks with different observation and action spaces and human input devices. Table 4 summarizes the differences.

For continuous action space, we use two driving environments, MetaDrive Safety Benchmark (Li et al., 2022a) and CARLA Town01 environment (Dosovitskiy et al., 2017). In both tasks, the agent need to steer the target car with low-level acceleration, brake and steering and move toward destination.

MetaDrive Safety Benchmark preserves the capacity to evaluate the safety and generalizability in unseen environments, since it uses procedural generation to synthesize an unlimited number of driving maps for the split of training and test sets, which is useful to benchmark the generalization

capability of different approaches in the context of safe driving. We train agents in the training set, which contains 50 different scenes, and roll out the learning agents in the test set, which contains another 50 unique scenes. At each episode, the scene (road network) and the spawn location of traffic vehicles and ego vehicle are randomized. We use sensory state vector in MetaDrive as the observation for agents and thus apply MLP network architecture.

In CARLA, we train and test agents in the Town01 environment. There exist many predefined routes in the town with different spawn locations and destinations. The route is randomized for each episode. We use the bird-eye view image in CARLA as observation and thus CNN are used as feature extractors.

For discrete action space, we test on MiniGrid Two Room task (Chevalier-Boisvert et al., 2018) and Atari game Skiing task (Bellemare et al., 2013).

MiniGrid Two Room is a task requiring heavy exploration since the agent needs to move toward a door and open the door before reaching the destination. The spawn locations, the destinations, door locations and the geometry of each room are randomized. The observation of MiniGrid is the semantic map of agent's local neighborhood. MiniGrid tasks only support using keyboard as input device. Only in the MiniGrid task, we render the agent's action in the environment so that the human can decide whether to take over or return back based on both current state and agent's action. But this is not feasible in other tasks since other tasks require real-time response from humans and there is not enough time for humans to observe agent's actions even if we plot those actions in visualization interface.

The Atari game is difficult since the agent needs to learn to output meaningful action based on the RGB observation. We use the default setting provided by the Gym Atari environment. We experiment on Atari Skiing game. In preliminary experiments, we tried a few other Atari games. We find that human-in-the-loop method is not applicable to many games in Atari that have long horizons or are hard to play by human experts. For example, in the Breakout where the player must knock down as many bricks as possible by using the paddle below to ricochet the ball against the bricks and eliminate them, a full episode requires more than 120 seconds (can be much longer to reach a higher score) and the human expert has to attentively focus on playing the game. It is quite common for the human expert to miscalculate the trajectory and fail to rescue the ball from falling down. Another example is the Enduro, where the player controls a super fast racing car to run on an infinite long track. Experts cannot always give the optimal intervention in such a fast moving scene. In these environments, a pure RL agent will usually perform better even than the human experts. Besides as we discussed in the Limitation, representation learning is also a major challenge. It takes a long time for the convolutional neural network to learn good representations from the RGB frames. Therefore, the human-in-the-loop methods with active human involvement are not satisfactory in Atari games.

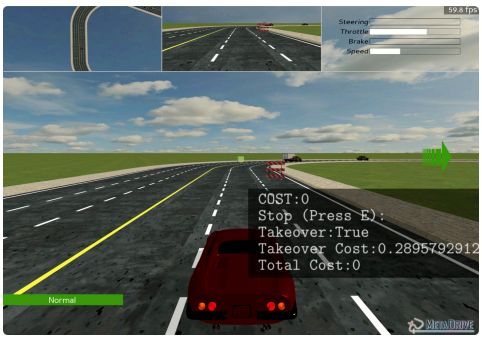


Figure 6: MetaDrive Safety benchmark.

Figure 7: CARLA Town01.

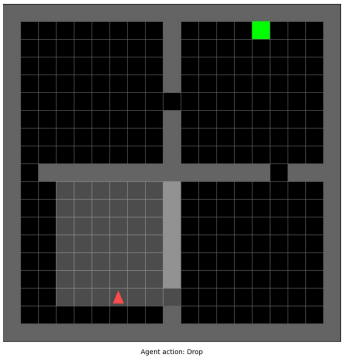


Figure 8: MiniGrid (Four Room).

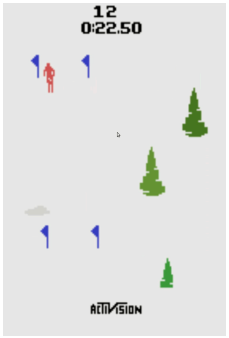


Figure 9: Atari (Skiing).

F Extra Experimental Results

F.1 Comparing PVP and HACO

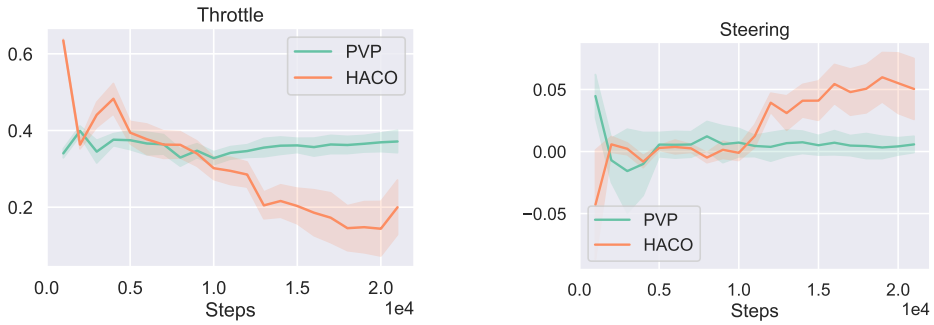


Figure 10: Control signals in a straight road in CARLA.

Randomness in HACO's policy is a reason leading to the suboptimal performance compared to PVP. We run the agents trained from HACO and PVP in CARLA task in a straight road in CARLA town and plot their actuating signals in Fig. 10. In this task, the steering should be always close to zero but it turn out that, as the training iterations increase, the HACO agents gradually demonstrate unstable steering. In human-AI shared control, such unstable behaviors force human subjects to involve frequently. In contrast, the PVP policies learn a much better solution in lane keeping. In Appendix F.2, we further present the visualization of the trajectories generated by human-robot shared control. The actions proposed by PVP policy are smoother than those by HACO. These results explain the performance of PVP in CARLA task and is aligned with the behavior shown in the supplementary video.

F.2 VISUALIZAATION OF HUMAN-ROBOT SHARED CONTROL

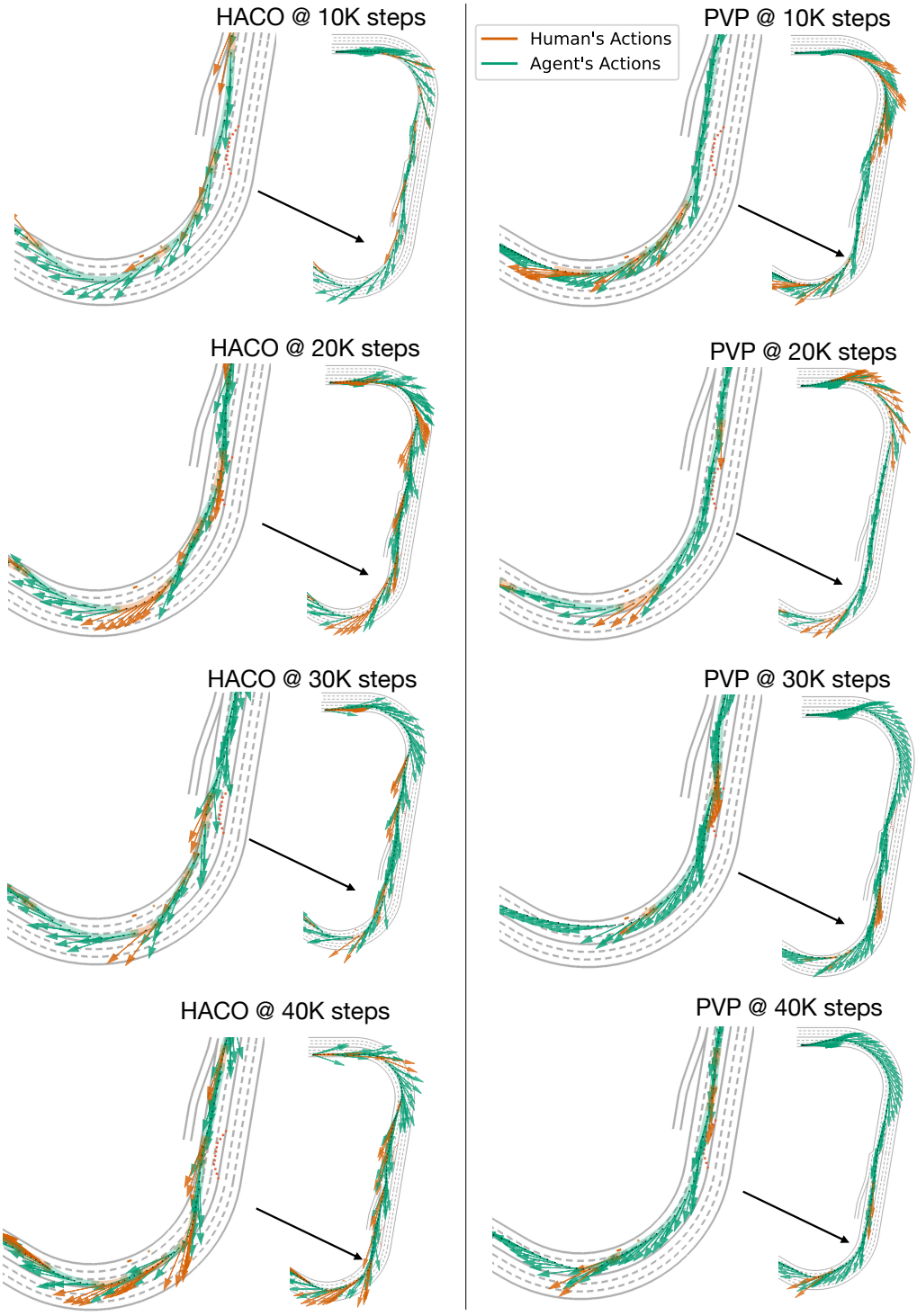
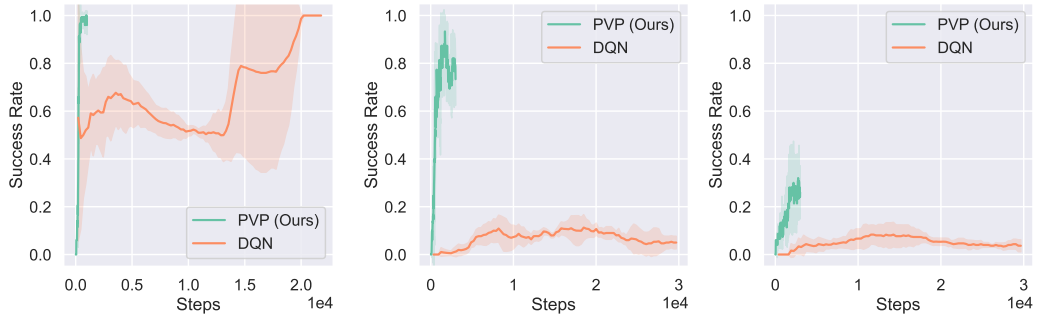


Figure 11: In MetaDrive task, we use the top-down view to plot the trajectories of human-robot shared control. We use dense arrows to represent the actions that are applied to the environments. The arrow starts at the position of the car at that time step and its direction is the steering angle, projected into ego car's local coordination. The length of arrow represents the acceleration. We use green and yellow arrows to denote agent's action and human's action, respectively.

In Fig. 11, we present the visualization of the trajectories during human-robot shared control. Comparing the visualization of HACO and PVP, we find that PVP generates smoother trajectories. The credit belongs to the deterministic policy we used. Stable and smooth agent actions greatly improve human subjects' experience and relieve their stress during human-robot shared control. We can also find that as the training goes, PVP requires less human involvement.

F.3 Extra Results in MiniGrid



(a) MiniGrid-Empty-Random-6x6-v0 (b) MiniGrid-TwoRooms-v0 (c) MiniGrid-FourRooms-v0

Figure 12: MiniGrid results.

In Fig. 12, we present the extra results in two additional MiniGrid environments. PVP achieves superior performance compared to RL baseline. Note that we use a CNN without recurrent module as the feature extractor. The performance of PVP can be further improved if we utilize the neural architecture with memory capacity.

G Hyper-parameters

In MetaDrive safety benchmark (Li et al., 2022a) task, the observation is a state vector. There exists a split of training and test environments in MetaDrive. We present the result of learned agent performing in test environment.

In CARLA (Dosovitskiy et al., 2017), the observation is the bird-eye view image in $[84, 84, 5]$ shape, where 5 is the number of semantic channels. We train and evaluate the agents in the same NoCrashTown01 environment.

In both driving tasks, the agent needs to steer the target car with low-level acceleration, brake and steering and move toward destination and thus the action space is a two dimensional space.

In MiniGrid tasks (Chevalier-Boisvert et al., 2018) MiniGrid-Empty-Random-6x6-v0 (Empty Room), MiniGrid-MultiRoom-N2-S4-v0 (Two Room) and MiniGrid-MultiRoom-N4-S5-v0 (Four Room), the observation is the top-down view semantic map in shape $[7, 7, 3]$.

In Atari game Skiing task (Bellemare et al., 2013), the observation is originally [210, 160] and we resize and preprocess the images following (Mnih et al., 2015). We also stack 4 consecutive frames so the input to neural network is in shape [84, 84, 4].

For CARLA and Atari tasks, since the input image has the same size of [84, 84] pixels, we use the same 5-layers CNN architecture with [16, 32, 64, 128, 256] filters in each layers. The corresponding kernel-size is [[4, 4], [3, 3], [3, 3], [3, 3], [4, 4]], and strides [3, 2, 2, 2, 4]. We use ReLU as activation functions between each layer.

For MiniGrid tasks, we use 3-layers CNN architecture with [16, 16, 32] filters in each layer. All three layers have kernel-size 2 and there is a max-pooling layer between the first two layers. We use ReLU as activation functions between each layer.

Table 5: PVP (MetaDrive)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate	0.0001
Steps before Learning Start	100
Steps per Iteration	1
Gradient Steps per Iteration	1
Train Batch Size	100
Q Value Bound	1

Table 6: PVP (CARLA)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate	0.0001
Steps before Learning Start	100
Steps per Iteration	1
Gradient Steps per Iteration	1
Train Batch Size	128
Q Value Bound	1

Table 7: PVP (MiniGrid)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate	0.0001
Steps before Learning Start	50
Steps per Iteration	1
Gradient Steps per Iteration	32
Target Network Update Interval	1
Train Batch Size	256
Q Value Bound	1
Exploration Reducing Fraction	0
Random Action Probability Initial Value	0
Random Action Probability Final Value	0

Table 8: PVP (Atari)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate	0.0001
Steps before Learning Start	100
Steps per Iteration	1
Gradient Steps per Iteration	8
Target Network Update Interval	1
Train Batch Size	256
Q Value Bound	1
Exploration Reducing Fraction	0
Random Action Probability Initial Value	0
Random Action Probability Final Value	0

Table 9: HACO (MetaDrive)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate Actor	0.0003
Learning Rate Critic	0.0003
Learning Rate Entropy	0.0003
Steps before Learning Start	100
Steps per Iteration	1
Gradient Steps per Iteration	1
Target Network Update Interval	1
Train Batch Size	128
CQL Loss Temperature	1.0

Table 10: HACO (Carla)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate Actor	0.0003
Learning Rate Critic	0.0003
Learning Rate Entropy	0.0003
Steps before Learning Start	100
Steps per Iteration	1
Gradient Steps per Iteration	1
Target Network Update Interval	1
Train Batch Size	128
CQL Loss Temperature	1.0

Table 11: TD3 (MetaDrive)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate	0.0001
Steps before Learning Start	10000
Steps per Iteration	1
Gradient Steps per Iteration	1
Train Batch Size	100

Table 12: TD3 (Carla)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate	0.0001
Steps before Learning Start	10000
Steps per Iteration	1
Gradient Steps per Iteration	1
Train Batch Size	100

Table 13: DQN (MiniGrid)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	0.005
Learning Rate	0.0001
Steps before Learning Start	50
Steps per Iteration	1
Gradient Steps per Iteration	32
Target Network Update Interval	1
Train Batch Size	256
Exploration Reducing Fraction	0.3
Random Action Probability Initial Value	0
Random Action Probability Final Value	0.05

Table 14: DQN (Atari)

Hyper-parameter	Value
Discounted Factor γ	0.99
τ for Target Network Update	1
Learning Rate	0.0001
Steps before Learning Start	100000
Steps per Iteration	4
Gradient Steps per Iteration	1
Target Network Update Interval	1000
Train Batch Size	32
Exploration Reducing Fraction	0.1
Random Action Probability Initial Value	0
Random Action Probability Final Value	0.01