Appendix

A Derivation Details of DMIL and D2MIL

In this section, we provide the complete theoretical derivation of DMIL and D2MIL in Section A.1 and A.2. As D2MIL is a direct extension of DMIL with the addition of a second optimality discriminator, hence we will only discuss the detail model design philosophy of DMIL.

A.1 Derivation Details of DMIL

A Naïve Model-Based Offline IL Framework. We begin the derivation of DMIL by first inspecting the following naïve model-based offline IL framework, which simply incorporates a learned probabilistic dynamics model $f(s'|s,a)$ to generate rollout data $\mathcal{D}_r$ for policy learning:

$$\text{BC policy learning objective:} \quad \min_{\pi} \mathcal{L}_\pi := \mathbb{E}_{(s,a)\sim\mathcal{D}_e}[-\log \pi(a|s)] \tag{11}$$

$$\text{Dynamics model learning objective:} \quad \min_{f} \mathcal{L}_f := \mathbb{E}_{(s,a,s')\sim\mathcal{D}_e}[-\log f(s'|s,a)] \tag{12}$$

$$\text{Policy learning with } \mathcal{D}_e \text{ and } \mathcal{D}_r: \quad \min_{\pi} \mathcal{L}_\pi^{\text{fine-tune}} := \mathbb{E}_{(s,a)\sim\mathcal{D}_e\cup\mathcal{D}_r}[-\log \pi(a|s)] \tag{13}$$

Specifically, when we only use Eq.(12) and (13), it corresponds to the BC+d baseline in Section 3; if we first use Eq.(11) and (12) to pretrain the rollout policy and dynamics model to generate rollouts $\mathcal{D}_r$, then use Eq.(13) to fine-tune the policy, this corresponds to the 2-phase-BC+d baseline. Obviously, these two methods all bear some drawbacks. Both methods fully trust the model rollout data, which can be problematic when the dynamics model has high prediction errors or the policy is suboptimal. Although 2-phase-BC-d uses the higher quality pretrained dynamics model and policy to generate rollouts, it may still suffer from performance degeneration when the expert dataset is small.

A remedy for this is to selectively trust and train on good rollout data, but penalize the learning on problematic rollouts. A seemingly valid approach is to jointly learn a discriminator $d(s,a)$ together with policy π and dynamics model f to judge the dynamics correctness and optimality of rollouts in a GAN-like framework [30]. In this paradigm, π and f are jointly treated as the generator and optimized implicitly through solving a min-max optimization problem on the discriminator loss $\mathcal{L}_d$, which is the cross-entropy loss between $\mathcal{D}_e$ and $\mathcal{D}_r$. Although looks reasonable, this approach faces several technical problems. First, solving the GAN-style min-max optimization problem is costly and known to suffer from training instability and issues like mode collapse [48]. Second, as data in $\mathcal{D}_r$ are generated from a special multi-step rollout process using both π and f, rather than single-step outputs directly from a generator model in typical GAN framework, obtaining the correct gradients of π and f for back propagation through the discriminator loss $\mathcal{L}_d$ can be highly complex. Lastly, although we have explicit loss functions for policy π (Eq.(11) or (Eq.(13)) and f (Eq.(12)), they are not used to learn π and f in such a GAN-style framework. This could cause potential loss of information and performance degeneration when the expert data $\mathcal{D}_e$ contain noisy or suboptimal data. Since under the GAN framework, the only objectives of π and f are to fool the discriminator, rather than maximizing the likelihood on expert data.

Problem Reformulation Under the Cooperative-yet-Adversarial Learning Scheme. To address above issues, we introduce an adversarial-yet-cooperative learning scheme to jointly learn the policy π, dynamics model f and discriminator d. In particular, we first include the element-wise loss information from policy and dynamics model ($\log \pi$ and $\log f$) into the inputs of the discriminator d (i.e., $d(s,a,\log \pi(a|s), \log f(s'|s,a))$) to establish cooperative information sharing, and then use the following adversarial learning objective to learn the discriminator d:

$$\begin{aligned}\min_{d}\max_{\pi,f} \mathcal{L}_d := &\mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_e}[-\log d(s,a,\log \pi(a|s), \log f(s'|s,a))] + \\ &\mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_r}[-\log(1 - d(s,a,\log \pi(a|s), \log f(s'|s,a)))]\end{aligned} \tag{14}$$

Although this design looks not very intuitive, we can show that it offers a series of benefits. First, the information sharing couples the learning process of π, f and d, and also provides valuable information for d to make better judgment, as discussed in the main article in Section 2.2. Second, making π and f challenge the discriminator d by injecting adversarial information through $\log \pi(a|s)$ and $\log f(s'|s,a)$ will force the discriminator d to minimize the worst-case error of $\mathcal{L}_d$, which has

been shown in adversarial learning studies to greatly improve model robustness [31, 32]. Last and most importantly, we can show that this design enables reformulating the original complex coupled optimization problems (LHS of Eq.(15)) into three simple minimization problems as follows, which can be easily solved in a fully supervised learning manner to achieve high computation efficiency.

$$\begin{cases} \min_\pi \ \mathcal{L}_\pi \\ \min_f \ \mathcal{L}_f \\ \min\limits_d \max\limits_{\pi,f} \mathcal{L}_d \end{cases} \Rightarrow \quad \begin{cases} \min_\pi \ \mathcal{L}_\pi^{\text{DMIL}} := \alpha_\pi \cdot \mathcal{L}_\pi + \mathcal{L}_\pi^{corr} \\ \min_f \ \mathcal{L}_f^{\text{DMIL}} := \alpha_f \cdot \mathcal{L}_f + \mathcal{L}_f^{corr} \\ \min_d \ \mathcal{L}_d \end{cases} \tag{15}$$

where $\mathcal{L}_\pi$ and $\mathcal{L}_f$ are defined on $\mathcal{D}_e$ as shown in Eq.(11) and (12); $\mathcal{L}_\pi^{corr}$ and $\mathcal{L}_f^{corr}$ are corrective loss terms capturing the adversarial behavior of π and f on d, which are computed based on output values of the discriminator d on samples from both $\mathcal{D}_e$ and $\mathcal{D}_r$; $\alpha_\pi, \alpha_f \geq 1$ are weight factors of π and f to balance their original learning objectives and the additional adversarial behavior.

The corrective loss terms $\mathcal{L}_\pi^{corr}$ and $\mathcal{L}_f^{corr}$ are derived by finding equivalent relaxed conditions of the inner maximization problem for π and f in $\min\limits_d \max\limits_{\pi,f} \mathcal{L}_d$. This avoids solving the original complex functional min-max problem for the discriminator, and also enables learning π and f on both expert data $\mathcal{D}_e$ and model rollouts $\mathcal{D}_r$. Utilizing calculus of variation [33] and the analysis method introduced in Xu et al. [20], we provide the detailed derivation of the exact forms of $\mathcal{L}_\pi^{corr}$ and $\mathcal{L}_f^{corr}$ as follows.

Derivation of the Corrective Loss Terms. Under the proposed cooperative-yet-adversarial learning scheme, both the discriminator d and its loss $\mathcal{L}_d$ become functionals of π and f (i.e., function of a function), which can be expressed as $d(s, a, \log \pi(a|s), \log f(s'|s, a))$ and $\mathcal{L}_d(d, \log \pi, \log f)$. Denote $x = (s, a, s')$. Note that $\mathcal{L}_d(d, \log \pi, \log f)$ can be rewritten as following integral form of a new functional $F(x, d, \log \pi, \log f)$:

$$\begin{aligned} \mathcal{L}_d(d, \log \pi, \log f) =& \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_e} [-\log d(s, a, \log \pi(a|s), \log f(s'|s, a))] \\ &+ \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_r} [-\log(1 - d(s, a, \log \pi(a|s), \log f(s'|s, a)))] \\ =& \int_{\Omega_{sas'}} \big[P_{D_e}(x) \cdot [-\log d(s, a, \log \pi(a|s), \log f(s'|s, a))] \\ &+ P_{D_r}(x) \cdot [-(1 - \log d(s, a, \log \pi(a|s), \log f(s'|s, a)))]\big] \mathrm{d}x \\ \triangleq& \int_{\Omega_{sas'}} F(x, d, \log \pi, \log f) \mathrm{d}x \end{aligned} \tag{16}$$

where P_{D_e} and P_{D_r} are probability distributions of x in $\mathcal{D}_e$ and $\mathcal{D}_r$; and $\Omega_{sas'}$ is the domain of x under $\mathcal{D}_e \cup \mathcal{D}_r$.

To avoid solving the complex functional min-max problem $\min\limits_d \max\limits_{\pi,f} \mathcal{L}_d(d, \log \pi, \log f)$, we will focus on its inner maximization problem, which essentially requires to find the maxima of functional $\mathcal{L}_d(d, \log \pi, \log f)$ with respect to π and f, given an unknown functional d decided by the outer minimization problem. From functional analysis and calculus of variation[33], the extrema (maxima or minima) of $\mathcal{L}_d$ can be obtained by solving the following associate Euler-Lagrangian equations:

$$\begin{cases} F_\pi - \frac{\partial}{\partial x} F_{\frac{\partial \pi}{\partial x}} = F_\pi = 0 \\ F_f - \frac{\partial}{\partial x} F_{\frac{\partial f}{\partial x}} = F_f = 0 \end{cases} \tag{17}$$

where F_y stands for $\frac{\partial F}{\partial y}$. As $\frac{\partial \pi}{\partial x}$, and $\frac{\partial f}{\partial x}$ do not appear in the our form of $F(x, d, \log \pi, \log f)$, hence $F_{\frac{\partial \pi}{\partial x}} = F_{\frac{\partial f}{\partial x}} = 0$. Let θ_π and θ_f denote model parameters of π and f, above equations also indicate:

$$\begin{cases} F_\pi \cdot \frac{\partial \pi}{\partial \theta_\pi} = \frac{\partial F}{\partial d} \cdot \frac{\partial d}{\partial \log \pi} \cdot \frac{\partial \log \pi}{\pi} \cdot \frac{\partial \pi}{\partial \theta_\pi} = \frac{\partial F}{\partial d} \cdot \frac{\partial d}{\partial \log \pi} \cdot \nabla_{\theta_\pi} \log \pi = 0 \\ F_f \cdot \frac{\partial f}{\partial \theta_f} = \frac{\partial F}{\partial d} \cdot \frac{\partial d}{\partial \log f} \cdot \frac{\partial \log f}{f} \cdot \frac{\partial f}{\partial \theta_f} = \frac{\partial F}{\partial d} \cdot \frac{\partial d}{\partial \log f} \cdot \nabla_{\theta_f} \log f = 0 \end{cases} \tag{18}$$

In our problem, d, F, π and f are real-value functions, hence the same with the derivatives $\frac{\partial F}{\partial d}$, $\frac{\partial d}{\partial \log \pi}$ and $\frac{\partial d}{\partial \log f}$. If the continuity of previous functions and derivatives are satisfied, then according

to Hewitt [49], the set of real-valued continuous functions is a commutative ring, we can safely swap the order of $\frac{\partial F}{\partial d}$ and $\frac{\partial d}{\partial \log \pi}$, as well as $\frac{\partial F}{\partial d}$ and $\frac{\partial d}{\partial \log f}$ in above equations.

As d is determined by the outer minimization problem of Eq.(14), thus the exact forms of $\frac{\partial d}{\partial \log \pi}$ and $\frac{\partial d}{\partial \log f}$ are not obtainable by only inspecting the inner maximization problem. We can instead consider a alternative solution by making $\frac{\partial F}{\partial d} \cdot \nabla_{\theta_\pi} \log \pi = 0$ and $\frac{\partial F}{\partial d} \cdot \nabla_{\theta_f} \log f = 0$ for state-action pairs in $\Omega_s \times \Omega_a$. For practical IL tasks, $\mathcal{D}_e$ and $\mathcal{D}_r$ are finite, and the domains Ω_s and Ω_a are closed and bounded, hence the integration on $\frac{\partial F}{\partial d} \cdot \nabla_{\theta_\pi} \log \pi$ and $\frac{\partial F}{\partial d} \cdot \nabla_{\theta_f} \log f$ will still be zero. Interestingly, although it is intractable to directly solve $\frac{\partial F}{\partial d} \cdot \nabla_{\theta_\pi} \log \pi = 0$ and $\frac{\partial F}{\partial d} \cdot \nabla_{\theta_f} \log f = 0$, the integration on these equations leads to two new relaxed and tractable necessary conditions for $\mathcal{L}_d$ to reach its extrema. Using the condition on π as an example, we have:

$$
\begin{aligned}
0 &= \int_{\Omega_{sas'}} \frac{\partial F(x, d, \pi(a|s), f(s'|s,a))}{\partial d(s, a, \pi(a|s), f(s'|s,a))} \cdot \nabla_{\theta_\pi} \log \pi(a|s) \mathrm{d}x \\
&= \int_{\Omega_{sas'}} \Bigg[- P_{\mathcal{D}_e}(x) \cdot \frac{1}{d(s, a, \log \pi(a|s), \log f(s'|s,a))} \\
&\quad + P_{\mathcal{D}_o}(x) \cdot \frac{1}{1 - d(s, a, \log \pi(a|s), \log f(s'|s,a))} \Bigg] \cdot \nabla_{\theta_\pi} \log \pi(a|s) \mathrm{d}x \\
&= \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_e} \left[-\frac{1}{d} \cdot \nabla_{\theta_\pi} \log \pi \right] - \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_r} \left[-\frac{1}{1-d} \cdot \nabla_{\theta_\pi} \log \pi \right]
\end{aligned} \tag{19}
$$

where in the last equation, we slightly abuse the notations and write the output value of $d(s, a, \log \pi(a|s), \log f(s'|s,a))$ as d. Note that the above condition can be equivalently perceived as the first-order optimality condition of minimizing a new loss term $\mathcal{L}_\pi^{corr}$ with respect to π, i.e., derivative equal to zero, given as

$$
\mathcal{L}_\pi^{corr} = - \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_e} \left[-\frac{1}{d} \cdot \log \pi(a|s) \right] + \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_r} \left[-\frac{1}{1-d} \cdot \log \pi(a|s) \right] \tag{20}
$$

where we introduce a negative sign on the last equation in Eq.(19) to ensure minimizing $\mathcal{L}_\pi^{corr}$ leads to update π in the gradient ascent direction of $\mathcal{L}_d$, so as to find the maxima of $\mathcal{L}_d$ rather than minima.

Similarly to the derivation of $\mathcal{L}_\pi^{corr}$, we can get the corrective loss for the dynamics model $\mathcal{L}_f^{corr}$ as:

$$
\mathcal{L}_f^{corr} = - \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_e} \left[-\frac{1}{d} \cdot \log f(s'|s,a) \right] + \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_r} \left[-\frac{1}{1-d} \cdot \log f(s'|s,a) \right] \tag{21}
$$

Add these corrective loss terms to their original losses according to Eq.(15), we can get the final objectives for π and f in DMIL:

$$
\begin{aligned}
\mathcal{L}_\pi^{\text{DMIL}} &= \alpha_\pi \mathop{\mathbb{E}}_{(s,a) \sim \mathcal{D}_e} [-\log \pi(a|s)] - \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_e} \left[-\frac{1}{d} \cdot \log \pi(a|s) \right] + \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_r} \left[-\frac{1}{1-d} \cdot \log \pi(a|s) \right] \\
&= \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_e} \left[- \left(\alpha_\pi - \frac{1}{d} \right) \cdot \log \pi(a|s) \right] + \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_r} \left[-\frac{1}{1-d} \cdot \log \pi(a|s) \right]
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
\mathcal{L}_f^{\text{DMIL}} &= \alpha_f \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_e} [-\log f(s'|s,a)] - \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_e} \left[-\frac{1}{d} \cdot \log f(s'|s,a) \right] + \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_r} \left[-\frac{1}{1-d} \cdot \log f(s'|s,a) \right] \\
&= \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_e} \left[- \left(\alpha_f - \frac{1}{d} \right) \cdot \log f(s'|s,a) \right] + \mathop{\mathbb{E}}_{(s,a,s') \sim \mathcal{D}_r} \left[-\frac{1}{1-d} \cdot \log f(s'|s,a) \right]
\end{aligned} \tag{23}
$$

Note that we use $d(s, a, \log \pi(a|s), \log f(s'|s,a))$ as values in $\mathcal{L}_\pi^{corr}$ and $\mathcal{L}_f^{corr}$, thus there is no gradient passing from the discriminator d to π and f when minimizing $\mathcal{L}_\pi^{\text{DMIL}}$ and $\mathcal{L}_f^{\text{DMIL}}$. This greatly simplifies the learning processes of π, f and d, as all of them can be trained in a decoupled manner with their own optimization objectives (Eq.(15)), while also enabling capturing the coupled relationship with d using $\mathcal{L}_\pi^{corr}$ and $\mathcal{L}_f^{corr}$.

Interpretations of DMIL. The final learning objectives of π and f in Eq.(22) and (23) are actually intuitively reasonable. It can be perceived as assigning credibility weights on different samples

based on the judgment of the discriminator d, with weight $\alpha_\pi - 1/d$ and $\alpha_f - 1/d$ assigned to expert demonstrations and $1/(1-d)$ assigned to model rollout data. Suppose the discriminator is well-learned, then it will output small values for problematic model rollouts, resulting in lower weights ($1/(1-d) \to 1$) on these samples; whereas for credible rollout samples ($d \to 1$), the weights will be boosted and encourage the policy π to learn more on these samples. Moreover, the learned discriminator can also serve as a denoiser to alleviate noisy or suboptimal data in the expert dataset $\mathcal{D}_e$. For such samples, the output values of d will be small, and the weights $\alpha_\pi - 1/d$ and $\alpha_f - 1/d$ will be reduced for policy π and dynamics model f.

It should be noted that during our derivation, the continuity assumption of $\frac{\partial F}{\partial d}$ needs to be satisfied. We thus clip the output range of d to $[0.1, 0.9]$ to avoid $1/d$ and $1/(1-d)$ taking infinite values. We further set $\alpha_\pi = \alpha_f = 10$ in our implementation to ensure expert demonstrations in $\mathcal{D}_e$ always get positive weights.

A.2 Derivation Details of D2MIL

Problem Formulation of D2MIL. As for offline IL scenarios with a small expert dataset $\mathcal{D}_e$ and a large unknown, potentially suboptimal dataset $\mathcal{D}_o$, we can extend the proposed DMIL framework by adding a second optimality discriminator $d_o(s, a, \log \pi)$ to distinguish expert and non-expert samples, following a similar treatment as in DWBC [20]. Moreover, we also introduce a second pair of adversarial relationship between the policy π and d_o to carry over the similar reformulation design as in DMIL. For clarity, we will refer the original rollout discriminator in DMIL as d_r in the following discussion. Under this scenario, the set of problems we need to jointly solve are:

$$\begin{cases} \min_\pi \ \mathcal{L}_\pi := \mathbb{E}_{(s,a)\sim\mathcal{D}_e}[-\log \pi(a|s)] \\ \min_f \ \mathcal{L}_f^{'} := \mathbb{E}_{(s,a,s')\sim\mathcal{D}_e\cup\mathcal{D}_o}[-\log f(s'|s,a)] \\ \min\limits_{d_r} \max\limits_{\pi,f} \mathcal{L}_{d_r} \\ \min\limits_{d_o} \max\limits_{\pi} \mathcal{L}_{d_o} \end{cases} \tag{24}$$

where we use the same policy learning objective $\mathcal{L}_\pi$ to make it only learn from the expert demonstrations, but use an updated objective $\mathcal{L}_f^{'}$ for the dynamics model f, as it can learn from both the real expert and suboptimal datasets $\mathcal{D}_e \cup \mathcal{D}_o$ regardless of the optimality of data. For the rollout discriminator d_r, now it needs to distinguish both the real expert and suboptimal data $\mathcal{D}_e \cup \mathcal{D}_o$ from model generated rollouts $\mathcal{D}_r$, hence we update its learning objective as follows:

$$\begin{aligned} \mathcal{L}_{d_r} = &\mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_e\cup\mathcal{D}_o}[-\log d(s,a,\log\pi(a|s),\log f(s'|s,a))] + \\ &\mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_r}[-\log(1-d(s,a,\log\pi(a|s),\log f(s'|s,a)))] \end{aligned} \tag{25}$$

For the additional optimality discriminator d_o, we follow the treatment in previous works [13, 20] to adopt a positive-unlabeled (PU) learning [35] objective, as the the unknown suboptimal dataset $\mathcal{D}_o$ may also contain some expert-like data. Utilizing PU learning allows us to learn from positive (expert data $\mathcal{D}_e$) and unlabeled data ($\mathcal{D}_o \cup \mathcal{D}_r$ in our case). The learning objective of d_o is given as:

$$\begin{aligned} \mathcal{L}_{d_o} = &\eta \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e}[-\log d_o(s,a,\log\pi(a|s))] + \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_o\cup\mathcal{D}_r}[-\log(1-d_o(s,a,\log\pi(a|s)))] \\ &- \eta \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e}[-\log(1-d_o(s,a,\log\pi(a|s)))] \end{aligned} \tag{26}$$

where η is a hyperparameter, corresponds to the proportion of positive samples to unlabeled samples. We set it as 0.5 in all our experiments.

Following a similar reformulation scheme as in DMIL, we can avoid solving the two complex functional min-max optimization problems in Eq.(24) by considering the following reformulation:

$$\begin{cases} \min_\pi \ \mathcal{L}_\pi^{\text{D2MIL}} := \alpha_\pi \cdot \mathcal{L}_\pi + \mathcal{L}_\pi^{corr} = \alpha_\pi \cdot \mathcal{L}_\pi + \beta_r \cdot \mathcal{L}_\pi^{corr_r} + \beta_o \cdot \mathcal{L}_\pi^{corr_o} \\ \min_f \ \mathcal{L}_f^{\text{D2MIL}} := \alpha_f \cdot \mathcal{L}_f^{'} + \mathcal{L}_f^{corr} \\ \min_{d_r} \ \mathcal{L}_{d_r} \\ \min_{d_o} \ \mathcal{L}_{d_o} \end{cases} \tag{27}$$

Due to the existence of two pairs of adversarial relationships involving policy π, the corrective loss term on π will become the sum of two terms, i.e., $\mathcal{L}_\pi^{corr} = \beta_r \cdot \mathcal{L}_\pi^{corr_r} + \beta_o \cdot \mathcal{L}_\pi^{corr_o}$. β_r and β_o

are the weight factors to balance the impact from both the original rollout discriminator d_r and the optimality discriminator d_o on policy π. To reduce the number of hyperparameters in the model, we set $\beta_o = 1 - \beta_r$. The derivation of the exact forms of $\mathcal{L}_\pi^{corr_r}$, $\mathcal{L}_\pi^{corr_o}$ and $\mathcal{L}_f$ under D2MIL are described below.

Corrective Loss Terms under D2MIL. Following the same derivation procedure of DMIL in Appendix A.1, the updated corrective loss terms $\mathcal{L}_f^{corr}$ and $\mathcal{L}_\pi^{corr_r}$ for dynamics model f and policy π under D2MIL can be easily obtained as follows:

$$\mathcal{L}_f^{corr} = -\mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_e\cup\mathcal{D}_o}\left[-\frac{1}{d_r}\cdot\log f(s'|s,a)\right] + \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_r}\left[-\frac{1}{1-d_r}\cdot\log f(s'|s,a)\right] \tag{28}$$

$$\mathcal{L}_\pi^{corr_r} = -\mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_e\cup\mathcal{D}_o}\left[-\frac{1}{d_r}\cdot\log \pi(a|s)\right] + \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_r}\left[-\frac{1}{1-d_r}\cdot\log \pi(a|s)\right] \tag{29}$$

While for the learning objective of discriminator d_o in Eq.(26), let $z = (s,a)$ and Ω_{sa} as its domain, then it can be rewritten as the integral of a new functional $F_o(z, d_o, \log\pi(a|s))$:

$$\begin{aligned}\mathcal{L}_{d_o} =& \int_{\Omega_{sa}}\Big[P_{D_e}(z)\cdot\eta[-\log d_o(z,\log\pi(a|s))] + (P_{D_o}(z)+P_{D_r}(z))\cdot[-\log(1-d_o(z,\log\pi(a|s)))] \\ & - P_{D_e}(z)\cdot\eta[-\log(1-d_o(z,\log\pi(a|s)))]\Big]\mathrm{d}z \\ \triangleq& \int_{\Omega_{sa}} F_o(z,d_o,\log\pi(a|s))\mathrm{d}z\end{aligned} \tag{30}$$

where $P_{D_e}(z)$, $P_{D_o}(z)$ and $P_{D_r}(z)$ are the probability distributions of z in $\mathcal{D}_e$, $\mathcal{D}_o$ and $\mathcal{D}_r$, respectively. Following the derivation in previous section, we can get the similar relaxed necessary condition for $\mathcal{L}_{d_o}$ to reach its extrema with respect to π as:

$$\begin{aligned}&\int_{\Omega_{sa}}\frac{\partial F_o(z,d_o,\log\pi(a|s))}{\partial d_o(z,\log\pi(a|s))}\cdot\nabla_{\theta_\pi}\log\pi(a|s)\mathrm{d}z \\ =&\int_{\Omega_{sa}}\Bigg[-P_{\mathcal{D}_e}(z)\cdot\frac{\eta}{d_o(z,\log\pi(a|s))} + (P_{D_o}(z)+P_{D_r}(z))\cdot\frac{1}{1-d_o(z,\log\pi(a|s))} \\ &- P_{\mathcal{D}_e}(z)\cdot\frac{\eta}{1-d_o(z,\log\pi(a|s))}\Bigg]\cdot\nabla_{\theta_\pi}\log\pi(a|s)\mathrm{d}z \\ =&\mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e}\left[-\frac{\eta}{d_o}\cdot\nabla_{\theta_\pi}\log\pi(a|s)\right] - \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_o\cup\mathcal{D}_r}\left[-\frac{1}{1-d_o}\cdot\nabla_{\theta_\pi}\log\pi(a|s)\right] \\ &+\mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e}\left[-\frac{\eta}{1-d_o}\cdot\nabla_{\theta_\pi}\log\pi(a|s)\right] = 0\end{aligned} \tag{31}$$

Again, we slightly abuse the notations and write the output values of $d_o(s,a,\log(a|s))$ as d_o in the last equation. Similar to the derivation of DMIL, above condition can be perceived as the first-order optimality condition of the corrective loss term $\mathcal{L}_\pi^{corr_o}$ with the following form:

$$\mathcal{L}_\pi^{corr_o} = -\mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_e}\left[-\frac{\eta}{d_o(1-d_o)}\cdot\log\pi(a|s)\right] + \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}_o\cup\mathcal{D}_r}\left[-\frac{1}{1-d_o}\cdot\log\pi(a|s)\right] \tag{32}$$

Plug these corrective loss terms back to the reformulated problem in Eq.(27), we obtain the final learning objectives of π and f in D2MIL:

$$\begin{aligned}\mathcal{L}_\pi^{\text{D2MIL}} =& \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_e}\left[-\left(\alpha_\pi - \frac{\beta_o\eta}{d_o(1-d_o)} - \frac{\beta_r}{d_r}\right)\cdot\log\pi(a|s)\right] + \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_o}\left[-\left(\frac{\beta_o}{1-d_o} - \frac{\beta_r}{d_r}\right)\cdot\log\pi(a|s)\right] \\ &+\mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_r}\left[-\left(\frac{\beta_o}{1-d_o} + \frac{\beta_r}{1-d_r}\right)\cdot\log\pi(a|s)\right]\end{aligned} \tag{33}$$

$$\mathcal{L}_f^{\text{D2MIL}} = \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_e\cup\mathcal{D}_o}\left[-\left(\alpha_f - \frac{1}{d_r}\right)\cdot\log f(s'|s,a)\right] + \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}_r}\left[-\frac{1}{1-d_r}\cdot\log f(s'|s,a)\right] \tag{34}$$

Again, to ensure the continuity assumption is satisfied during derivation, we clip the output range of both d_o and d_r to $[0.1, 0.9]$.

In the final objective of $\mathcal{L}_\pi^{\text{D2MIL}}$, β_o and β_r ($\beta_o + \beta_r = 1$) actually reflect the trade-off between the reliability and optimality of samples in $\mathcal{D}_o$ and $\mathcal{D}_r$. When $\beta_o = \beta_r$, D2MIL tends to learn policy with high d_o and d_r samples with similar preference. However, if the suboptimal dataset $\mathcal{D}_o$ is known to have high quality, one can use a larger β_r to pay more attention to the quality of rollout data. In such cases, both d_o and d_r will output values close to 1 on $\mathcal{D}_o$ samples, resulting high weights to encourage policy learning on these samples. Conversely, if the expert demonstrations $\mathcal{D}_e$ and suboptimal dataset $\mathcal{D}_o$ has considerably large gap, a large β_o should be used to ensure policy learning focus more on those expert-like samples.

B Algorithm and Implementation Details

B.1 Algorithm Details

We outline the pseudocode of DMIL in Algorithm 1 and D2MIL in Algorithm 2.

Algorithm 1 Discriminator-guided Model-based Offline Imitation Learning (DMIL)

Require: Expert dataset D_e, hyperparameter α_π, α_f
1: Initialize the discriminator d, dynamics model f and imitation policy π; set $\mathcal{D}_r = \emptyset$.
2: Train a preliminary dynamics model f using samples from D_e
3: **for** training step $t = 1 \cdots N$ **do**
4: Utilize dynamics model f and imitation policy π to generate rollouts and add into D_r
5: Sample $(s_e, a_e, s'_e) \sim D_e$ and $(s_r, a_r, s'_r) \sim D_r$ to form a training batch
6: Update d by minimizing the objective in Eq.(14)
7: Update π by minimizing the objective in Eq.(22)
8: Update f by minimizing the objective in Eq.(23)
9: **end for**

Algorithm 2 Dual-Discriminator Guided Model-based Offline Imitation Learning (D2MIL)

Require: Expert dataset D_e, suboptimal dataset D_o, hyperparameter α_π, α_f, β_r, β_o
1: Initialize the discriminators d_o, d_r, dynamics model f and imitation policy π; set $\mathcal{D}_r = \emptyset$.
2: Train a preliminary dynamics model f using samples from $D_e \cup D_o$
3: **for** training step $t = 1 \cdots N$ **do**
4: Utilize dynamics model f and imitation policy π to generate rollouts and add into D_r
5: Sample $(s_e, a_e, s'_e) \sim D_e$, $(s_o, a_o, s'_o) \sim D_o$ and $(s_r, a_r, s'_r) \sim D_r$ to form a training batch
6: Update d_r by minimizing the objective in Eq.(25)
7: Update d_o by minimizing the objective in Eq.(26)
8: Update π by minimizing the objective in Eq.(33)
9: Update f by minimizing the objective in Eq.(34)
10: **end for**

Hyperparameters	Values in experiments	
	D4RL tasks	Real-world tasks
DMIL-α_π	10	10
DMIL-α_f	10	10
D2MIL-α_π	10	10
D2MIL-α_f	10	10
D2MIL-η	0.5	0.5
D2MIL-β_o	0.5	0.6
D2MIL-β_r	0.5	0.4

Table 2: Hyperparemeter values.

Tasks	Transitions
MuJoCo-exp-10%	100,000
MuJoCo-exp-5%	50,000
MuJoCo-exp-2%	20,000
Pen-human	5,000
Hammer-human	11,310
Door-human	6,729
exp-med-0.3	$\mathcal{D}_e$: 7,000, $\mathcal{D}_o$: 23,000
exp-med-0.6	$\mathcal{D}_e$: 4,000, $\mathcal{D}_o$: 26,000

Table 3: Datesets details for D4RL tasks.

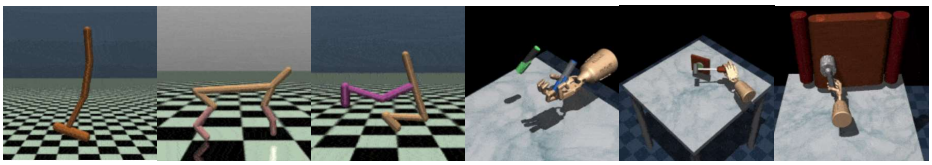
B.2 Implementation Details

For all experiments on MuJoCo tasks, all models (dynamics model f, imitation policy π, discriminator d (d_r, d_o for D2MIL)) are implemented as 2-layer neural networks with 256 hidden units each layer for dynamics model and policy, and 512 hidden units for the discriminator. For Adroit tasks, we use the same network configuration for dynamics model and discriminator, but increase the policy networks to 3 layers with 1024 hidden units due to the high dimensional state space. We use Relu activations for hidden layers and Adam optimizer. The batch size is 256, and the learning rate is $1e-4$. For discriminators, to satisfy the continuity assumption when deriving the corrective loss terms in Appendix A.1, the output is clipped to $[0.1, 0.9]$ after sigmoid activation.

For both DMIL and D2MIL, we set α_π and α_f as 10 across all tasks, which are found to achieve good performance. For D2MIL, $\eta = 0.5$ is used in all experiments, and the additional weight hyperparameters β_r and β_o are set to 0.5 in simulation experiments. In real-world experiments, due to large quality gap between the expert dataset and suboptimal human demonstrations, β_o is set to 0.6, and β_r is set to $1-\beta_o = 0.4$. Although DMIL and D2MIL contain several hyperparameters, we found them do not need careful tuning. Even using the same set of default parameters in different tasks, the model still provides good performance. We summarize these hyperparameters in Table 2 and provide evaluation and discussions on the different choices of hyperparameters in Appendix C.3.

B.3 Detailed Experiment Settings

D4RL Benchmark Experiments. In D4RL benchmark tasks under simulation environment, we use the medium and expert datasets in Mujoco and human dataset in Adroit of D4RL [29] to conduct our experiments. There are 1 million samples in each expert or medium dataset for D4RL-MuJoCo tasks. We randomly sample 10%, 5% and 2% of transitions from these MuJoCo datasets (correspond to 100,000, 50,000, 20,000 transitions) to evaluate policy performance under small datasets. For Adroit tasks, there are only 5,000, 11,310 and 6,729 transitions in human datasets for Pen, Hammer and Door tasks respectively, which are already small compared with their high dimensionality in state space. Hence we directly use the original human datasets in our experiments. To evaluate the policy robustness, we randomly pick 20% samples from previous constructed datasets and add a Gaussian noise with $\mathcal{N}(0, \sigma^2)$ on the states, where σ stands for the standard deviation of each dimension of states in training dataset. As for the evaluation on D2MIL, we first sample 1% trajectories (10,000 transitions) from D4RL-MuJoCo expert datasets, then sample X proportion of these trajectories and combine them with 2% medium dataset (20,000 transitions) to constitute the suboptimal dataset $\mathcal{D}_o$. The remaining 1-X trajectories constitute the expert dataset $\mathcal{D}_e$. We term each task in different environments as exp-med-X. Detailed statistics of the datasets used in the experiments are summarized in Table 3.



(a) Hopper (b) Halfcheetah (c) Walker (d) Pen (e) Hammer (f) Door

Figure 7: Simulated tasks in D4RL benchmarks.

Real-world Experiments. For real-world validation, we deploy our methods and baselines on a wheel-legged robot. The control action is the sum of the torque τ of the motors at the two wheels ($\frac{\tau}{2}$ for each). The control frequency of the robot is 200Hz. We elaborate the two task settings as follows:

(1) **Standing still**: The state space of the robot is represented by $\mathbf{s} = (\theta, \dot{\theta}, x, \dot{x})$, where θ denotes the forward tilt angle of the body, x is the displacement of the robot, $\dot{\theta}$ is angular velocity, and $\dot{x}$ is linear velocity. We collect datasets containing human controlled transitions of $(\mathbf{s}, a, \mathbf{s}', r, d)$, where $\mathbf{s}$ is the current state, a is the torque of motors, $\mathbf{s}'$ is the next state, r is the reward and d is the flag of terminal. During performance evaluation, we run all algorithms for 200,000 training steps and report the final results in the main text.

Table 4: Normalized scores for models trained on different proportion of D4RL MuJoCo-medium datasets. Results are averaged over 3 random seeds.

	Ratio	BC	BC+d	2-phase BC+d	valueDICE	IQ-Learn	DMIL
Hopper-med	10%	46.26±8.69	47.55±7.56	48.55±7.30	**53.96±5.48**	47.01±5.59	**53.72±8.78**
	5%	43.31±8.81	45.19±7.86	46.47±7.13	52.43±8.92	43.88±5.67	**52.81±8.47**
	2%	41.35±8.38	41.44±6.51	46.07±6.87	51.43±6.48	25.42±3.02	**52.89±8.42**
Halfcheetah-med	10%	41.58±1.69	41.12±1.49	41.35±2.23	40.81±2.32	40.36±1.92	**41.86±2.19**
	5%	40.46±2.61	40.47±1.65	41.15±2.31	40.23±2.46	36.66±4.27	**42.19±2.56**
	2%	36.29±5.71	34.59±5.91	39.37±3.46	37.21±1.89	27.45±8.24	**41.26±1.61**
Walker2d-med	10%	66.14±16.54	66.25±15.54	68.08±15.28	47.11±3.55	54.28±11.74	**71.66±12.51**
	5%	62.62±19.84	64.38±18.97	64.95±18.13	37.86±8.99	13.57±8.28	**67.51±15.75**
	2%	44.84±25.50	47.82±25.39	59.52±21.00	33.35±6.11	5.87±4.24	**62.25±17.05**

(2) **Moving straight**: The state space in this task is represented by $\mathbf{s} = (\theta, \dot{\theta}, \dot{x})$, without the forementioned displacement x since we only want to keep the velocity of the robot stable. Datasets contain human controlled transitions of $(\mathbf{s}, a, \mathbf{s}', r, d)$ when the robot moves forward. Our goal is to keep the robot at the target speed of 0.2m/s. During performance evaluation, we run all algorithm for 200,000 training steps and report the final results in the main text.

For each of the above two tasks, we collect 10,000 transition data from human demonstrations, which are about 50 second human control. As the actual control frequency of the robot is high (200Hz), human demonstrations can only be perceived as mediocre or suboptimal data. To evaluate the performance of D2MIL, we additionally collect very few transitions (140 transitions, less than 1 second's control) generated by a high quality Linear Quadratic Regulator (LQR) policy for the standing still task. We use such very small amount of expert data combined with human demonstrations to evaluate and compare the performance of D2MIL against baseline methods.

C Additional Experiment Results

C.1 Additional Comparative Evaluation Results

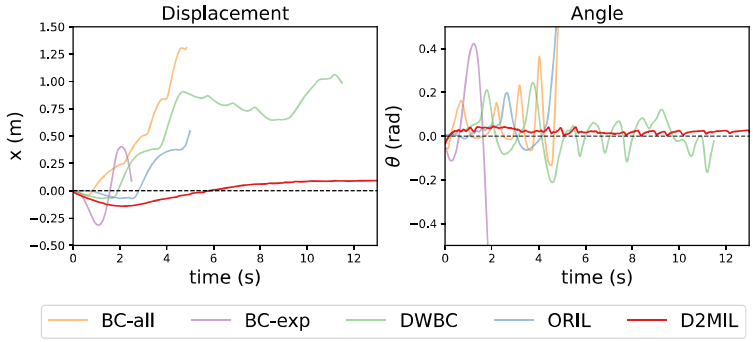
Simulation Experiments on D4RL-MuJoCo Medium Datasets. We also evaluate DMIL on D4RL-MuJoCo medium-quality datasets, which are generated from a policy trained to approximately 1/3 the performance of an expert policy. The comparative results are shown in Table 4. Due to the suboptimality in medium datasets, the gap between different methods is not as large as the experiments on expert data (Table 1 in the main text). However, we can still observe that DMIL consistently outperforms other baselines in all tasks.

Real-world Experiments for Scenarios with Additional Suboptimal Dataset. We also conduct real-world experiments on standing still task for D2MIL. In this setting, we collect 140 transitions generated from a high quality LQR expert policy. In particular, we consider two different sizes of expert dataset $\mathcal{D}_e$, one contains all the 140 transitions, the other contains only 1/10 of the data, 14 transitions. We also sample 5,000 transitions from the human demonstrations to constitute the suboptimal dataset $\mathcal{D}_o$. The amount of expert data, especially the second case, is extremely small compared with the suboptimal data, which requires the IL algorithm to maximally extract information from the suboptimal datset $\mathcal{D}_o$ for policy learning.

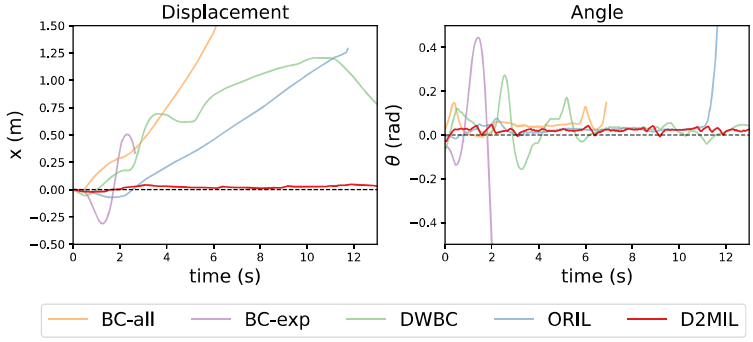
The evaluation results are shown in Figure 8. Robot trained with BC-all, BC-exp and ORIL polices cannot maintain balance in both task settings. Although robot trained with DWBC can maintain a rather stable tilt angle, it fails to stay still and shows a slight drift. While for D2MIL, robot can stay in place and keep balance at the same time, indicating superior performance over other baselines.

C.2 Ablation on the Cooperative-yet-Adversarial Learning Scheme.

We conduct ablation study on D4RL-MuJoCO expert datasets to examine the benefits of introducing the proposed cooperative-yet-adversarial learning scheme in DMIL. This scheme has two ingredients, first is the incorporating element-wise loss information $\log \pi$ and $\log f$ into the discriminator d to



(a) D2MIL trained on 140 expert transitions and 5,000 suboptimal human demonstration transitions.



(b) D2MIL trained on 14 expert transitions and 5,000 suboptimal human demonstration transitions.

Figure 8: Evaluation results of D2MIL on the standing still task on the real-world wheel-legged robot

establish cooperative information sharing; the second is adding adversarial learning strategy between both π and f against d. To examine the impact of these ingredients, we evaluate the following baselines or variants of DMIL on MuJoCo expert and 20% state noise datasets:

- **DMIL-no-d-adv**: removing the coupling and the adversarial relationship between discriminator d and dynamics model f. In this variant, we remove both the additional information $\log f$ from the inputs of d, as well as the corrective loss term $\mathcal{L}_f^{corr}$ of f to remove its adversarial behavior on d.

- **DMIL-no-d-adv&π-info**: on the basis of DMIL-no-d-adv, we further remove the additional information $\log \pi$ from the inputs of d. This removes the cooperative information sharing in DMIL, but we keep the corrective loss term $\mathcal{L}_\pi^{corr}$ of π to enable discriminator-guided policy learning.

- **2-phase BC+d**: this baseline can be perceived as the reduction of DMIL that completely removes the cooperative-yet-adversarial learning scheme.

- **BC** and **BC+d**: minimal baselines without or with a dynamics model used for comparison.

The results are presented in Table 5. From the results, we can see that without the cooperative-yet-adversarial learning scheme (BC, BC+d, 2-phase BC+d), the performance of imitation policy degenerates significantly on small datasets. When incorporating the adversarial relationship between policy π and discriminator d (DMIL, DMIL-no-d-adv&π-info, DMIL-no-d-adv), the performance of policy is substantially improved under small dataset. As for DMIL-no-d-adv and DMIL-no-d-adv&π-info that remove adversarial relationship between f and d, they have similar performance with DMIL when the training data are sufficient, but suffer from noticeable performance drop when dataset is extremely small or contains noisy inputs. On the contrary, DMIL can maintain nearly the same performance with reduced datasets as well as involvement of noisy data. Therefore, we can see that the cooperative-yet-adversarial learning scheme involving π, f and d indeed help with improving policy robustness and imitation performance.

Table 5: Ablation study of DMIL on different proportion of D4RL-MuJoCo expert and 20% state noise datasets.

	ratio	BC	BC+d	2-phase BC+d	DMIL-no-d-adv&π-info	DMIL-no-d-adv	DMIL
Hopper	10%	83.52±30.58	100.59±13.21	104.35±9.44	110.58±1.26	110.14±1.92	**111.56±1.51**
	5%	73.35±37.04	94.82±19.72	99.66±14.98	109.26±2.51	108.44±4.49	**111.14±1.83**
	2%	53.54±36.89	61.57±30.18	88.24±25.63	105.45±10.46	103.99±11.26	**108.51±3.88**
Halfcheetah	10%	90.64±2.21	89.71±2.88	71.27±19.33	92.38±2.69	92.22±2.42	**92.69±1.82**
	5%	82.90±11.71	76.40±16.94	70.89±23.06	88.19±7.77	88.26±6.46	**90.18±4.43**
	2%	23.58±16.36	21.48±16.86	57.48±25.63	59.79±28.56	53.71±28.70	**76.87±15.31**
Walker2d	10%	105.36±4.38	107.61±1.14	106.40±1.96	107.68±0.91	**108.29±1.13**	107.62±0.83
	5%	103.21±7.81	105.42±3.93	104.51±4.54	107.11±1.02	106.30±1.36	**107.89±0.71**
	2%	58.34±35.86	60.64±35.10	86.71±21.20	101.40±10.76	103.76±5.43	**105.55±4.42**
Hopper+noise	10%	74.28±29.69	75.66±31.14	100.32±15.21	106.84±7.57	107.79±6.07	**110.17±1.95**
	5%	66.71±30.23	71.48±30.98	93.21±22.28	104.98±10.02	105.64±6.23	**109.62±3.02**
	2%	47.86±29.18	43.56±29.12	59.63±33.40	101.21±15.43	98.76±18.37	**108.47±4.78**
Halfcheetah+noise	10%	84.90±7.58	86.84±4.96	71.56±23.06	88.17±7.51	88.13±7.93	**88.42±6.88**
	5%	68.63±20.45	66.87±21.61	67.46±25.85	74.76±18.82	73.38±20.94	**74.56±19.24**
	2%	58.21±24.17	23.79±22.31	61.74±23.08	64.83±27.91	65.58±26.11	**73.14±18.01**
Walker2d+noise	10%	104.28±5.69	97.21±16.99	102.84±8.37	107.01±2.03	105.40±5.71	**107.94±0.64**
	5%	89.84±20.52	91.86±23.72	97.38±15.87	103.39±7.85	100.61±12.79	**105.89±3.92**
	2%	66.98±37.23	74.76±35.07	92.01±22.61	92.22±22.76	95.13±23.93	**103.54±6.98**

C.3 Discussion and Evaluations on Different Choices of Hyperparameters

In the proposed DMIL, the hyperparameters involved are α_π and α_f, which are used to balance the impact of correction loss terms. In general cases, we can simply choose $\alpha_\pi = \alpha_f > 1$. In all our experiments, the values of α_π and α_f are set to be 10 without tuning (see Table 2), as we find this choice already produces good model performance. To further verify their impact, we conducted additional experiments on Hopper tasks with 2% expert data by setting α_π and α_f to different values, the results are presented below. It is found that these hyperparameters generally do not need careful tuning and produce similar performance.

Table 6: Experimental results for different values of hyperparameters in DMIL

α_π, α_f	5	10	20
Results	106.07±7.86	108.51±3.88	105.79±8.61

For the extended D2MIL, although we have hyperparemeters α_π, α_f, η, β_o and β_r in the model, most of them do not need to be tuned. As in DMIL, we set $\alpha_\pi = \alpha_f = 10$. We adopt $\eta = 0.5$ as a constant, which is same as in ORIL [13] and DWBC [20]. In our implementation, we make $\beta_o + \beta_r = 1$ to reduce the parameter numbers. β_o and β_r reflect the trade-off between the reliability and optimality of samples in the suboptimal dataset $\mathcal{D}_o$ and rollout data $\mathcal{D}_r$. The detailed discussion on the impact of β_o and β_r is presented in the last paragraph of Appendix A.2. In practical scenarios, we suggest the practitioners just setting $\beta_r = \beta_o = 0.5$, which generally leads to reasonably good performance. In our real-world experiments, due to the large quality gap between the expert dataset and suboptimal human demonstrations, we set β_o to be slightly larger value ($\beta_o = 0.6$, $\beta_r = 1 - \beta_o = 0.4$).

Although DMIL and D2MIL contain several hyperparameters, the associated hyperparameter tuning effort during practical use is actually very minor. We use the same set of hyperparameters in most of our experiments without tuning. Moreover, we found that using the default hyperparameter values summarized in Table 2 in most cases lead to good performance. This can be a particularly nice feature for DMIL and D2MIL in practical applications.

C.4 Co-evolution of Models During the Learning Process

To get a better understanding of our cooperative-yet-adversarial learning scheme in DMIL, we plot the TSNE visualization of generated model rollouts at different model training stages together with the original expert data in Figure 9 and 10. Moreover, we also plot the discriminator output values on these generated rollouts to examine how do the policy, dynamics model and discriminator co-evolve during training. We find that at the initial stage, the generated rollouts are inconsistent with expert data due to less well-learned policy, and the discriminator d is also incapable of discriminating the credibility of samples, which outputs around 0.5 for every rollout sample. As the training process continues and the policy is learned better, we can see that the generated rollouts start to align with the expert data, and the discriminator tends to believe most rollout data are reliable ($d \to 1$). However, at the later stage of training, as the discriminator is learned to be stronger, it can identify most of the generated rollouts are fake data ($d \to 0$). Under this stage, the policy will receive high learning weights only on few highly reliable samples, and the final imitation performance (illustrated as the average return scores in Figure 9 and 10) is gradually saturated.

It is intriguing that above co-evolution pattern is almost universal across tasks, as observed in both Halfcheetah and Walker2d tasks with 5% expert data. It is also worth noting that such a co-evolution pattern is very different from typical GAN-like methods. As in these approaches, the generator will eventually become stronger and the discriminator cannot tell whether the generated samples are real or fake (i.e., $d \to 0.5$). In DMIL, the discriminator d can generally learned to be stronger compared with those in GAN-like method, due to additional cooperative information shared from π and f (i.e., adding $\log \pi$ and $\log f$ to the input of d). Moreover, since both π and f also optimize their own objectives in addition to enforcing the adversarial behavior on d, it is more likely the discriminator in DMIL can eventually distinguish most of the generated rollouts as fake. When such phenomenon occurs, it also suggests the saturation or convergence of the learning process.

C.5 Learning Curves

The learning curves on D4RL benchmark tasks for DMIL are shown in Figure 11.

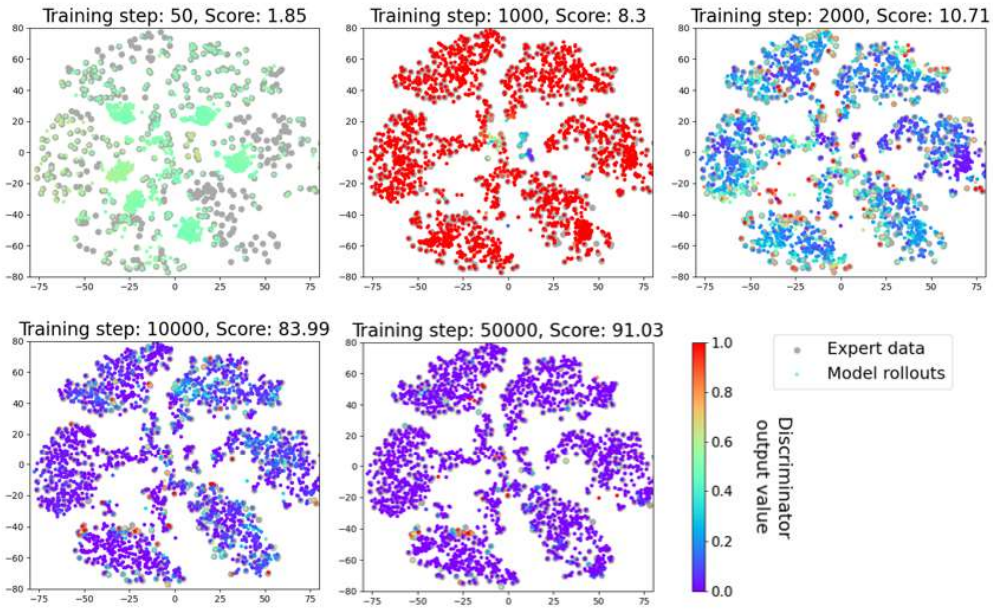


Figure 9: TSNE visualization of the expert data and the generated rollout data under different stages of training on the Halfcheetah-5% task. The color of rollouts points indicates the output value of the discriminator.

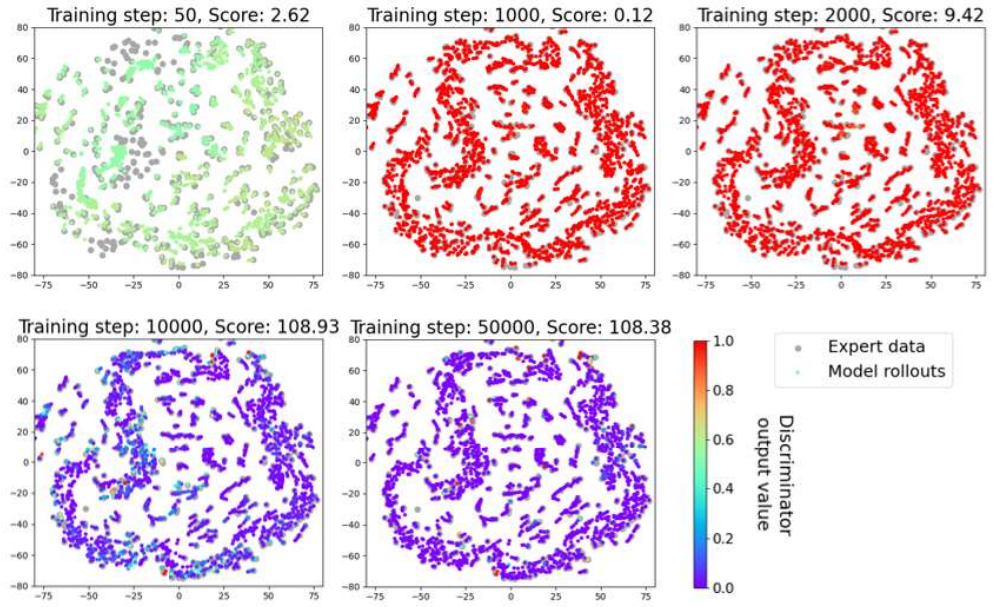


Figure 10: TSNE visualization of the expert data and the generated rollout data under different stages of training on the Walker2d-5% task. The color of rollouts points indicates the output value of the discriminator.

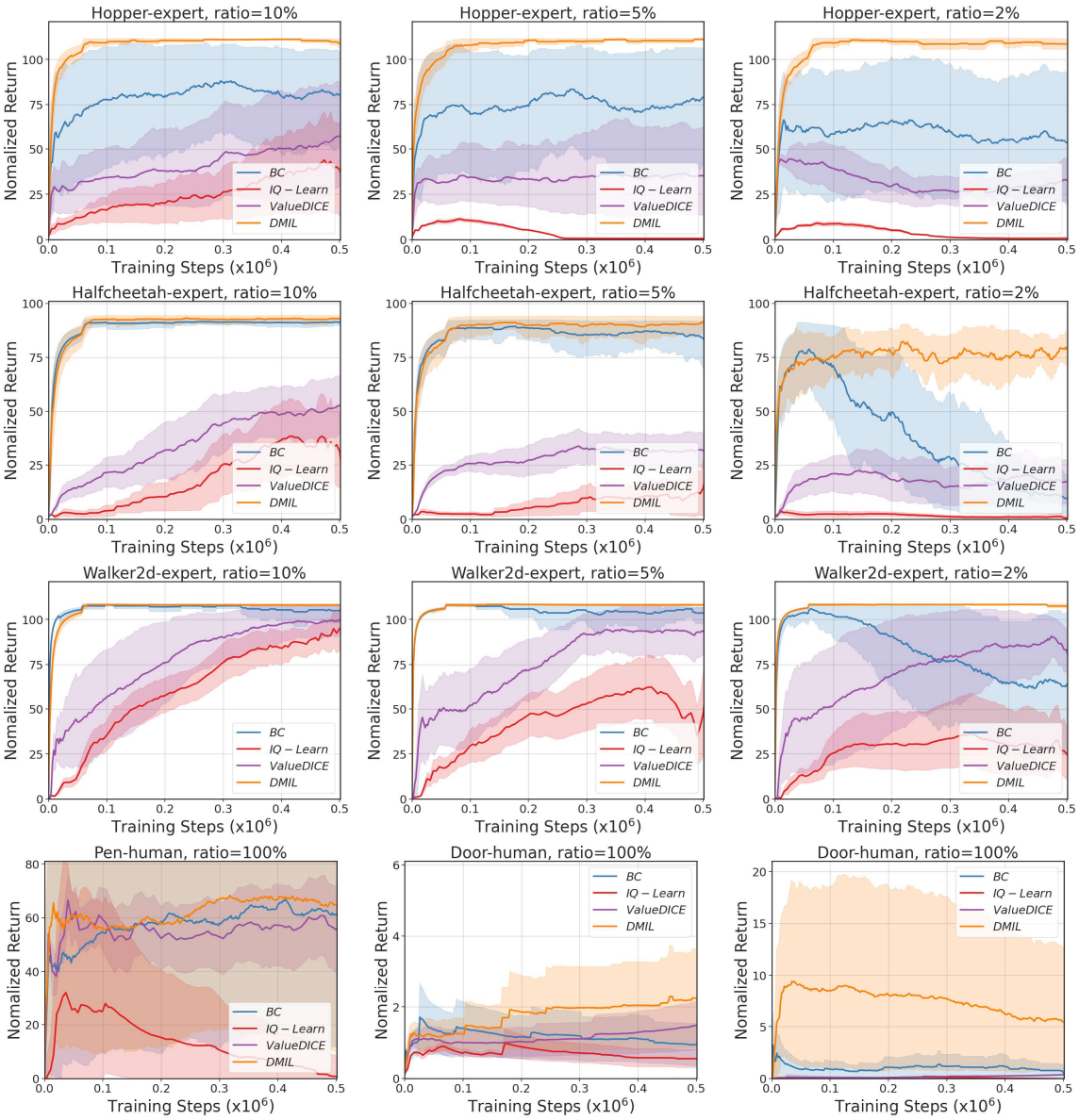


Figure 11: Learning curves of DMIL on D4RL benchmark tasks.