Appendices

A ALGORITHM

Algorithm 1 Decoupled Policy Optimization

Input: State-only expert demonstration data $\mathcal{D} = \{(s_i)\}_{i=1}^{N}$, empty replay buffer $\mathcal{B}$, randomly initialized discriminator model D_ϕ, state transition predictor h_ψ and parameterized inverse dynamics model I_ϕ;
for $k = 0, 1, 2, \cdots$ **do** ▷ Pre-training stage
  Collect trajectories $\{(s, a, s', r, \text{done})\}$ using a random initialized policy $\pi = I_\phi(h_\psi)$ and store in $\mathcal{B}$
  Sample $(s, a, s') \sim \mathcal{B}$ and update ϕ by $\mathcal{L}_\phi(I)$
  Sample $(s, s') \sim \mathcal{D}$ and update ψ by $\mathcal{L}_\psi^h$
end for
for $k = 0, 1, 2, \cdots$ **do** ▷ Online training stage
  Collect trajectories $\{(s, a, s', r, \text{done})\}$ using current policy $\pi = I_\phi(h_\psi)$ and store in $\mathcal{B}$
  Sample $(s, a, s') \sim \mathcal{B}, (s, s') \sim \mathcal{D}$
  Update the discriminator D_ω with the loss:

$$\mathcal{L}_\omega^D = -\mathbb{E}_{(s,s')\sim\mathcal{B}}[\log D_\omega(s, s')] - \mathbb{E}_{(s,s')\sim\mathcal{D}}[\log(1 - D_\omega(s, s'))] , \tag{17}$$

  Update ϕ, ψ by $\mathcal{L}_{\phi,\psi}^{h,I}$
end for

B PROOFS

In our proofs we will work in finite state and action spaces $\mathcal{S}$ and $\mathcal{A}$ to avoid technical machinery out of the scope of this paper.

Proposition 1. *Suppose Π is the policy space and $\mathcal{P}$ is a valid set of state transition OMs such that $\mathcal{P} = \{\rho : \rho \geq 0$ and $\exists \pi \in \Pi$, s.t. $\rho(s, s') = \rho_0(s) \int_a \pi(a|s)\mathcal{T}(s'|s, a)\,\mathrm{d}a + \int_{s'',a} \pi(a|s)\mathcal{T}(s'|s, a)\rho(s'', s)\,\mathrm{d}s''\,\mathrm{d}a\}$, then a policy $\pi \in \Pi$ corresponds to one state transition OM $\rho_\pi \in \mathcal{P}$. However, under the action-redundant assumption about the dynamics $\mathcal{T}$, a state transition OM $\rho \in \mathcal{P}$ can correspond to more than one policy in Π.*

Proof. We first provide the proof for the one-to-one correspondence between marginal distribution $\sum_a \pi(a|s)\mathcal{T}(s'|s, a)$ and state transition OM $\rho(s, s') \in \mathcal{P}$.

For a given policy π, by definition of state transition OM, we have

$$\begin{aligned} \rho_\pi(s, s') &= \sum_a \mathcal{T}(s'|s, a)\rho_\pi(s, a) \\ &= \sum_a \pi(a|s)\mathcal{T}(s'|s, a) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi) . \end{aligned} \tag{18}$$

For all t greater than or equal to 1, we have

$$P(s_t = s|\pi) = \sum_{s''} P(s_{t-1} = s'', s_t = s|\pi) . \tag{19}$$

Take Eq. (19) into Eq. (18), we have

$$\begin{aligned}\rho_\pi(s,s') &= P(s_0=s,s_1=s') + \sum_a \pi(a|s)\mathcal{T}(s'|s,a)\sum_{t=1}^{\infty}\gamma^t \sum_{s''} P(s_{t-1}=s'', s_t=s|\pi) \\ &= P(s_0=s,s_1=s') + \gamma\sum_a \pi(a|s)\mathcal{T}(s'|s,a)\sum_{s''}\sum_{t=0}^{\infty}\gamma^t P(s_t=s'', s_{t+1}=s|\pi) \\ &= P(s_0=s,s_1=s') + \gamma\sum_a \pi(a|s)\mathcal{T}(s'|s,a)\sum_{s''}\rho_\pi(s'',s) \\ &= \rho_0(s)\sum_a \pi(a|s)\mathcal{T}(s'|s,a) + \gamma\sum_{s'',a}\pi(a|s)\mathcal{T}(s'|s,a)\rho_\pi(s'',s)\end{aligned} \tag{20}$$

Consider the following equation of variable ρ:

$$\rho(s,s') = \rho_0(s)\sum_a \pi(a|s)\mathcal{T}(s'|s,a) + \gamma\sum_{s'',a}\pi(a|s)\mathcal{T}(s'|s,a)\rho(s'',s)\ . \tag{21}$$

According to Eq. (20), ρ_π is a solution of Eq. (21). Now we proceed to prove ρ_π as the unique solution of Eq. (21).

Define the matrix

$$A_{(ss',s''s)} \triangleq \begin{cases} 1-\gamma\sum_a \pi(a|s)\mathcal{T}(s'|s,a) & \text{if } (s,s')=(s'',s) \\ -\gamma\sum_a \pi(a|s)\mathcal{T}(s'|s,a) & \text{otherwise .}\end{cases}$$

Note that A is a two-dimensional matrix indexed by state transition pairs. Also define the vector

$$b_{s,s'} \triangleq \rho_0(s)\sum_a \pi(a|s)\mathcal{T}(s'|s,a)\ .$$

We can rewrite Eq. (21) equivalently as

$$A\rho = b\ . \tag{22}$$

Since $\sum_{s',a}\pi(a|s)\mathcal{T}(s'|s,a)=1$ and $\gamma<1$, for all (s'',s), we have

$$\begin{aligned}&\sum_{s,s'}\gamma\sum_a \pi(a|s)\mathcal{T}(s'|s,a) = \gamma < 1 \\ \Rightarrow\ & 1-\gamma\sum_a \pi(a|s'')\mathcal{T}(s|s'',a) > \sum_{(s,s')\neq(s'',s)}\gamma\sum_a \pi(a|s)\mathcal{T}(s'|s,a) \\ \Rightarrow\ & \left|A_{(s''s,s''s)}\right| \geq \sum_{(s,s')\neq(s'',s)}\left|A_{(ss',s''s)}\right|\ .\end{aligned}$$

Therefore, we have proven A as column-wise strictly diagonally dominant, which implies that A is non-singular, so Eq. (21) has at most one solution. Since for all ρ in $\mathcal{P}$, it must satisfy the constraint Eq. (21), which means that for any marginal distribution $\sum_a \pi(a|s)\mathcal{T}(s'|s,a)$, there is only one corresponding ρ in $\mathcal{P}$.

Now, we proceed to prove that for every ρ in $\mathcal{P}$, there is only one corresponding marginal distribution $\sum_a \pi(a|s)\mathcal{T}(s'|s,a)$ such that $\rho_\pi=\rho$. By definition of $\mathcal{P}$, ρ is the solution of Eq. (21) for some policy π. By rewriting Eq. (21), the marginal distribution can be written in the form of a function expression of ρ as

$$\sum_a \pi(a|s)\mathcal{T}(s'|s,a) = \frac{\rho(s,s')}{\rho_0(s)+\gamma\sum_{s''}\rho(s'',s)}\ . \tag{23}$$

This means every $\rho \in \mathcal{P}$ only corresponds to one marginal distribution $\sum_a \pi(a|s)\mathcal{T}(s'|s,a)$. As we discussed before, ρ is the state transition OM of π, i.e., $\rho = \rho_\pi$.

By establishing the one-to-one correspondence between the marginal distribution $\sum_a \pi(a|s)\mathcal{T}(s'|s,a)$ and state transition OM $\rho \in \mathcal{P}$, we can alternatively study the correspondence between the marginal distribution and policy. Obviously, one policy can only corresponds to one marginal distribution. We now prove that if the dynamics $\mathcal{T}$ has redundant actions, one marginal distribution can correspond to more than one policy in π.

We prove the statement by counterexample construction. If the dynamics $\mathcal{T}$ has redundant actions, there exist $s_m \in \mathcal{S}$, $a_n \in \mathcal{A}$ and distribution p defined on $\mathcal{A} \setminus \{a_n\}$ such that $\sum_{a \in \mathcal{A}\setminus\{a_n\}} p(a)\mathcal{T}(s'|s_m,a) = \mathcal{T}(s'|s_m,a_n)$. Consider two policy π_0 and π_1 such that

$$\begin{cases} \pi_0(a|s) = \pi_1(a|s) & \text{if } s \neq s_m \\ \pi_0(a_n|s_m) = 1 & \\ \pi_0(a|s_m) = 0 & \text{if } a \neq a_n \\ \pi_1(a_n|s_m) = 0 & \\ \pi_1(a|s_m) = p(a) & \text{if } a \neq a_n \ . \end{cases} \tag{24}$$

From Eq. (24), we know that π_0 and π_1 are two different policies. However, they share the same marginal distribution $\sum_a \pi(a|s)\mathcal{T}(s'|s,a)$. To justify this, we first consider the case when s equals to s_m, where we have

$$\begin{aligned} \sum_a \pi_0(a|s_m)\mathcal{T}(s'|s_i,a) &= \pi_0(a_n|s_m)\mathcal{T}(s'|s_m,a_n) + \sum_{a\in\mathcal{A}\setminus\{a_n\}} \pi_0(a|s_m)\mathcal{T}(s'|s_m,a) \\ &= \mathcal{T}(s'|s_m,a_n) \\ &= \sum_{a\in\mathcal{A}\setminus\{a_n\}} \pi_1(a|s_m)\mathcal{T}(s'|s_m,a) \\ &= \sum_{a\in\mathcal{A}\setminus\{a_n\}} \pi_1(a|s_m)\mathcal{T}(s'|s_m,a) + \pi_1(a_n|s_m)\mathcal{T}(s'|s_m,a_n) \\ &= \sum_a \pi_1(a|s_m)\mathcal{T}(s'|s_m,a) \ . \end{aligned} \tag{25}$$

When s does not equal to s_m, the equality holds trivially, since the action selection probability of π_0 and π_1 defined on these states are exactly the same. Thus, one marginal distribution can correspond to more than one policy in π when there are redundant actions.

□

Proposition 2. *Suppose the state transition predictor h_Ω is defined as in Eq. (3) and $\Gamma = \{h_\Omega : \Omega \in \Lambda\}$ is a valid set of the state transition predictors, $\mathcal{P}$ is a valid set of the state-transition OMs defined as in Proposition 1, then a state transition predictor $h_\Omega \in \Gamma$ corresponds to one state transition OM $\rho_\Omega \in \mathcal{P}$; and a state transition OM $\rho \in \mathcal{P}$ only corresponds to one hyper-policy state transition predictor such that $h_\rho = \rho(s,s')/\int_{s'} \rho(s,s')\,\mathrm{d}s'$.*

Proof. During the proof of Proposition 1, we have an intermediate result that there is one-to-one correspondence between the marginal distribution $\sum_a \pi(a|s)\mathcal{T}(s'|s,a)$ and state transition OM $\rho \in \mathcal{P}$. Since the definition of state transition predictor is exactly $h_\Omega(s'|s) = \sum_a \pi(a|s)\mathcal{T}(s'|s,a)$ ($\forall \pi \in \Omega$), the one-to-one correspondence naturally holds between state transition predictor $h(s'|s)$ and state transition OM $\rho \in \mathcal{P}$.

□

Theorem 1 (Error Bound of DPO)**.** *Consider a deterministic environment whose transition function $\mathcal{T}(s,a)$ is deterministic and L-Lipschitz. Assume the ground-truth state transition $h_{\Omega_E}(s)$ is deterministic, and for each policy $\pi \in \Pi$, its inverse dynamics I_π is also deterministic and C-Lipschitz.*

Then for any state s, the distance between the desired state s'_E and reaching state s' sampled by the decoupled policy is bounded by:

$$\|s' - s'_E\| \leq LC\|h_{\Omega_E}(s) - h_\psi(s)\| + L\|I_{\tilde{\pi}}(s, \hat{s}') - I_\phi(s, \hat{s}')\| \,, \tag{26}$$

where $\tilde{\pi}$ is a sampling policy that covers the state transition support of the expert hyper-policy and $\hat{s}' = h_\psi(s)$ is the predicted consecutive state.

Proof. Given a state s, the expert takes a step in a deterministic environment and get s'. We assume that the expert Ω_E can use any feasible policy $\tilde{\pi}$ that covers the support of Ω_E to reach s:

$$s'_E = \mathcal{T}(s, I_{\tilde{\pi}}(s, h_\Omega(s))) \tag{27}$$

Similarly, using decoupled policy, the agent predict $\hat{s}' = h_\psi(s)$ and infer an executing action by an inverse dynamics model $a = I_\phi(s, s')$, which is learned from the sampling policy $\tilde{\pi}$. Denote the reaching state of the agent as s':

$$s' = \mathcal{T}(s, I_\phi(s, h_\psi(s))) \tag{28}$$

Therefore, the distance between s' and s'_E is:

$$\|s' - s'_E\| = \|\mathcal{T}(s, I_{\tilde{\pi}}(s, h_\Omega(s))) - \mathcal{T}(s, I_\phi(s, h_\psi(s)))\|$$

Lets consider the deterministic transition on s is a function of a such that $s' = \mathcal{T}^s(a)$, then we continue the deviation:

$$\begin{aligned}
\|s' - s'_E\| &\leq \|\mathcal{T}^s(I_{\tilde{\pi}}(s, h_\Omega(s))) - \mathcal{T}^s(I_\phi(s, h_\psi(s)))\| \\
&\leq L\|I_{\tilde{\pi}}(s, h_\Omega(s))) - I_\phi(s, h_\psi(s))\| \\
&\leq L\|I_{\tilde{\pi}}(s, h_\Omega(s))) - I_\phi(s, h_\psi(s))\| \\
&\leq L\|I_{\tilde{\pi}}(s, h_\Omega(s))) - I_{\tilde{\pi}}(s, h_\psi(s))) + I_{\tilde{\pi}}(s, h_\psi(s))) - I_\phi(s, h_\psi(s))\|
\end{aligned}$$

Similarly we also take the inverse transition on s is a function of s' such that $a = I^s(s')$, then we have that:

$$\begin{aligned}
\|s' - s'_E\| &\leq L\|I^s_{\tilde{\pi}}(h_\Omega(s))) - I^s_{\tilde{\pi}}(h_\psi(s))) \\
&\quad + I^s_{\tilde{\pi}}(h_\psi(s))) - I^s_\phi(h_\psi(s))\| \\
&\leq L\|I^s_{\tilde{\pi}}(h_\Omega(s))) - I^s_{\tilde{\pi}}(h_\psi(s)))\| + L\|I^s_{\tilde{\pi}}(h_\psi(s))) - I^s_\phi(h_\psi(s))\| \\
&\leq LC\|h_\Omega(s)) - h_\psi(s))\| + L\|I^s_{\tilde{\pi}}(\hat{s}') - I^s_\phi(\hat{s}')\| \,.
\end{aligned} \tag{29}$$

□

Theorem 2 (Error Bound of BCO)**.** *Consider a deterministic environment whose transition function $\mathcal{T}(s, a)$ is deterministic and L-Lipschitz, and a parameterized policy $\pi_\psi(a|s)$ that learns from the label provided by a parameterized inverse dynamics model I_ϕ. Then for any state s, the distance between the desired state s'_E and reaching state s' sampled by a state-to-action policy as BCO (Torabi et al., 2018) is bounded by:*

$$\begin{aligned}
\|s' - s'_E\| \leq L \left\| \pi_\psi(a|s) - \int_{s'^*} p_{\pi_E}(s'^*|s) I_\phi(a|s, s'^*) \, \mathrm{d}s'^* \right\| \\
+ L \left\| \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\tilde{\pi}}(a|s, s'^*)) - p_{\pi_E}(s'^*|s) I_\phi(a|s, s'^*) \, \mathrm{d}s'^* \right\| \,,
\end{aligned} \tag{30}$$

where $\tilde{\pi} \in \omega_E$ is a policy instance of the expert hyper-policy ω_E such that $\mathcal{T}(s, \tilde{\pi}(s)) = s'_E$.

Proof.

$$
\begin{aligned}
\|s' - s'_E\| &= \|\mathcal{T}(s, \pi_\psi(s)) - \mathcal{T}(s, \tilde{\pi}(s))\| \\
&= \|\mathcal{T}^s(\pi_\psi(s)) - \mathcal{T}^s(\tilde{\pi}(s))\| \\
&\leq L\|\tilde{\pi}(a|s) - \pi_\psi(a|s)\| \\
&= L \Bigg\| \pi_\psi(a|s) - \int_{s'^*} p_{\pi_E}(s'^*|s) I_\phi(a|s, s'^*)\, \mathrm{d}s'^* \\
&\quad + \int_{s'^*} p_{\pi_E}(s'^*|s) I_\phi(a|s, s'^*)\, \mathrm{d}s'^* - \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\tilde{\pi}}(a|s, s'^*))\, \mathrm{d}s'^* \Bigg\| \\
&\leq L \left\| \pi_\psi(a|s) - \int_{s'^*} p_{\pi_E}(s'^*|s) I_\phi(a|s, s'^*)\, \mathrm{d}s'^* \right\| \\
&\quad + L \left\| \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\tilde{\pi}}(a|s, s'^*)) - p_{\pi_E}(s'^*|s) I_\phi(a|s, s'^*)\, \mathrm{d}s'^* \right\|
\end{aligned} \tag{31}
$$

□

An intuitive explanation for the bound is that BCO (Torabi et al., 2018) first seeks to recover a policy that shares the same hyper-policy with π_E via learning an inverse dynamics model and then try to conduct behavior cloning. Therefore the errors comes from the reconstruction error of $\tilde{\pi}$ using I_ϕ (the second term) and the fitting error of behavior cloning (the first term).

By comparing Theorem 1 and Theorem 2, it is observed that for reaching each state, BCO requires a good inverse dynamics model over the state space to construct $\tilde{\pi}$ and then conduct imitation learning to $\tilde{\pi}$, while DPO only requires to learn a good inverse dynamics model on the predicted state and directly construct $\tilde{\pi}$ without the second behavior cloning step. This intuition meets our evaluation results in experiment Section 5.1.

C State Transition Occupancy Measure Matching

In the literature of inverse reinforcement learning (Syed et al., 2008; Abbeel & Ng, 2004; Finn et al., 2016), the ambiguity comes from the multiple answer for matching the feature of the expert demonstrations. A feasible solution to this problem is the maximum entropy principle that models the expert data with probability models. In a recent work (Liu et al., 2021), the authors show that state-action OM matching corresponds to maximum entropy reinforcement learning. Specifically, consider modeling the state-action OM with the Boltzmann distribution as $\rho_\pi(s, a) \propto \exp r(s, a)$, then we have that:

$$
\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(\rho_\pi(s,a) \| \rho_{\pi_E}(s,a)) &= \sum_{s,a} \rho_\pi(s,a) \log \frac{\rho_\pi(s,a)}{\rho_{\pi_E}(s,a)} \\
&= \sum_{s,a} \rho_\pi(s,a) \left(-r(s,a) + \log \rho_\pi(s,a)\right) + \text{const} \\
&= \mathbb{E}_\pi \left[-r(s,a)\right] + \sum_{s,a} \rho_\pi(s,a) \log \rho_\pi(s,a) + \text{const} \\
&= \mathbb{E}_\pi \left[-r(s,a)\right] + \sum_{s,a} \rho_\pi(s,a) \log \left(\rho_\pi(s)\pi(a|s)\right) + \text{const} \\
&= \mathbb{E}_\pi \left[-r(s,a)\right] - H(\pi(a|s)) - H(\rho_\pi(s)) + \text{const} \\
&\leq \mathbb{E}_\pi \left[-r(s,a)\right] - H(\pi(a|s)) + \text{const}\,,
\end{aligned} \tag{32}
$$

Therefore, maximizing the entropy of the state-action OM accounts for maximizing the entropy of the policy such that conducting maximum entropy reinforcement learning with a recovered reward corresponds to the upper bound of the state-action OM matching problem. Similarly, if we model the

state transition OM with the Boltzmann distribution as $\rho_\pi(s, s') \propto \exp r(s, s')$, then:

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(\rho_\pi(s, s') \| \rho_{\pi_E}(s, s')) &= \sum_{s,s'} \rho_\pi(s, s') \log \frac{\rho_\pi(s, s')}{\rho_{\pi_E}(s, s')} \\
&= \sum_{s,s'} \rho_\pi(s, s') \left(-r(s, s') + \log \rho_\pi(s, s')\right) + \text{const} \\
&= \mathbb{E}_\pi \left[-r(s, s')\right] + \sum_{s,s'} \rho_\pi(s, s') \log \rho_\pi(s, s') + \text{const} \\
&= \mathbb{E}_\pi \left[-r(s, s')\right] + \sum_{s,s'} \rho_\pi(s, s') \log \rho_\pi(s, s') + \text{const} \\
&= \mathbb{E}_\pi \left[-r(s, s')\right] - H(\rho_\pi(s, s')) + \text{const} \,.
\end{aligned} \tag{33}$$

However, maximum the entropy of the state-transition OM $\rho_\pi(s, s') = \int_a \pi(a|s)\rho_\pi(s)T(s'|s, a)\,\mathrm{d}a$ does not account for maximizing the entropy of the policy, and therefore can not alleviate the ambiguity.

D Experiments

D.1 Experiment Settings

D.1.1 Real-World Traffic Dataset

NGSIM I-80 dataset includes three videos with a total length of 45 minutes recorded in a fixed area, from which 5596 driving trajectories of different vehicles can be obtained. We choose 85% of these trajectories as the training set and the remaining 15% as the test set. In our experiment, the state space includes the position and velocity vectors of the ego vehicle and six neighbor vehicles and the actions are acceleration and the change in steering angle.

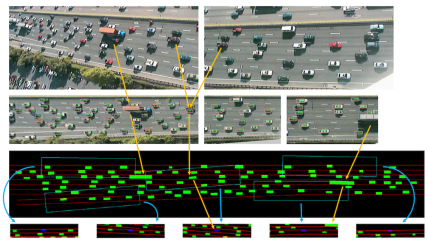

Figure 7: Visualization of NGSIM I-80 data set and its mapping on the simulator. This figure is borrowed from (Henaff et al., 2019).

D.1.2 Implementation Details

For all experiments, we implement the decoupled policy network, value network as two-layer MLPs with 256 hidden units and the discriminator as 128 hidden units. For fairness, we re-implement all the algorithms based on a Pytorch code framework[2] and adopt Soft Actor-Critic (SAC) (Haarnoja et al., 2018) as the RL learning algorithm for GAIfO and DPO.

For Mujoco benchmarks, we train an SAC agent to collect expert data, and take it for training the imitation learning agents without any normalization. At training time we remove the terminal state and episode will end until 1000 steps. At testing time the terminal state are set for fair comparison.

For NGSIM driving experiment, the original state contains the information of other cars, which is hard to predict. Therefore, we ignore it when predicting the state transition and the action of inverse dynamics. During training, we randomly pick one car to be controlled by the policy at the beginning of every episode, and we replay the other cars by data. The episode ends when cars

[2]https://github.com/KamyarGh/rl_swiss

collide or successfully get through the road. To reduce the sampling time in the driving simulator, we implemented parallel sampling using Python *multiprocessing* library. In practice, we ran 25 simulators to collect samples at the same time.

D.1.3 HYPERPARAMETERS

We list the key hyperparameters of the best performance of DPO on each task in Tab. 4. For each task, we first fine-tune GAIfO to find good hyperparameters for generative adversarial training, depending on which we further fine-tune state predictor coefficient λ_h and inverse dynamics coefficient λ_I from a initial hyperparameter $\lambda_h = 1.0$ and $\lambda_I = 0.5$. We find λ_h affects the performance most, along with the multi-step number k and the cycle loss. Note that DPO needs at least 1-step rollout for training the state transition predictor. In our experiment, we do not fine-tune the number of pre-training steps, and the final performances are almost the same with / without pre-training in most of environments. However, in few tasks, it can even deteriorate the training.

Table 4: Hyperparameters of DPO.

Environments	Invert.	InvDouble.	Hop.	Walk.	Half.	Ant.	NGSIM.
Trajectory maximum length	1000						1500
Optimizer	AdamOptimizer						
Discount factor γ	0.99						
Replay buffer size	2e5						2e6
Batch size	256						1024
State predictor coefficient λ_h	1.0				0.35	1.2	1.0
Tuning range of λ_h	[1.0]				[0.3,0.35,0.45,0.5,1.0]	[0.9,1.0,1.1,1.2,1.3]	[1.0]
Inverse dynamics coefficient λ_I	0.5				0.25	0.5	
Tuning range of λ_I	[0.5]				[0.25,0.5]	[0.5]	
Generative adversarial coefficient λ_G	1.0						
Generative adversarial reward form	$\log D$		$-\log(1-D)$	$\log D$			
Multi-step k	1		3	1	2		1
Cycle loss	✗			✓		✗	
Pre-train step	0		50000				0
Q learning rate	3e-4						
π learning rate	3e-4						
D learning rate	3e-4						
Gradient penalty weight	4.0				0.5		4.0
Reward scale	2.0						

D.2 QUALITATIVE ANALYSIS ON THE LEARNED POLICY

In this section, we provide qualitative experiments on investigating how the learned policy behaves. Based on the setting as in Section 5.1, we first analyse how the state transition predictor behaves. Specifically, we compare the learned prediction with the expert state transition, shown in Fig. 8, which indicates that the prediction by state transition predictor exactly match the state transition in the demonstration and achieve to the duty of 'plan the target'.

In addition, we further discuss the output action probability of the inverse dynamics, shown in Fig. 9. In the figure, we compare the action distribution among 20 possible actions at 3 different states with the expert ground truth. We conclude that the inverse dynamics mismatch is not the key for imitating expert the state sequence since the agent can select different actions from the expert as long as they lead to the same transition. Therefore, the inverse dynamics module play his role for 'learn the skill' of the agent's own.

D.3 ANALYSIS ON NO REDUNDANT ACTION

To better understand what the effect of the added action ambiguity achieves, we also include an experiment on the grid world environment in Section 5.1 when there are no redundant actions to show if empirically the baseline algorithms suffer from environments with action ambiguity. As shown in Fig. 10, BCO and DPO share similar asymptotic performance (KLD), but DPO has a significantly faster convergence rate. On the contrary, GAIfO also fails to find the second path.

D.4 DISTRIBUTIONAL EVALUATION METRIC

Apart from the accumulated reward reported in Tab. 2, the performance of imitation learning methods should also be evaluated by distributional similarities to expert data. For example, in SOIL tasks we try

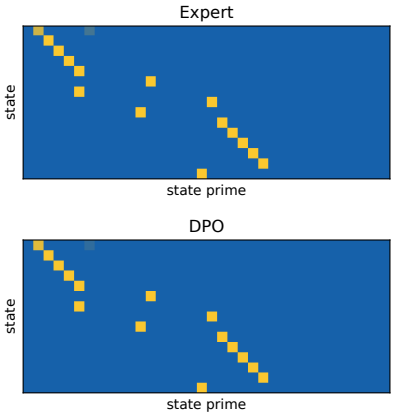

Figure 8: Each grid in the 'Expert' graph represents the transition probability from s (state) to s' (state prime) in demonstration data. Since some states do not appear in the demonstration, the transition probabilities from such states are undefined, and we exclude them from the graph. The 'DPO' graph shows the state prime output by state transition predictor from each state accordingly.

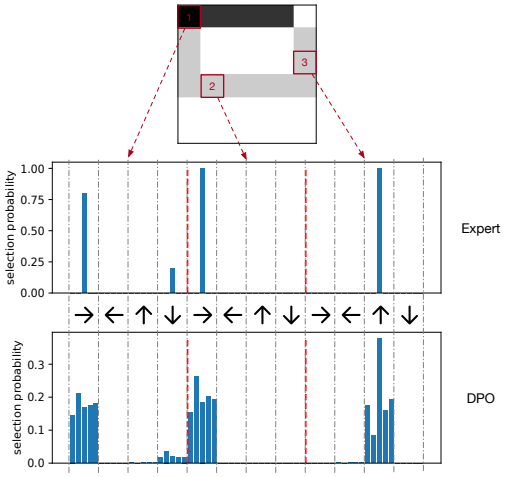

Figure 9: The learned policy (action selection probability) among 20 possible actions at 3 states (which is marked on the top sub-graph). The distributions in different states are split by red lines, and the resulting transition is labeled between two sub-graphs. This shows that the inverse dynamics mismatch is not the key for imitating expert the state sequence since the agent can select different actions from the expert as long as they lead to the same transition.

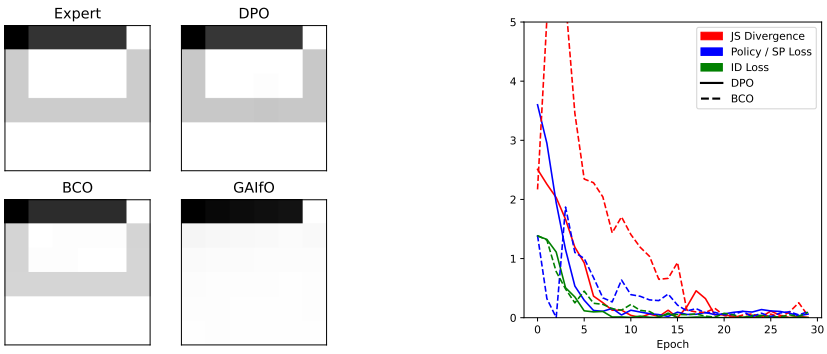

Figure 10: The rollout density and loss curves when $k = 1$. BCO and DPO have similar asymptotic performance (KLD), but DPO has a significantly faster convergence rate. On the contrary, GAIfO still fails to find the second path.

to evaluate the KL divergence between policy and expert state transitions $\mathrm{D}_{\mathrm{KL}}(\rho_{\pi_E}(s, s')\|\rho_{\pi}(s, s'))$ for different methods. Since it is hard to compute the distributional distance in high-dimensional continuous control environments, we reduce the dimension of the input data to 2 dimensions. Specifically, we adopt UMAP (McInnes & Healy, 2018), which maintains a mapping function that can be used for transforming new data collections. In our case, we first fit a UMAP model on the expert demonstration and then use it to transform (s, s') pairs collected by different algorithms. We first estimate the distribution via Kernel Density Estimation (KDE) (Rosenblatt, 1956) with Gaussian kernel to compute the Kullback-Leibler (KL) divergence, and show the qualitative results in Tab. 5. Furthermore, we visualize a 2 dimensional distributional density example of these trajectories on Halfcheetah in Fig. 11. Higher frequency positions in collected data are colored darker in the plane, and higher the value with respect to its marginal distributions. And it is noticeably that DPO does not reach a higher return but recover the better expert state transition occupancy measure.

Table 5: KL divergence between policy-sampled and the expert state transitions distribution.

	Hopper	Walker2d	HalfCheetah	Ant
BCO	**1.32 ± 0.04**	1.63 ± 0.26	5.76 ± 0.31	3.76 ± 0.42
GAIfO	1.77 ± 0.05	1.32 ± 0.21	2.47 ± 0.79	**0.40 ± 0.04**
DPO	1.76 ± 0.05	**1.13 ± 0.09**	**1.68 ± 0.16**	0.48 ± 0.06

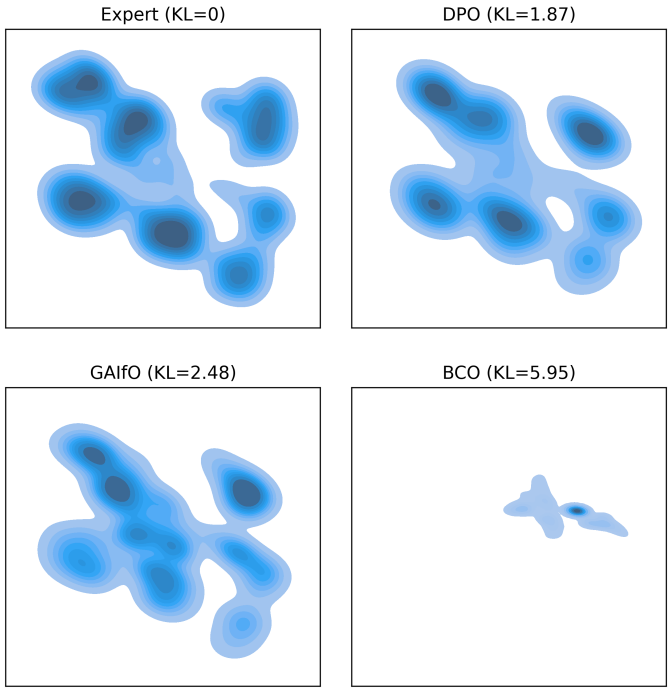

Figure 11: Visualization of sampled state transition distributions on HalfCheetah environment using UMAP reduction.

D.5 Ablation Study On Hyperparameters

In this section we investigate the effect on different values of hyperparameter λ_h. As illustrated in Fig. 12, the final performance is robust upon a range of λ_h. However, we find it affects the sample efficiency and the optimal hyperparameter among different tasks differs.

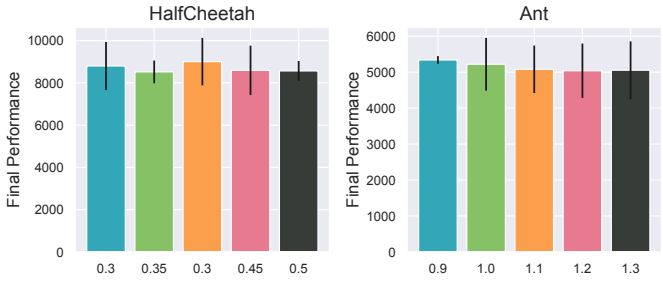

Figure 12: Hyperparameter study on λ_h.

D.6 Empirical Correlation between Compounding Error and Reward

The motivation of DPO indicates that if the agent can exactly predict where the expert will go and then learn a skill to reach that place, it can solve SOIL efficiently. In previous sections we propose to evaluate the distance of the reaching states and the predicted consecutive states to quantify the compounding error. Interestingly, in our experiments, we do find that the compounding error has a great impact on the efficacy of DPO. Therefore, we analyze the empirical correlation between the prediction-real distance and the reward. Specifically, we sample several epochs from experiments with different hyperparameters on each tasks and draw the connection of its prediction-real distance and its reward. As shown in Fig. 13, lower distance always achieves higher performance, indicating the rationality of the intuition and the key ingredient for utilizing DPO.

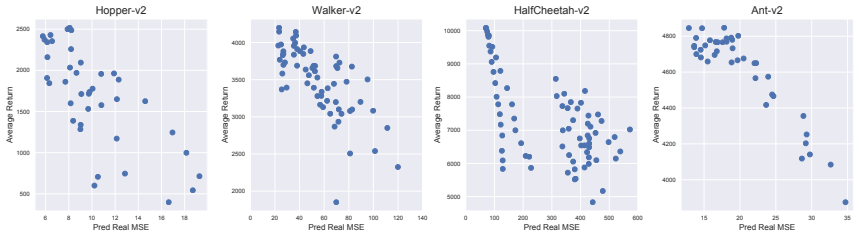

Figure 13: The empirical correlation between the prediction-real distance and the reward. Typically, less prediction-real distance achieves better performance

D.7 Complete Evaluation Results

In this section we show complete evaluation training curves of DPO with different regularization in Fig. 14. Typically, experiments with less prediction-real distance can achieve better performance. It is worth noting that, DPO can generally achieve better efficiency than the baselines in most of the environments. However, with fine-tuning the regularization, we are able to dig the potential of DPO.

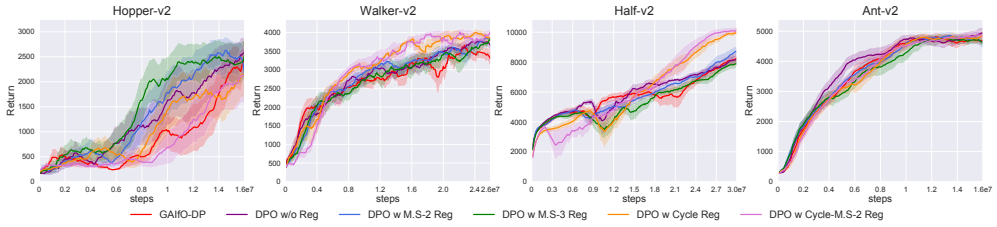

Figure 14: Complete learning curves of DPO with different regularization.

D.8 Additional Comparison and Ablation On Inverse Dynamics Regularization

In this section we emphasize on the regularization on the inverse dynamics. In this paper, specifically, we propose to utilize the policy gradient with generative adversarial methods to encourage the agent

Table 6: Additional ablation studies on the inverse dynamics regularization.

	InvertedPendulum	InvertedDoublePendulum	Hopper	Walker2d	HalfCheetah	Ant
BCO	**1000.0 ± 0.00**	416.92 ± 141.56	1516.91 ± 524.86	270.45 ± 33.22	6.56 ± 151.49	456.45 ± 179.76
GAIfO	**1000.0 ± 0.00**	**8589.46 ± 1391.82**	3068.10 ± 24.90	3864.03 ± 326.64	8918.66 ± 1031.41	4879.13 ± 897.46
DPO (w/o PG)	**1000.00 ± 0.00**	3933.33 ± 3414.51	713.17 ± 369.05	310.53 ± 68.27	-442.55 ± 120.23	-383.12 ± 198.18
DPO (w PG)	**1000.00 ± 0.00**	**8587.96 ± 1394.29**	**3163.74 ± 64.46**	**4395.21 ± 216.96**	**10522.08 ± 394.44**	5413.03 ± 161.97
DPO (w/o SP)	801.14 ± 444.66	5977.65 ± 4655.55	2587.88 ± 1237.44	3902.7 ± 171.12	8816.94 ± 998.3	5043.12 ± 552.29
DPO (w/o PG, w CL)	**1000.0 ± 0.0**	120.73 ± 23.17	1235.19 ± 938.26	20.63 ± 37.73	-585.45 ± 73.58	7.33 ± 22.56

to match the state transition of the expert. This is proved to improve the performance and the sample efficiency as shown in Section 5.2. However, in other point of view, our work can also be seen as adding additional supervision signals for a generative adversarial imitation learning method, due to the flexibility of the decoupled policy structure. Therefore, one can regard utilizing policy gradient (PG) in DPO as encouraging the imitator to follow the expert demonstrations better than a naive inverse model. Under this view, we aim to conduct ablation studies on the importance of the PG.

To this end, we compare PG with a related work (Pathak et al., 2018), which studies the problem of state-only imitation learning by matching the demonstrated image sequence. In the training stage, Pathak et al. (2018) allow the agent to learn a goal-conditioned skill (GSP) policy (i.e, an recurrent inverse dynamics model) to predict a plausible action; and in the evaluation stage, the agent is provided with demonstrated images, and it executes to achieve the every intermediate goal states one-by-one. Compared with DPO, Pathak et al. (2018) does not predict the target, instead the policy takes both the intermediate goal states s_g from the demonstration along with the action history a_h into planning the next action to reach the goal states. To achieve that, Pathak et al. (2018) also requires a binary classifier to identify whether the agent achieves the goals so that it can switch to the next goal state. Furthermore, Pathak et al. (2018) proposes a foward-consistency loss, which is used to regularize the inverse dynamics model to predict a plausible action; on the contrary, in our work, we want to regularize the state transition predictor to be consistent with the environmental forward dynamics, and therefore we do not apply the cycle loss to update the inverse dynamics model but utilize the PG regularization. Intuitively, in Pathak et al. (2018), the consistency is computed as:

$$s, s_g, a_h \xrightarrow{\text{GSP Policy}} a \xrightarrow{\text{Forward Model}} \underbrace{\tilde{s'} \quad s'}_{\text{Consistency Loss}} \xleftarrow{\text{execute (s,a)}} \text{Real Env}$$

However, our cycle consistency is more like the one in Edwards et al. (2020):

$$s_E \xrightarrow{\text{State Predictor}} \underbrace{\hat{s'} \xrightarrow{\text{Inverse Dynamics}} a \xrightarrow{\text{Forward Model}} \tilde{s'}}_{\text{Consistency Loss}}$$

Considering that using cycle consistency of Pathak et al. (2018) or discriminator rewards of ours in training of inverse dynamics are both encouraging state-matching, we include a comparison experiment for replace the policy gradients as the cycle consistency loss in Pathak et al. (2018), as denoted as DPO (w/o PG, w CL) shown in Tab. 6. From the results, we observe that the consistency regularization on inverse dynamics can decrease the compounding error in some environments (from the performance gain of DPO (w/o PG, w CL) over DPO (w/o PG)), but is far less effective than the PG regularization. This is rather obvious on the harder tasks with higher-dimensional state spaces.

To further illustrate if both supervision signals count, we also test the performance of DPO without the loss of expert state prediction, denoted as DPO (w/o SP) in Tab. 6. Obviously, without predicting the expert states, DPO (w/o SP) behaves even worse than GAIfO on some tasks since the inputs for the inverse dynamics no longer have semantic meanings and therefore the supervised loss for the inverse dynamics might hurt the performance. This ablation shows that explicitly predicting the expert's next states actually matters and is meaningful in achieving higher performance.