

485 Appendices

486 A Algorithm

Algorithm 1 Decoupled Policy Optimization

- 1: **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 ;
 - 2: **for** $k = 0, 1, 2, \dots$ **do** ▷ Pre-training stage
 - 3: Collect trajectories $\{(s, a, s', r, \text{done})\}$ using a random initialized policy $\pi = I_\phi(h_\psi)$ and store in \mathcal{B}
 - 4: Sample $(s, a, s') \sim \mathcal{B}$ and update ϕ by $\mathcal{L}_\phi(I)$
 - 5: Sample $(s, s') \sim \mathcal{D}$ and update ψ by \mathcal{L}_ψ^h
 - 6: **end for**
 - 7: **for** $k = 0, 1, 2, \dots$ **do** ▷ Online training stage
 - 8: Collect trajectories $\{(s, a, s', r, \text{done})\}$ using current policy $\pi = I_\phi(h_\psi)$ and store in \mathcal{B}
 - 9: Sample $(s, a, s') \sim \mathcal{B}, (s, s') \sim \mathcal{D}$
 - 10: Update the discriminator D_ω with the loss:

$$\mathcal{L}_\omega^D = -\mathbb{E}_{(s, s') \sim \mathcal{B}}[\log D_\omega(s, s')] - \mathbb{E}_{(s, s') \sim \mathcal{D}}[\log(1 - D_\omega(s, s'))], \quad (17)$$
 - 11: Update ϕ, ψ by $\mathcal{L}_{\phi, \psi}^{h, I}$
 - 12: **end for**
-

487 B Proofs

488 **Proposition 1.** *Suppose Π is the policy space and $\mathcal{P} = \{\rho : \rho \geq 0\}$ is a feasible set of OM, then a*
 489 *policy $\pi \in \Pi$ corresponds to one state transition OM $\rho_\pi \in \mathcal{P}$. However, a state transition OM $\rho \in \mathcal{P}$*
 490 *can correspond to more than one policy in Π .*

491 *Proof.* This is a trivial conclusion and we briefly explain a proof scratch. First, there is a one-to-one
 492 correspondence between the state-action OM and the policy since $\pi = \rho(s, a) / \int_{a^*} \rho(s, a^*) da^*$.
 493 Then it is easy to derive that there is a one-to-one correspondence between the joint OM and the
 494 policy since $\pi = \int_{s'^*} \rho(s, a, s'^*) ds'^* / \int_{a^*, s'^*} \rho(s, a^*, s'^*) da^* ds'^*$. The state transition OM can be
 495 obtained via marginalizing the policy $\rho(s, s') = \int_{a^*} \rho(s, a^*, s') da^* \int_{a^*, s'^*} \pi(a|s) \mathcal{T}(s'^*|s, a) da^* s'^*$.
 496 As a result, there is no one-to-one correspondence between $\rho(s, s')$ and $\pi(a|s)$. \square

497 **Theorem 1** (Error Bound of DPO). *Consider a deterministic environment whose transition function*
 498 *$\mathcal{T}(s, a)$ is deterministic and L -Lipschitz. Assume the ground-truth state transition $h_{\Omega_E}(s)$ is deter-*
 499 *ministic, and for each policy $\pi \in \Pi$, its inverse dynamics I_π is also deterministic and C -Lipschitz.*
 500 *Then for any state s , the distance between the desired state s'_E and reaching state s' sampled by the*
 501 *decoupled policy is bounded by:*

$$\|s' - s'_E\| \leq LC \|h_{\Omega_E}(s) - h_\psi(s)\| + L \|I_{\tilde{\pi}}(s, \hat{s}') - I_\phi(s, \hat{s}')\|, \quad (18)$$

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

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

$$s'_E = \mathcal{T}(s, I_{\tilde{\pi}}(s, h_\Omega(s))) \quad (19)$$

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

$$s' = \mathcal{T}(s, I_\phi(s, h_\psi(s))) \quad (20)$$

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

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

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

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

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

$$\begin{aligned} \|s' - s'_E\| &\leq L\|I_{\tilde{\pi}}^s(h_{\Omega}(s)) - I_{\tilde{\pi}}^s(h_{\psi}(s)) \\ &\quad + I_{\tilde{\pi}}^s(h_{\psi}(s)) - I_{\phi}^s(h_{\psi}(s))\| \\ &\leq L\|I_{\tilde{\pi}}^s(h_{\Omega}(s)) - I_{\tilde{\pi}}^s(h_{\psi}(s))\| + L\|I_{\tilde{\pi}}^s(h_{\psi}(s)) - I_{\phi}^s(h_{\psi}(s))\| \\ &\leq LC\|h_{\Omega}(s) - h_{\psi}(s)\| + L\|I_{\tilde{\pi}}^s(s') - I_{\phi}^s(s')\|. \end{aligned} \tag{21}$$

514 □

515 **Theorem 2** (Error Bound of BCO). *Consider a deterministic environment whose transition function*
516 *$\mathcal{T}(s, a)$ is deterministic and L -Lipschitz, and a parameterized policy $\pi_{\psi}(a|s)$ that learns from the*
517 *label provided by a parameterized inverse dynamics model I_{ϕ} . Then for any state s , the distance*
518 *between the desired state s'_E and reaching state s' sampled by a state-to-action policy as BCO [22]*
519 *is bounded by:*

$$\begin{aligned} \|s' - s'_E\| &\leq L \left\| \pi_{\psi}(a|s) - \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\phi}(a|s, s'^*) ds'^* \right\| \\ &\quad + L \left\| \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\tilde{\pi}}(a|s, s'^*) - p_{\pi_E}(s'^*|s) I_{\phi}(a|s, s'^*) ds'^* \right\|, \end{aligned} \tag{22}$$

520 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.

$$\begin{aligned} \|s' - s'_E\| &= \|\mathcal{T}(s, \pi_{\psi}(s)) - \mathcal{T}(s, \tilde{\pi}(s))\| \\ &= \|\mathcal{T}^s(\pi_{\psi}(s)) - \mathcal{T}^s(\tilde{\pi}(s))\| \\ &\leq L\|\tilde{\pi}(a|s) - \pi_{\psi}(a|s)\| \\ &= L \left\| \pi_{\psi}(a|s) - \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\phi}(a|s, s'^*) ds'^* \right. \\ &\quad \left. + \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\phi}(a|s, s'^*) ds'^* - \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\tilde{\pi}}(a|s, s'^*) ds'^* \right\| \\ &\leq L \left\| \pi_{\psi}(a|s) - \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\phi}(a|s, s'^*) ds'^* \right\| \\ &\quad + L \left\| \int_{s'^*} p_{\pi_E}(s'^*|s) I_{\tilde{\pi}}(a|s, s'^*) - p_{\pi_E}(s'^*|s) I_{\phi}(a|s, s'^*) ds'^* \right\| \end{aligned} \tag{23}$$

521 □

522 An intuitive explanation for the bound is that BCO [22] first seeks to recover a policy that shares the
523 same hyper-policy with π_E via learning an inverse dynamics model and then try to conduct behavior
524 cloning. Therefore the errors comes from the reconstruction error of $\tilde{\pi}$ using I_{ϕ} (the second term)
525 and the fitting error of behavior cloning (the first term).

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

531 **C State Transition Occupancy Measure Matching**

532 In the literature of inverse reinforcement learning [21, 1, 4], the ambiguity comes from the multiple
 533 answer for matching the feature of the expert demonstrations. A feasible solution to this problem is the
 534 maximum entropy principle that models the expert data with probability models. In a recent work [17],
 535 the authors show that state-action OM matching corresponds to maximum entropy reinforcement
 536 learning. Specifically, consider modeling the state-action OM with the Boltzmann distribution as
 537 $\rho_\pi(s, a) \propto \exp r(s, a)$, then we have that:

$$\begin{aligned}
 D_{\text{KL}}(\rho_\pi(s, a) \parallel \rho_{\pi_E}(s, a)) &= \sum_{s, a} \rho_\pi(s, a) \log \frac{\rho_\pi(s, a)}{\rho_{\pi_E}(s, a)} \\
 &= \sum_{s, a} \rho_\pi(s, a) (-r(s, a) + \log \rho_\pi(s, a)) + \text{const} \\
 &= \mathbb{E}_\pi [-r(s, a)] + \sum_{s, a} \rho_\pi(s, a) \log \rho_\pi(s, a) + \text{const} \quad (24) \\
 &= \mathbb{E}_\pi [-r(s, a)] + \sum_{s, a} \rho_\pi(s, a) \log (\rho_\pi(s) \pi(a|s)) + \text{const} \\
 &= \mathbb{E}_\pi [-r(s, a)] - H(\pi(a|s)) - H(\rho_\pi(s)) + \text{const} \\
 &\leq \mathbb{E}_\pi [-r(s, a)] - H(\pi(a|s)) + \text{const} ,
 \end{aligned}$$

538 Therefore, maximizing the entropy of the state-action OM accounts for maximizing the entropy of
 539 the policy such that conducting maximum entropy reinforcement learning with a recovered reward
 540 corresponds to the upper bound of the state-action OM matching problem. Similarly, if we model the
 541 state transition OM with the Boltzmann distribution as $\rho_\pi(s, s') \propto \exp r(s, s')$, then:

$$\begin{aligned}
 D_{\text{KL}}(\rho_\pi(s, s') \parallel \rho_{\pi_E}(s, s')) &= \sum_{s, s'} \rho_\pi(s, s') \log \frac{\rho_\pi(s, s')}{\rho_{\pi_E}(s, s')} \\
 &= \sum_{s, s'} \rho_\pi(s, s') (-r(s, s') + \log \rho_\pi(s, s')) + \text{const} \\
 &= \mathbb{E}_\pi [-r(s, s')] + \sum_{s, s'} \rho_\pi(s, s') \log \rho_\pi(s, s') + \text{const} \quad (25) \\
 &= \mathbb{E}_\pi [-r(s, s')] + \sum_{s, s'} \rho_\pi(s, s') \log \rho_\pi(s, s') + \text{const} \\
 &= \mathbb{E}_\pi [-r(s, s')] - H(\rho_\pi(s, s')) + \text{const} .
 \end{aligned}$$

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

545 **D Experiments**

546 **D.1 Experiment Settings**

547 **D.1.1 Real-World Traffic Dataset**

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

553 **D.2 Implementation Details**

554 For all experiments, we implement the decoupled policy network, value network as two-layer MLPs
 555 with 256 hidden units and the discriminator as 128 hidden units. For Mujoco benchmarks, we train

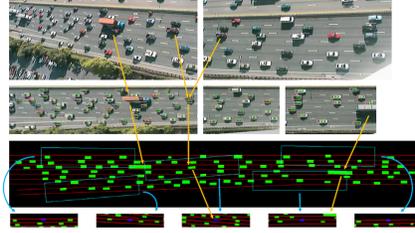


Figure 7: Visualization of NGSIM I-80 data set and its mapping on the simulator. This figure is borrowed from [9].

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

559 For NGSIM driving experiment, the original state contains the information of other cars, which
 560 is hard to predict. Therefore, we ignore it when predicting the state transition and the action of
 561 inverse dynamics. During training, we randomly pick one car to be controlled by the policy at the
 562 beginning of every episode, and we replay the other cars by data. The episode ends when cars
 563 collide or successfully get through the road. To reduce the sampling time in the driving simulator,
 564 we implemented parallel sampling using Python *multiprocessing* library. In practice, we ran 25
 565 simulators to collect samples at the same time.

566 D.3 Hyperparameters

567 We list the key hyperparameters of the best performance of DPO on each task in Tab. 4. For each task,
 568 we first fine-tune GAIfo to find good hyperparameters for generative adversarial training, depending
 569 on which we further fine-tune state predictor coefficient λ_h and inverse dynamics coefficient λ_I from
 570 a initial hyperparameter $\lambda_h = 1.0$ and $\lambda_I = 0.5$. We find λ_h affects the performance most, along
 571 with the multi-step number k and the cycle loss. We also find that pre-training does not help a lot for
 572 the final performance, sometimes it will even deteriorate the training. Note that DPO needs at least
 573 1-step rollout for training the state transition predictor.

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		\times			\checkmark		\times
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		

574 D.4 Distributional Evaluation Metric

575 Apart from the accumulated reward reported in Tab. 2, the performance of imitation learning methods
 576 should also be evaluated by distributional similarities to expert data. For example, in SOIL tasks we try
 577 to evaluate the KL divergence between policy and expert state transitions $D_{\text{KL}}(\rho_{\pi_E}(s, s') || \rho_{\pi}(s, s'))$
 578 for different methods. Since it is hard to compute the distributional distance in high-dimensional
 579 continuous control environments, we reduce the dimension of the input data to 2 dimensions. Specifi-

580 cally, we adopt UMAP [18], which maintains a mapping function that can be used for transforming
 581 new data collections. In our case, we first fit a UMAP model on the expert demonstration and then use
 582 it to transform (s, s') pairs collected by different algorithms. We first estimate the distribution via
 583 Kernel Density Estimation (KDE) [20] with Gaussian kernel to compute the Kullback-Leibler (KL)
 584 divergence, and show the qualitative results in Tab. 5. Furthermore, we visualize a 2 dimensional
 585 distributional density example of these trajectories on Halfcheetah in Fig. 8. Higher frequency
 586 positions in collected data are colored darker in the plane, and higher the value with respect to its
 587 marginal distributions. And it is noticeably that DPO does not reach a higher return but recover the
 588 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

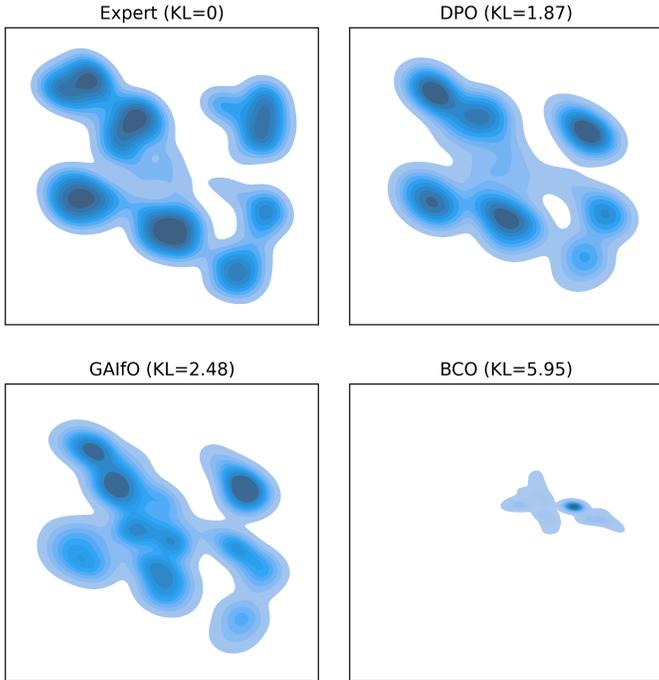


Figure 8: Visualization of sampled state transition distributions on HalfCheetah environment using UMAP reduction.

589 D.5 Ablation Study

590 In this section we investigate the effect on different values of hyperparameter λ_h . As illustrated in
 591 Fig. 9, the final performance is robust upon a range of λ_h . However, we find it affects the sample
 592 efficiency and the optimal hyperparameter among different tasks differs.

593 D.6 Empirical Correlation between Compounding Error and Reward

594 The motivation of DPO indicates that if the agent can exactly predict where the expert will go and
 595 then learn a skill to reach that place, it can solve SOIL efficiently. In previous sections we propose
 596 to evaluate the distance of the reaching states and the predicted consecutive states to quantify the
 597 compounding error. Interestingly, in our experiments, we do find that the compounding error has a

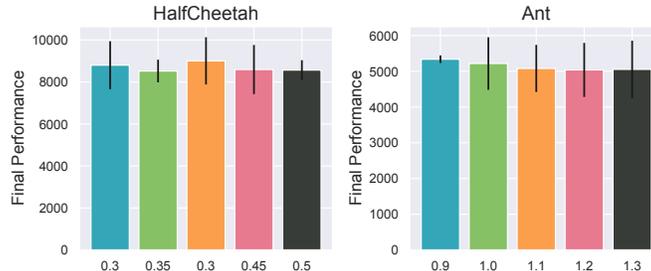


Figure 9: Hyperparameter study on λ_h .

598 great impact on the efficacy of DPO. Therefore, we analyze the empirical correlation between the
 599 prediction-real distance and the reward. Specifically, we sample several epochs from experiments
 600 with different hyperparameters on each tasks and draw the connection of its prediction-real distance
 601 and its reward. As shown in Fig. 10, lower distance always achieves higher performance, indicating
 the rationality of the intuition and the key ingredient for utilizing DPO.

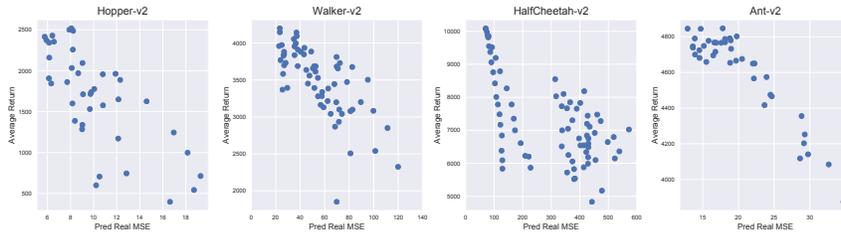


Figure 10: The empirical correlation between the prediction-real distance and the reward. Typically, less prediction-real distance achieves better performance

602

603 **D.7 Complete Evaluation Results**

604 In this section we show complete evaluation training curves of DPO with different regularization in
605 Fig. 11. Typically, experiments with less prediction-real distance can achieve better performance.

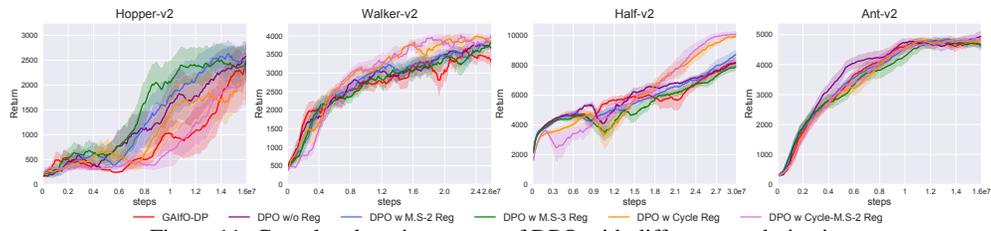


Figure 11: Complete learning curves of DPO with different regularization.