A Algorithm Description

The pseudocode of *VGDF* is presented in Algorithm 1. We utilize SAC [20] as our backbone algorithm. We employ a fixed entropy temperature coefficient in all experiments, demonstrating sufficient empirical performance. The training of the dynamics model ensemble follows prior works [8, 26] with the MLE loss. The calculation of the Fictitious Value Proximity follows Eq. (6) proposed in Section 5.1. Furthermore, the pseudocode of *VGDF + BC* is presented in Algorithm 2. We introduce the value-normalized tradeoff between the behavior cloning loss and the policy gradient following the prior work [18].

Algorithm 1 Value-Guided Data Filtering (VGDF)

Input: Source domain $\mathcal{M}_{src}$, target domain $\mathcal{M}_{tar}$, and transition ratio Γ $(= 10)$ (source vs. target).
Initialization: Policy π, exploration policy π^{E}, value functions $\{Q_{\theta_i}\}_{i=1,2}$, replay buffers $\{D_{src}, D_{tar}\}$, dynamics model ensemble $\{T_{\phi_i}\}_{i=1}^{M}$, data selection ratio ξ, batch size B, entropy temperature coefficient λ.

for t = 1, 2, ... **do**
  `# Interact with the source domain`
  Sample transition $(s_{src}, a_{src}, r_{src}, s'_{src})$ using π^{E} in $\mathcal{M}_{src}$
  $D_{src} \leftarrow D_{src} \cup (s_{src}, a_{src}, r_{src}, s'_{src})$
  `# Interact with the target domain`
  if $t \,\%\, \Gamma == 0$ **then**
    Sample transition $(s_{tar}, a_{tar}, r_{tar}, s'_{tar})$ using π in $\mathcal{M}_{tar}$
    $D_{tar} \leftarrow D_{tar} \cup (s_{tar}, a_{tar}, r_{tar}, s'_{tar})$
  end if
  Optimize dynamics ensemble $\{T_{\phi_i}\}_{i=1}^{M}$ with D_{tar} via Eq. (13)
  Sample $b_{src} := \{(s, a, r, s')\}_{src}^{B}$ from D_{src}
  Sample $b_{tar} := \{(s, a, r, s')\}_{tar}^{B}$ from D_{tar}
  Obtain Fictitious Value Proximity (FVP) $\{\Lambda(s, a, s')\}^{B}$ via Eq. (6) for transitions in b_{src}
  Obtain FVP quantile $\Lambda_{\xi\%}$ of $\{\Lambda(s, a, s')\}^{B}$
  `# Optimize value function with data filtering`
  $\theta_{i=1,2} \leftarrow \arg\min_{\theta_i} \frac{1}{2B} \sum_{b_{tar}} \left[\left(Q_{\theta_i} - \mathcal{T}Q_{\theta_i}\right)^2 \right] +$
  $\frac{1}{\lfloor 2B \cdot \xi\% \rfloor} \sum_{b_{src}} \left[\mathbb{1}\left(\Lambda(s, a, s') > \Lambda_{\xi\%}\right) \left(Q_{\theta_i} - \mathcal{T}Q_{\theta_i}\right)^2 \right]$
  `# Optimize policies`
  $\pi^{\mathrm{E}} \leftarrow \arg\max_{\pi^{\mathrm{E}}} \frac{1}{2B} \sum_{b_{tar} \cup b_{src}} \left[\max\left\{Q_{\theta_1}(s, a), Q_{\theta_2}(s, a)\right\} |_{a \sim \pi^{\mathrm{E}}(\cdot|s)} + \lambda \mathcal{H}[\pi^{\mathrm{E}}] \right]$
  $\pi \leftarrow \arg\max_{\pi} \frac{1}{2B} \sum_{b_{tar} \cup b_{src}} \left[\min\left\{Q_{\theta_1}(s, a), Q_{\theta_2}(s, a)\right\} |_{a \sim \pi(\cdot|s)} + \lambda \mathcal{H}[\pi] \right]$
end for

B Proofs of the Performance Guarantees

This section presents the proof of our main results. Specifically, we propose that the value discrepancy can be leveraged for the performance guarantee across different domains Lemma C.3. In Theorem B.1, we convert the performance bound induced by the value discrepancy into a novel form for the offline source domain setting.

Theorem B.1. (Performance bound controlled by dynamics discrepancy.) *Denote the source domain and target domain with different dynamics as $\mathcal{M}_{src}$ and $\mathcal{M}_{tar}$, respectively. We have the*

Algorithm 2 Value-Guided Data Filtering + Behavior Cloning (VGDF + BC)

Input: Source domain offline dataset D_{src}, target domain $\mathcal{M}_{tar}$, max interaction steps with the target domain $T_{\max}$, and transition ratio Γ ($:= \frac{|D_{src}|}{T_{\max}} = 10$) (source vs. target).
Initialization: Policy π, value functions $\{Q_{\theta_i}\}_{i=1,2}$, target domain replay buffer D_{tar}, dynamics model ensemble $\{T_{\phi_i}\}_{i=1}^{M}$, data selection ratio ξ, batch size B, entropy temperature coefficient λ, train repeat K, behavior cloning constant α.

for t = 1, 2, ..., $T_{\max}$ **do**
  `# Interact with the target domain`
  Sample transition $(s_{tar}, a_{tar}, r_{tar}, s'_{tar})$ using π in $\mathcal{M}_{tar}$
  $D_{tar} \leftarrow D_{tar} \cup (s_{tar}, a_{tar}, r_{tar}, s'_{tar})$
  `# Repeat training for` K `times per step`
  for k = 1, 2, ..., K **do**
    Optimize dynamics ensemble $\{T_{\phi_i}\}_{i=1}^{M}$ with D_{tar} via Eq. (13)
    Sample $b_{src} := \{(s, a, r, s')\}_{src}^{B}$ from D_{src}
    Sample $b_{tar} := \{(s, a, r, s')\}_{tar}^{B}$ from D_{tar}
    Obtain Fictitious Value Proximity (FVP) $\{\Lambda(s, a, s')\}^{B}$ via Eq. (6) for transitions in b_{src}
    Obtain FVP quantile $\Lambda_{\xi\%}$ of $\{\Lambda(s, a, s')\}^{B}$
    `# Optimize value function with data filtering`
    $$\theta_{i=1,2} \leftarrow \arg\min_{\theta_i} \frac{1}{2B} \sum_{b_{tar}} \left[\left(Q_{\theta_i} - \mathcal{T} Q_{\theta_i} \right)^2 \right] + \frac{1}{\lfloor 2B \cdot \xi\% \rfloor} \sum_{b_{src}} \left[\mathbb{1}\left(\Lambda(s, a, s') > \Lambda_{\xi\%} \right) \left(Q_{\theta_i} - \mathcal{T} Q_{\theta_i} \right)^2 \right]$$
    `# Optimize policy with behavior cloning regularization`
    $$\beta = \alpha / \left\{ \tfrac{1}{2B} \sum_{b_{tar} \cup\, b_{src}} \left[\left| \min \left\{ Q_{\theta_1}(s, a), Q_{\theta_2}(s, a) \right\}_{a \sim \pi(\cdot|s)} \right| \right] \right\}$$
    $$\pi \leftarrow \arg\max_{\pi} \frac{\beta}{2B} \sum_{b_{tar} \cup\, b_{src}} \left[\min \left\{ Q_{\theta_1}(s, a), Q_{\theta_2}(s, a) \right\}_{a \sim \pi(\cdot|s)} + \lambda \mathcal{H}[\pi] \right] - \frac{1}{B} \sum_{(s,a) \sim b_{src}} \left[\left(\pi(s) - a \right)^2 \right]$$
  end for
end for

performance difference of any policy π evaluated under $\mathcal{M}_{src}$ and $\mathcal{M}_{tar}$ be bounded as below,

$$\eta_{\mathcal{M}_{tar}}(\pi) \geq \eta_{\mathcal{M}_{src}}(\pi) - \frac{2\gamma r_{\max}}{(1-\gamma)^2} \cdot \mathbb{E}_{\rho^{\pi}_{src}} \left[D_{\mathrm{TV}} \left(P_{src}(\cdot|s, a) \| P_{tar}(\cdot|s, a) \right) \right].$$

Proof. We have

$$\begin{aligned}
\eta_{src}(\pi) - \eta_{tar}(\pi) &= \frac{\gamma}{1-\gamma} \mathbb{E}_{\rho^{\pi}_{src}(s,a)} \left[\int_{s'} P_{src}(s'|s, a) V^{\pi}_{tar}(s') - \int_{s'} P_{tar}(s'|s, a) V^{\pi}_{tar}(s') ds' \right] \quad \text{(Lemma C.1)} \\
&= \frac{\gamma}{1-\gamma} \mathbb{E}_{\rho^{\pi}_{src}(s,a)} \left[\int_{s'} \left(P_{src}(s'|s, a) - P_{tar}(s'|s, a) \right) V^{\pi}_{tar}(s') ds' \right] \\
&\leq \frac{\gamma}{1-\gamma} \mathbb{E}_{\rho^{\pi}_{src}(s,a)} \left[\int_{s'} \left| \left(P_{src}(s'|s, a) - P_{tar}(s'|s, a) \right) V^{\pi}_{tar}(s') \right| ds' \right] \\
&\leq \frac{\gamma}{1-\gamma} \cdot \frac{r_{\max}}{1-\gamma} \mathbb{E}_{\rho^{\pi}_{src}(s,a)} \left[\int_{s'} \left| P_{src}(s'|s, a) - P_{tar}(s'|s, a) \right| ds' \right] \\
&= \frac{2\gamma r_{\max}}{(1-\gamma)^2} \mathbb{E}_{\rho^{\pi}_{src}(s,a)} \left[D_{\mathrm{TV}} \left(P_{src}(\cdot|s, a) \| P_{tar}(\cdot|s, a) \right) \right]. \qquad (9)
\end{aligned}$$

□

Theorem B.2. (Performance bound controlled by value difference.) *Denote the source domain and target domain as $\mathcal{M}_{src}$ and $\mathcal{M}_{tar}$, respectively. We have the performance guarantee of any policy π over the two MDPs:*

$$\eta_{\mathcal{M}_{tar}}(\pi) \geq \eta_{\mathcal{M}_{src}}(\pi) - \frac{\gamma}{1-\gamma} \cdot \mathbb{E}_{\rho^{\pi}_{\mathcal{M}_{src}}} \left[\left| \mathbb{E}_{P_{src}} \left[V^{\pi}_{\mathcal{M}_{tar}}(s') \right] - \mathbb{E}_{P_{tar}} \left[V^{\pi}_{\mathcal{M}_{tar}}(s') \right] \right| \right].$$

Proof. We have

$$\begin{aligned}
\eta_{src}(\pi) - \eta_{tar}(\pi) &= \frac{\gamma}{1-\gamma} \mathbb{E}_{\rho^{\pi}_{src}(s,a)} \left[\int_{s'} P_{src}(s'|s,a) V^{\pi}_{tar}(s') - \int_{s'} P_{tar}(s'|s,a) V^{\pi}_{tar}(s') ds' \right] \quad \text{(Lemma C.1)} \\
&= \frac{\gamma}{1-\gamma} \mathbb{E}_{\rho^{\pi}_{src}(s,a)} \left[\int_{s'} (P_{src}(s'|s,a) - P_{tar}(s'|s,a)) V^{\pi}_{tar}(s') ds' \right] \\
&\leq \frac{\gamma}{1-\gamma} \mathbb{E}_{\rho^{\pi}_{src}(s,a)} \left[\left| \int_{s'} (P_{src}(s'|s,a) - P_{tar}(s'|s,a)) V^{\pi}_{tar}(s') ds' \right| \right] \\
&= \frac{\gamma}{1-\gamma} \cdot \mathbb{E}_{\rho^{\pi}_{\mathcal{M}_{src}}} \left[\left| \mathbb{E}_{P_{src}} \left[V^{\pi}_{\mathcal{M}_{tar}}(s') \right] - \mathbb{E}_{P_{tar}} \left[V^{\pi}_{\mathcal{M}_{tar}}(s') \right] \right| \right]
\end{aligned}$$

□

Theorem B.3. *Under the setting with offline source domain dataset D whose empirical estimation of the data collection policy is $\pi_D(a|s) := \frac{\sum_D \mathbb{1}(s,a)}{\sum_D \mathbb{1}(s)}$, let $\mathcal{M}_{src}$ and $\mathcal{M}_{tar}$ denote the source and target domain, respectively. We have the performance guarantee of any policy π over the two MDPs:*

$$\eta_{\mathcal{M}_{tar}}(\pi) \geq \eta_{\mathcal{M}_{src}}(\pi) - \frac{4\gamma r_{\max}}{(1-\gamma)^2} \mathbb{E}_{\rho^{\pi_D}_{\mathcal{M}_{src}}, P_{src}} \left[D_{TV}(\pi_D || \pi) \right] - \frac{\gamma}{1-\gamma} \mathbb{E}_{\rho^{\pi_D}_{\mathcal{M}_{src}}} \left[\left| \zeta(s,a) \right| \right], \quad (10)$$

where $\zeta(s,a) := \mathbb{E}_{P_{src},\pi} \left[Q^{\pi}_{\mathcal{M}_{tar}}(s',a') \right] - \mathbb{E}_{P_{tar},\pi} \left[Q^{\pi}_{\mathcal{M}_{tar}}(s',a') \right]$.

Proof. We have

$$\eta_{\mathcal{M}_{tar}}(\pi) - \eta_{\mathcal{M}_{src}}(\pi) = \underbrace{\left(\eta_{\mathcal{M}_{src}}(\pi_D) - \eta_{\mathcal{M}_{src}}(\pi) \right)}_{(a)} - \underbrace{\left(\eta_{\mathcal{M}_{src}}(\pi_D) - \eta_{\mathcal{M}_{tar}}(\pi) \right)}_{(b)}.$$

According to Lemma C.2, we have

$$\begin{aligned}
\eta_{\mathcal{M}_{src}}(\pi_D) - \eta_{\mathcal{M}_{src}}(\pi) &\geq -\frac{1}{1-\gamma} \mathbb{E}_{\substack{s,a \sim \rho^{\pi_D}_{\mathcal{M}_{src}} \\ s' \sim P_{src}(\cdot|s,a)}} \left[\left| \mathbb{E}_{a' \sim \pi_D(\cdot|s')} \left[Q^{\pi}_{\mathcal{M}_{src}}(s',a') \right] - \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[Q^{\pi}_{\mathcal{M}_{src}}(s',a') \right] \right| \right] \\
&= -\frac{1}{1-\gamma} \mathbb{E}_{\substack{s,a \sim \rho^{\pi_D}_{\mathcal{M}_{src}} \\ s' \sim P_{src}(\cdot|s,a)}} \left[\left| \sum_{\mathcal{A}} \left(\pi_D(a'|s') - \pi(a'|s') \right) Q^{\pi}_{\mathcal{M}_{src}}(s',a') \right| \right] \\
&\geq -\frac{1}{1-\gamma} \mathbb{E}_{\substack{s,a \sim \rho^{\pi_D}_{\mathcal{M}_{src}} \\ s' \sim P_{src}(\cdot|s,a)}} \left[\left| \sum_{\mathcal{A}} \left(\pi_D(a'|s') - \pi(a'|s') \right) \frac{r_{\max}}{1-\gamma} \right| \right] \\
&\geq -\frac{r_{\max}}{(1-\gamma)^2} \mathbb{E}_{\substack{s,a \sim \rho^{\pi_D}_{\mathcal{M}_{src}} \\ s' \sim P_{src}(\cdot|s,a)}} \left[\sum_{\mathcal{A}} \left| \pi_D(a'|s') - \pi(a'|s') \right| \right] \\
&= -\frac{2 r_{\max}}{(1-\gamma)^2} \mathbb{E}_{\substack{s,a \sim \rho^{\pi_D}_{\mathcal{M}_{src}} \\ s' \sim P_{src}(\cdot|s,a)}} \left[D_{TV} \left(\pi_D(\cdot|s') \parallel \pi(\cdot|s') \right) \right],
\end{aligned}$$

and

$$
\begin{aligned}
&-\left(\eta_{\mathcal{M}_{src}}(\pi_D)-\eta_{\mathcal{M}_{tar}}(\pi)\right)\\
&=-\frac{\gamma}{1-\gamma}\mathbb{E}_{s,a\sim\rho^{\pi_D}_{\mathcal{M}_{src}}}\left[\mathcal{G}^{\pi_1,\pi_2}_{\mathcal{M}_1,\mathcal{M}_2}(s,a)\right]\\
&\geq-\frac{2\gamma r_{max}}{(1-\gamma)^2}\mathbb{E}_{\substack{s,a\sim\rho^{\pi_D}_{\mathcal{M}_{src}}\\ s'\sim P_{src}(\cdot|s,a)}}\left[D_{TV}(\pi_D(\cdot|s')\parallel\pi(\cdot|s'))\right]\\
&\quad-\frac{\gamma}{1-\gamma}\mathbb{E}_{s,a\sim\rho^{\pi_D}_{\mathcal{M}_{src}}}\left[\left|\mathbb{E}_{s',a'\sim P_{src},\pi}\left[Q^{\pi}_{\mathcal{M}_{tar}}(s',a')\right]-\mathbb{E}_{s',a'\sim P_{tar},\pi}\left[Q^{\pi}_{\mathcal{M}_{tar}}(s',a')\right]\right|\right]. \text{ (Lemma C.3)}
\end{aligned}
$$

Combining the two inequalities above completes the proof. □

C Proofs of Lemmas

This section provides proof of several lemmas used for our theoretical results. The first lemma is adopted from [40], and the proof is essentially the same as the original paper. Lemma C.2 and Lemma C.3 support the derivation of the performance difference bound in Theorem B.3.

Lemma C.1. (Telescoping Lemma, Lemma 4.3 in [40].) Let $\mathcal{M}_1:=(\mathcal{S},\mathcal{A},P_1,r,\gamma)$ and $\mathcal{M}_2:=(\mathcal{S},\mathcal{A},P_2,r,\gamma)$ be two MDPs with different dynamics P_1 and P_2. Given a policy π, let

$$\mathcal{G}^{\pi}_{\mathcal{M}_1,\mathcal{M}_2}(s,a):=\mathbb{E}_{s'\sim P_1}\left[V^{\pi}_{\mathcal{M}_2}(s')\right]-\mathbb{E}_{s'\sim P_2}\left[V^{\pi}_{\mathcal{M}_2}(s')\right],$$

we have

$$\eta_{\mathcal{M}_1}(\pi)-\eta_{\mathcal{M}_2}(\pi)=\frac{\gamma}{(1-\gamma)}\mathbb{E}_{s,a\sim\rho^{\pi}_{\mathcal{M}_1}}\left[\mathcal{G}^{\pi}_{\mathcal{M}_1,\mathcal{M}_2}(s,a)\right].$$

Proof. Define W_j as the expected return when executing π on $\mathcal{M}_1$ for the first j steps, then switching to π and $\mathcal{M}_2$ for the remainder. That is

$$W_j:=\sum_{t=0}^{\infty}\gamma^t\mathbb{E}_{\substack{t<j:s_t,a_t\sim P_1,\pi\\ t\geq j:s_t,a_t\sim P_2,\pi_2}}\left[r(s_t,a_t)\right]=\mathbb{E}_{\substack{t<j:s_t,a_t\sim P_1,\pi\\ t\geq j:s_t,a_t\sim P_2,\pi}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right].$$

Then we have

$$
\begin{aligned}
W_0&=\mathbb{E}_{s,a\sim\rho_{\mathcal{M}_2,\pi}}\left[r(s_t,a_t)\right]=\eta_{\mathcal{M}_2}(\pi),\\
\text{and } W_\infty&=\mathbb{E}_{s,a\sim\rho_{\mathcal{M}_1,\pi}}\left[r(s_t,a_t)\right]=\eta_{\mathcal{M}_1}(\pi).
\end{aligned}
$$

Thus we can obtain

$$\eta_{\mathcal{M}_1}(\pi)-\eta_{\mathcal{M}_2}(\pi)=\sum_{j=0}^{\infty}(W_{j+1}-W_j). \tag{11}$$

Convert W_j and W_{j+1} as following:

$$
\begin{aligned}
W_j&=R_j+\mathbb{E}_{s_j,a_j\sim P_1,\pi}\left[\mathbb{E}_{s_{j+1}\sim P_2}\left[\gamma^{j+1}V^{\pi}_{\mathcal{M}_2}(s_{j+1})\right]\right]\\
W_{j+1}&=R_j+\mathbb{E}_{s_j,a_j\sim P_1,\pi}\left[\mathbb{E}_{s_{j+1}\sim P_1}\left[\gamma^{j+1}V^{\pi}_{\mathcal{M}_2}(s_j)\right]\right]
\end{aligned}
$$

Plug back to Eq.11 and we obtain

$$
\begin{aligned}
\eta_{\mathcal{M}_1}(\pi)-\eta_{\mathcal{M}_2}(\pi)&=\sum_{j=0}^{\infty}(W_{j+1}-W_j)\\
&=\sum_{j=0}^{\infty}\gamma^{j+1}\mathbb{E}_{s,a\sim\mathbb{P}^{\pi}_{\mathcal{M}_1,j}}\left[\mathbb{E}_{s'\sim P_1}\left[V^{\pi}_{\mathcal{M}_2}(s')\right]-\mathbb{E}_{s'\sim P_2}\left[V^{\pi}_{\mathcal{M}_2}(s')\right]\right]\\
&=\frac{\gamma}{(1-\gamma)}\mathbb{E}_{s,a\sim\rho^{\pi}_{\mathcal{M}_1}}\left[\mathbb{E}_{s'\sim P_1}\left[V^{\pi}_{\mathcal{M}_2}(s')\right]-\mathbb{E}_{s'\sim P_2}\left[V^{\pi}_{\mathcal{M}_2}(s')\right]\right]\\
&=\frac{\gamma}{(1-\gamma)}\mathbb{E}_{s,a\sim\rho^{\pi}_{\mathcal{M}_1}}\left[\mathcal{G}^{\pi}_{\mathcal{M}_1,\mathcal{M}_2}(s,a)\right].
\end{aligned}
$$

□

Lemma C.2. (Extension of Telescoping Lemma.) Let $\mathcal{M}_1 := (\mathcal{S}, \mathcal{A}, P_1, r, \gamma)$ and $\mathcal{M}_2 := (\mathcal{S}, \mathcal{A}, P_2, r, \gamma)$ be two MDPs with different dynamics P_1 and P_2. Given two policies $\pi_1,\ \pi_2$, let

$$\mathcal{G}^{\pi_1,\pi_2}_{\mathcal{M}_1,\mathcal{M}_2}(s,a) := \mathbb{E}_{s',a'\sim P_1,\pi_1}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right],$$

we have

$$\eta_{\mathcal{M}_1}(\pi_1) - \eta_{\mathcal{M}_2}(\pi_2) = \frac{1}{(1-\gamma)}\mathbb{E}_{s,a\sim\rho^{\pi_1}_{\mathcal{M}_1}}\left[\mathcal{G}^{\pi_1,\pi_2}_{\mathcal{M}_1,\mathcal{M}_2}(s,a)\right].$$

Proof. Define W_j as the expected return when executing π_1 on $\mathcal{M}_1$ for the first j steps, then switching to π_2 and $\mathcal{M}_2$ for the remainder. That is

$$W_j := \sum_{t=0}^{\infty}\gamma^t \mathbb{E}_{\substack{t<j: s_t,a_t\sim P_1,\pi_1\\ t\geq j: s_t,a_t\sim P_2,\pi_2}}\left[r(s_t,a_t)\right] = \mathbb{E}_{\substack{t<j: s_t,a_t\sim P_1,\pi_1\\ t\geq j: s_t,a_t\sim P_2,\pi_2}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right].$$

Then we have

$$\begin{aligned} W_0 &= \mathbb{E}_{s,a\sim\rho_{\mathcal{M}_2,\pi_2}}\left[r(s_t,a_t)\right] = \eta_{\mathcal{M}_2}(\pi_2),\\ \text{and } W_\infty &= \mathbb{E}_{s,a\sim\rho_{\mathcal{M}_1,\pi_1}}\left[r(s_t,a_t)\right] = \eta_{\mathcal{M}_2}(\pi_1). \end{aligned}$$

Thus we can obtain

$$\eta_{\mathcal{M}_1}(\pi_1) - \eta_{\mathcal{M}_2}(\pi_2) = \sum_{j=0}^{\infty}(W_{j+1} - W_j). \tag{12}$$

Convert W_j and W_{j+1} as following:

$$\begin{aligned} W_j &= R_j + \mathbb{E}_{s_j,a_j\sim P_1,\pi_1}\left[\mathbb{E}_{s_{j+1},a_{j+1}\sim P_2,\pi_2}\left[\gamma^{j+1}Q^{\pi_2}_{\mathcal{M}_2}(s_{j+1},a_{j+1})\right]\right]\\ W_{j+1} &= R_j + \mathbb{E}_{s_j,a_j\sim P_1,\pi_1}\left[\mathbb{E}_{s_{j+1},a_{j+1}\sim P_1,\pi_1}\left[\gamma^{j+1}Q^{\pi_2}_{\mathcal{M}_2}(s_{j+1},a_{j+1})\right]\right] \end{aligned}$$

Plug back to Eq.12 and we obtain

$$\begin{aligned} \eta_{\mathcal{M}_1}(\pi_1) - \eta_{\mathcal{M}_2}(\pi_2) &= \sum_{j=0}^{\infty}(W_{j+1} - W_j)\\ &= \sum_{j=0}^{\infty}\gamma^{j+1}\mathbb{E}_{s,a\sim\mathbb{P}^{\pi_1}_{\mathcal{M}_1,j}}\left[\mathbb{E}_{s',a'\sim P_1,\pi_1}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right]\right]\\ &= \frac{\gamma}{(1-\gamma)}\mathbb{E}_{s,a\sim\rho^{\pi_1}_{\mathcal{M}_1}}\left[\mathbb{E}_{s',a'\sim P_1,\pi_1}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right]\right]\\ &= \frac{\gamma}{(1-\gamma)}\mathbb{E}_{s,a\sim\rho^{\pi_1}_{\mathcal{M}_1}}\left[\mathcal{G}^{\pi_1,\pi_2}_{\mathcal{M}_1,\mathcal{M}_2}(s,a)\right]. \end{aligned}$$

□

Lemma C.3. (Bound of $\mathcal{G}^{\pi_1,\pi_2}_{\mathcal{M}_1,\mathcal{M}_2}(s,a)$.) Let

$$\mathcal{G}^{\pi_1,\pi_2}_{\mathcal{M}_1,\mathcal{M}_2}(s,a) := \mathbb{E}_{s',a'\sim P_1,\pi_1}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right],$$

we have

$$\begin{aligned} \mathcal{G}^{\pi_1,\pi_2}_{\mathcal{M}_1,\mathcal{M}_2}(s,a) \leq\ & \frac{2r_{\max}}{1-\gamma}\mathbb{E}_{s'\sim P_1}\left[D_{TV}(\pi_1(\cdot|s')\parallel\pi_2(\cdot|s'))\right]\\ &+ \left|\mathbb{E}_{s',a'\sim P_1,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right]\right|. \end{aligned}$$

Proof. We have

$$\begin{aligned}\mathcal{G}^{\pi_1,\pi_2}_{\mathcal{M}_1,\mathcal{M}_2}(s,a) :=& \mathbb{E}_{s',a'\sim P_1,\pi_1}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] \\ =& \underbrace{\mathbb{E}_{s',a'\sim P_1,\pi_1}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_1,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right]}_{(a)} \\ &+ \underbrace{\mathbb{E}_{s',a'\sim P_1,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right]}_{(b)}.\end{aligned}$$

For (a), we have

$$\begin{aligned}(a) &= \mathbb{E}_{s'\sim P_1}\left[\sum_{a'}\pi_1(a'|s')Q^{\pi_2}_{\mathcal{M}_2}(s',a') - \pi_2(a'|s')Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] \\ &\leq \mathbb{E}_{s'\sim P_1}\left[\sum_{a'}|\pi_1(a'|s') - \pi_2(a'|s')|\,\frac{r_{\max}}{1-\gamma}\right] \\ &= \frac{r_{\max}}{1-\gamma}\mathbb{E}_{s'\sim P_1}\left[\sum_{a'}|\pi_1(a'|s') - \pi_2(a'|s')|\right] \\ &= \frac{2r_{\max}}{1-\gamma}\mathbb{E}_{s'\sim P_1}\left[D_{TV}\left(\pi_1(\cdot|s')\parallel\pi_2(\cdot|s')\right)\right].\end{aligned}$$

For (b), we have

$$\begin{aligned}(b) &= \mathbb{E}_{s',a'\sim P_1,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] \\ &\leq \left|\mathbb{E}_{s',a'\sim P_1,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right] - \mathbb{E}_{s',a'\sim P_2,\pi_2}\left[Q^{\pi_2}_{\mathcal{M}_2}(s',a')\right]\right|.\end{aligned}$$

Adding these two bounds together yields the desired result. □

D Detailed Environment Setting

D.1 Grid World

In the grid world environment, the agent obtains the X-Y coordination as the state and executes one of the four actions (Up, Down, Left, Right) at each time step. A non-zero reward 1.0 is provided only if the agent reaches the goal. Each episode terminates when the agent reaches the goal or the episode length of 256 is reached. The source domain and the target domain of the grid world are shown in Figure 9. For each algorithm, the agent interacts with the source and target domains for $5e^5$ and $5e^4$ steps, respectively.

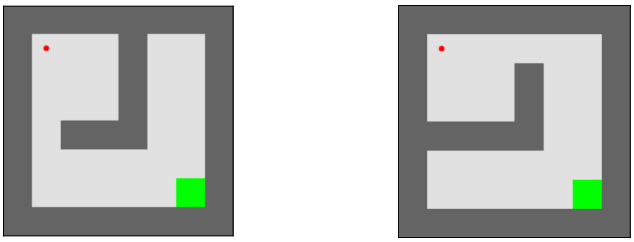

Figure 9: The source domain (Left) and the target domain (Right) of the grid world environments.

D.2 Mujoco Environments

To investigate the performance of the algorithm thoroughly, we design eight environments based on four Mujoco [66] benchmarks from Gym [5] including HalfCheetah-v2, Ant-v4, Walker2D-v2,

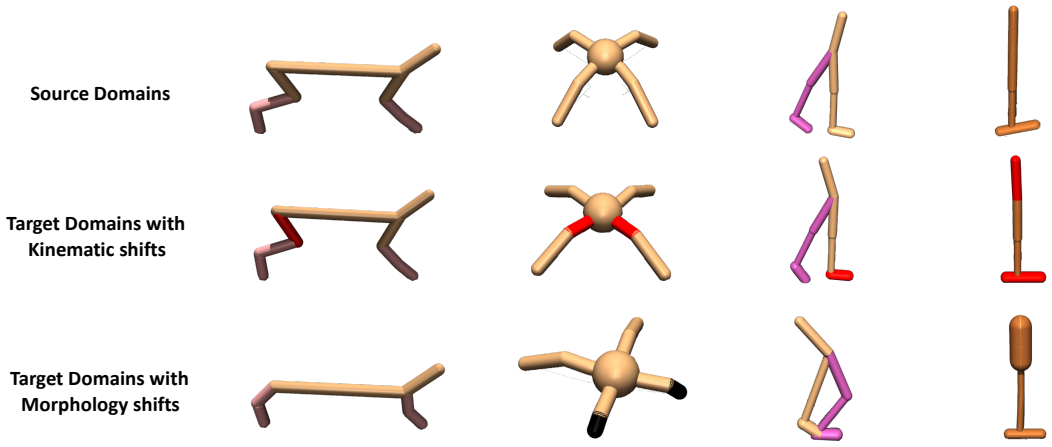

Figure 10: Illustration of all environments, including all source domains (*Top*), all target domains with kinematic shifts (*Middle*), and all target domains with morphology shifts (*Bottom*).

and Hopper-v2. For each benchmark, we propose two variants with kinematic shift or morphology shift. We run all experiments with the original environment as the source domain and the variation environment as the target domain. Detailed modifications of the environments are shown below, and the illustration of the environments is shown in Figure 10. For algorithms that access interactions with both domains, the agent interacts with the source and target domains for 10^6 and 10^5 steps, respectively.

Detailed modifications of the environments with kinematic shifts are shown below:

HalfCheetah - broken back thigh: We modify the rotation range of the joint on the thigh of the back leg from $[-0.52, 1.05]$ to $[-0.0052, 0.0105]$.

Ant - broken hips: We modify the rotation range of the joints on the hip of leg 1 and leg 2 from $[-30, 30]$ to $[-0.3, 0.3]$.

Walker - broken right foot: We modify the rotation range of the joint on the foot of the right leg from $[-45, 45]$ to $[-0.45, 0.45]$.

Hopper - broken joints: We modify the rotation range of the joint on the head from $[-150, 0]$ to $[-0.15, 0]$ and the joint on foot from $[-45, 45]$ to $[-18, 18]$.

Detailed modifications of the environments with morphology shifts are shown below:

HalfCheetah - no thighs: We modify the size of both thighs. Detailed modifications of the xml file are:

```
<geom fromto="0 0 0 -0.0001 0 -0.0001" name="bthigh" size="0.046" type
    ="capsule"/>
<body name="bshin" pos="-0.0001 0 -0.0001">
```

```
<geom fromto="0 0 0 0.0001 0 0.0001" name="fthigh" size="0.046" type="
    capsule"/>
<body name="fshin" pos="0.0001 0 0.0001">
```

Ant - short feet: We modify the size of feet on leg 1 and leg 2. Detailed modifications of the xml file are:

```
<geom fromto="0.0 0.0 0.0 0.1 0.1 0.0" name="left_ankle_geom" size="
    0.08" type="capsule"/>
```

```
<geom fromto="0.0 0.0 0.0 -0.1 0.1 0.0" name="right_ankle_geom" size="
    0.08" type="capsule"/>
```

Walker - no right thigh: We modify the size of thigh on the right leg. Detailed modifications of the xml file are:

```
<body name="thigh" pos="0 0 1.05">
    <joint axis="0 -1 0" name="thigh_joint" pos="0 0 1.05" range="-150
        0" type="hinge"/>
    <geom friction="0.9" fromto="0 0 1.05 0 0 1.045" name="thigh_geom"
        size="0.05" type="capsule"/>
    <body name="leg" pos="0 0 0.35">
      <joint axis="0 -1 0" name="leg_joint" pos="0 0 1.045" range="
          -150 0" type="hinge"/>
      <geom friction="0.9" fromto="0 0 1.045 0 0 0.3" name="leg_geom"
          size="0.04" type="capsule"/>
      <body name="foot" pos="0.2 0 0">
        <joint axis="0 -1 0" name="foot_joint" pos="0 0 0.3" range="
            -45 45" type="hinge"/>
        <geom friction="0.9" fromto="-0.0 0 0.3 0.2 0 0.3" name="
            foot_geom" size="0.06" type="capsule"/>
      </body>
    </body>
</body>
```

Hopper - big head: We modify the size of the head. Detailed modifications of the xml file are:

```
<geom friction="0.9" fromto="0 0 1.45 0 0 1.05" name="torso_geom" size
    ="0.125" type="capsule"/>
```

E Algorithms and Implementation Details

E.1 Implementation Details

The details of our algorithm and baseline methods are specified as follows:

SAC: We first specify the implementation of the shared backbone algorithm SAC utilized in all algorithms. The policy and the value function are two-layer MLP with 256 hidden units using ReLU activation. The learning rate is $3e^{-4}$. Discount γ is set as 0.99 in all environments. The temperature coefficient is fixed as 0.2. The batch size is 128. The smoothing coefficient of the target networks is 0.005. The training delay of the policy is set as 2. The replay buffer size is $1e^6$.

VGDF: We use a five-layer MLP with 200 units as the dynamics model using Swish activation following prior works [8, 26]. The ensemble size is 7. We set the data selection ratio $\xi\%$ as 25% in the experiments shown in Section 6.1. For each probabilistic dynamics model $T_{\phi_i}(s_{t+1}, r_t|s_t, a_t) = \mathcal{N}(\mu_{\phi_i}(s_t, a_t), \Sigma_{\phi_i}(s_t, a_t))$, $i = 1, \ldots, M$, we train the model by maximizing the objective:

$$J(\phi_i) := \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim D_{tar}}\big[\big[\mu_{\phi_i}(s_t, a_t) - (s_{t+1}, r_t)\big]^\top \Sigma_{\phi_i}^{-1}(s_t, a_t)\,[\mu_{\phi_i}(s_t, a_t) - (s_{t+1}, r_t)] + \log \det \Sigma_{\phi_i}(s_t, a_t)\big]. \tag{13}$$

The exploration policy is a two-layer MLP with 256 hidden units. We warm-start the algorithm by utilizing samples from both domains without selection for the first $1e5$ steps in the source domain.

DARC: We follow the default configurations of the public implementation (https://github.com/google-research/google-research/tree/master/darc). The domain classifiers $q_{\psi_{SAS}}(s_t, a_t, s_{t+1})$, $q_{\psi_{SA}}(s_t, a_t)$ are trained by maximizing the cross-entropy losses:

$$\begin{aligned} J(\psi_{SAS}) &:= \mathbb{E}_{(s_t,a_t,s_{t+1})\sim D_{tar}}\left[\log q_{\psi_{SAS}}(tar|s_t, a_t, s_{t+1})\right] \\ &\quad + \mathbb{E}_{(s_t,a_t,s_{t+1})\sim D_{src}}\left[\log(1 - q_{\psi_{SAS}}(tar|s_t, a_t, s_{t+1}))\right], \\ J(\psi_{SA}) &:= \mathbb{E}_{(s_t,a_t)\sim D_{tar}}\left[\log q_{\psi_{SA}}(tar|s_t, a_t)\right] + \mathbb{E}_{(s_t,a_t)\sim D_{src}}\left[\log(1 - q_{\psi_{SA}}(tar|s_t, a_t))\right]. \end{aligned}$$

Following the original implementation, we use the standard Gaussian noise for the domain classifier training. During training, a reward correction $\Delta r(s_t, a_t)$ is augmented to the original reward $r(s_t, a_t)$ of each source domain transition, *i.e.* $\tilde{r}(s_t, a_t) := r(s_t, a_t) + \Delta r(s_t, a_t)$. The reward correction is calculated by:

$$\Delta r(s_t, a_t) := \log \frac{q_{\psi_{SAS}}(tar|s, a, s')}{q_{\psi_{SAS}}(src|s, a, s')} \frac{q_{\psi_{SA}}(src|s, a)}{q_{\psi_{SA}}(tar|s, a)}.$$

Table 2: Hyperparameters. "-" denotes the hyperparameter is not used in the algorithm. "←" denotes the same choice as the algorithm in the first column.

Hyperparameters	VGDF	DARC	GARAT	IW Clip	Finetune
Hidden layers (Policy)	2	←	←	←	←
Hidden units per layer (Policy)	256	←	←	←	←
Hidden layers (Value)	2	←	←	←	←
Hidden units per layer (Value)	256	←	←	←	←
Hidden layers (Classifier)	-	2	-	2	-
Hidden units per layer (Classifier)	-	256	-	256	-
Hidden layers (Dynamics model)	5	-	-	-	-
Hidden units per layer (Dynamics model)	200	-	-	-	-
Ensemble size	7	-	-	-	-
Learning rate	$3e^{-4}$	←	←	←	←
Batch size	128	←	←	←	←
Fixed temperature coefficient	0.2	←	←	←	←
Target smoothing coefficient	0.005	←	←	←	←
Policy training delay	2	←	←	←	←
Buffer size	$1e^6$	←	←	←	←
Data selection ratio $\xi\%$	25%	-	-	-	-
Warm-start steps	$1e^5$	$1e^5$	-	$1e^5$	-
Importance weight clipping range	-	-	-	$[1e^{-4}, 1]$	-
Interactions with grounded src environment	-	-	$1e^5$	-	-

We warm-start the algorithm by training with samples from both domains for the first 10^5 steps following the original implementation.

GARAT: We use the author implementation with default configurations (`Supplemental` in https://proceedings.neurips.cc/paper/2020/hash/28f248e9279ac845995c4e9f8af35c2b-Abstract.html). We add the XML files of our customized environments to `rl_gat/envs/assets/` folder. We limit the extra interactions with the grounded source environments as 10^5 for fair comparisons with other algorithms.

Importance Weighting Clip (IW Clip): We use the domain classifiers same as DARC to calculate the importance weight $w(s, a, s')$. The importance weighting is calculated by:

$$w(s,a,s') := \frac{P_{tar}(s'|s,a)}{P_{src}(s'|s,a)} \approx \frac{q_{\psi_{SAS}}(tar|s,a,s')}{q_{\psi_{SAS}}(src|s,a,s')} \frac{q_{\psi_{SA}}(src|s,a)}{q_{\psi_{SA}}(tar|s,a)},$$

where $q_{\psi_{SAS}}$ and $q_{\psi_{SA}}$ are the domain classifiers proposed in [13]. We use the importance weighing to reweight the value training with source domain samples. Specifically,

$$\theta \leftarrow \arg\min_{\theta} \frac{1}{2}\mathbb{E}_{(s,a,r,s')\sim D_{src}}\left[w(s,a,s')(Q_\theta - \mathcal{T}Q_\theta)^2\right].$$

To stabilize training, we clip the importance weight between $[1e^{-4}, 1]$, same as the prior work [45].

Finetune: We first train a policy in the source domain with 10^6 steps. Then we transfer the policy to the target domain and further train the policy for 10^5 steps.

The detailed hyperparameters of all algorithms are listed in Table. 2, and we use the same hyperparameters across all environments.

E.2 Implementation Details of the Offline-Online Experiments

To evaluate the performance of our algorithm in the offline source online target setting, we use `medium` datasets from D4RL [16] for three environments (*i.e.*, HalfCheetah, Hopper, Walker). We use the same source domain offline dataset for each environment's two different target domains. For the algorithms performing online learning using offline data (*i.e.*, *Symmetric sampling*, *H2O*, *VGDF + BC*), we perform the online interactions with the target domain for 10^5 steps and use 10^6 source

domain transitions, the training is repeated for 10 times per step in the target domain. The details of the methods are specified as follows:

Offline only: We directly transfer the policy learned through CQL [31] with the source domain offline dataset. For the CQL implementation, we follow the suggested configurations in a public CQL implementation (https://github.com/tinkoff-ai/CORL). We perform training for 10^6 steps with the offline dataset and report the zero-shot performance of the learned policy in the target domain.

Symmetric sampling [2]: We perform the value function training by combining CQL optimization (with offline transitions) and SAC optimization (with online transitions). For each training step, we sample 50% of the data from the target domain replay buffer and the remaining 50% from the source domain offline dataset. The CQL and SAC loss is computed with the corresponding transitions.

H2O [45]: We follow the original implementation that learns the classifiers to estimate the dynamics discrepancy across domains and perform the clipped importance weighting on the CQL loss on the source domain data. Same as *Symmetric sampling*, we repeat the training for 10 times per step in the target domain.

VGDF + BC: We adapt VGDF to the Offline-Online setting by simply integrating the behavior cloning loss following (10). The training is repeated for 10 times per step with the target domain the same as the baseline methods. For the trade-off between the policy gradient and behavior cloning, we use the value-normalized regularization following the *TD3 + BC* [18] work and set the constant α as 5. Furthermore, we remove the exploration policy proposed in Section 5.1 since the online access to the source domain is no longer available in the offline-online setting.

F Additional Experiment Results

F.1 Quantifying Dynamics Shifts via FVP

In this section, we investigate whether the estimation of the value differences can quantify the difference across domains. Specifically, in different target domains of the same source domain, we demonstrate the estimation of FVP in two target domains. As the results show in Figure 11, the FVP differs in environments with different dynamics shifts (Kinematic or morphology). We observe that the FVP values in two target domains gradually approach each other in three out of four environments (HalfCheetah, Walker, Hopper), while the values in Ant remain relatively stationary. Furthermore, the FVP values in target domains with kinematic shifts are lower than those with morphology shifts across all four environments, which could result from the mismatched state space due to the limited joint ranges of robots in the target domain. Given the differences across different environments, we believe the FVP estimation could be used to quantify the domain differences.

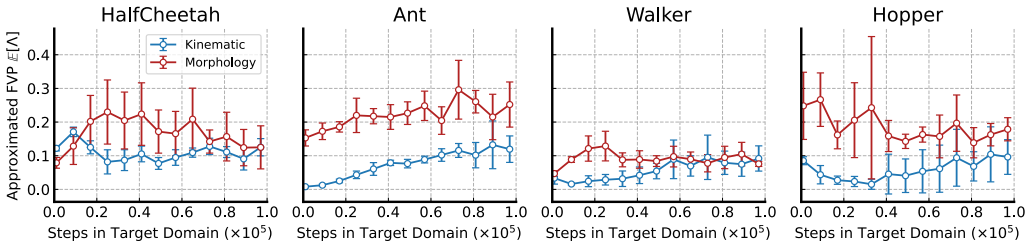

Figure 11: Quantification analysis of the approximated FVP in all environments with different dynamics shifts. The dots are averaged values, and the error bars indicate the standard error across five runs.

F.2 Sensitivity to Ensemble Size

We have introduced the dynamics model ensemble to capture the epistemic uncertainty induced by the limited samples from the target domain. However, training the ensemble of the dynamics model takes extra computation resources. Unlike prior works in model-based RL [26, 57] that utilize

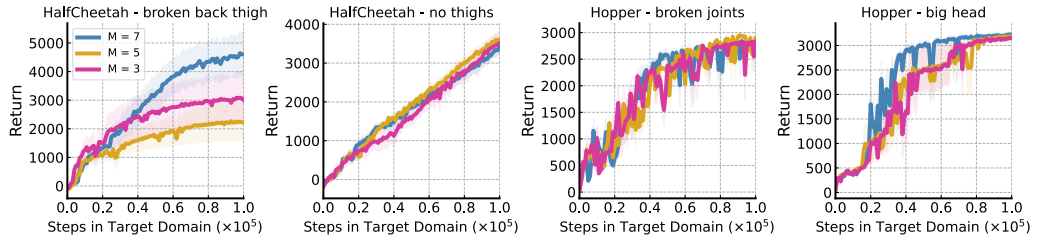

Figure 12: Performance of the variants with different ensemble size values M. The results validate that a smaller ensemble size is sufficient to achieve competitive asymptotic performance compared to the variant with a large ensemble size in most environments.

the generated samples for training, we measure the value difference with the help of the generated samples. Therefore, we aim to investigate whether a smaller ensemble size is sufficient to achieve competitive asymptotic performance. Here we set the ensemble size as different values ($M = 7$ in the original implementation) and run experiments in four environments. As the results show in Figure 12, variants with a small ensemble size (*e.g.*, $M = 3$ or $M = 5$) can achieve identical asymptotic performance compared to the variant with a large ensemble size (*e.g.*, $M = 7$) in three out of four environments.

F.3 What about Importance Weighting via FVP instead of Rejection Sampling?

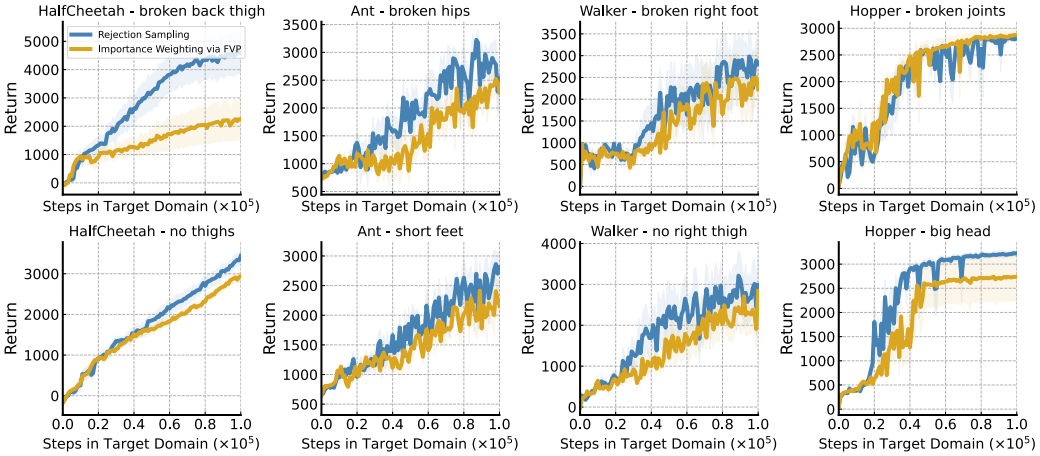

Figure 13: Performance of the variants with rejection sampling or importance weighting technique. The results demonstrate that the original algorithm using rejection sampling outperforms the variant using importance weighting via FVP in almost all environments.

In the case of data selection based on the estimated FVP (fictitious value proximity in Eq. (6), one may wonder about using importance weighting via the FVP rather than rejection sampling, which might be sample-inefficient due to the discarded partial data. Here we implement a variant of our algorithm that performs importance weighting with the estimated fictitious value proximity. Specifically, we train the value functions following:

$$\theta_{i=1,2} \leftarrow \arg\min_{\theta_i} \frac{1}{2B} \sum_{\{(s,a,r,s')\}^B_{tar}} \left[\left(Q_{\theta_i} - \mathcal{T} Q_{\theta_i} \right)^2 \right] + \frac{1}{2B} \sum_{\{(s,a,r,s')\}^B_{src}} \left[\frac{\Lambda(s,a,s')}{\sum_{\{s,a,s'\}^B} \Lambda(s,a,s')} \left(Q_{\theta_i} - \mathcal{T} Q_{\theta_i} \right)^2 \right].$$

We compare the variant with the original algorithm using rejection sampling in all eight environments and demonstrate the results in Figure 13. The original algorithm using rejection sampling outperforms

the variant with importance weighting in almost all environments. The accuracy of the value proximity depends on the generated state and the value function. Thus, the estimation of FVP could be biased due to the inaccurate dynamics models and value functions in the early training stage, in which case naively utilizing the source domain samples weighted by the FVP can harm the policy performance concerning the target domain. In contrast, rejection sampling that only utilizes a small portion of source domain samples alleviates the negative effect of the source domain samples.

F.4 What about Data Filtering via Value instead of FVP?

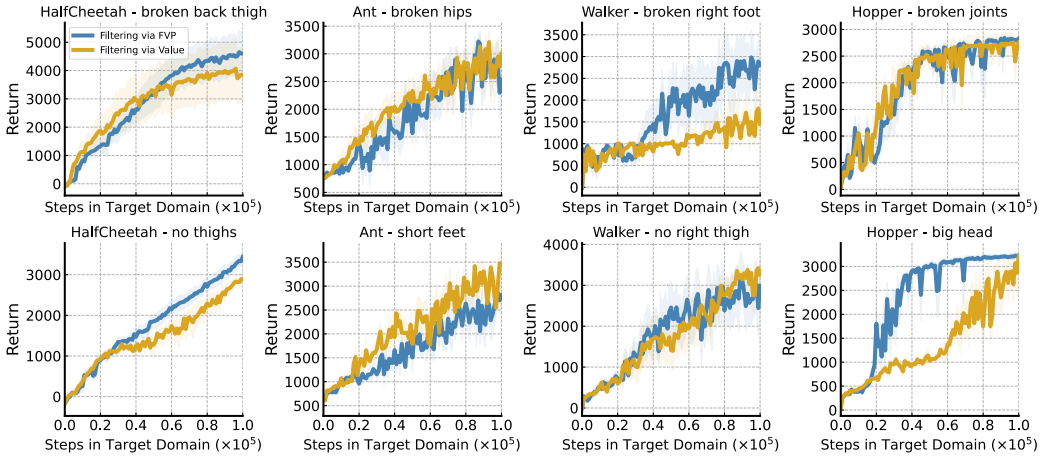

Figure 14: Performance of the variants that employ data filtering based on Value or FVP. The results demonstrate that the original algorithm outperforms the variant using data filtering via Value in four of eight environments.

Prior works have examined sharing data across tasks with different reward functions rather than dynamics [70]. To investigate whether selectively sharing data with a high Q value can address the online dynamics adaptation problem, we propose a variant of our algorithm that shares partial data with a relatively high Q value from the source domain. Specifically, we train the value functions following:

$$\theta_{i=1,2} \leftarrow \arg\min_{\theta_i} \frac{1}{2B} \sum_{\{(s,a,r,s')\}^B_{tar}} \left[\left(Q_{\theta_i} - \mathcal{T}Q_{\theta_i}\right)^2 \right] + \frac{1}{\lfloor 2B \cdot \xi\% \rfloor} \sum_{\{(s,a,r,s')\}^B_{src}} \left[\mathbb{1}\left(Q_{\theta_i}(s,a) > Q_{\xi\%}\right) \left(Q_{\theta_i} - \mathcal{T}Q_{\theta_i}\right)^2 \right],$$

where $Q_{\xi\%}$ is the top ξ-quantile Q value of a batch of source domain samples. We set $\xi\%$ as 25%, the same as our implementation. We compare the variant with the original algorithm in all eight environments and demonstrate the results in Figure 14. The results demonstrate that the original algorithm outperforms the variant using data filtering via value in four of eight environments. Due to the dynamics mismatch, a state-action pair from the source domain will lead to inconsistent states concerning two domains. Therefore, directly utilizing the transitions with high Q value without considering the consistency of the next state would provide a counterfactual value target for the state-action pair, which can result in an improper value estimation for learning.

F.5 Comparison with Dynamics-guided Data Filtering

To investigate the effect of value consistency, we perform the ablation study by comparing VGDF to a variant that shares partial data based on dynamics discrepancies, *i.e.*, Dynamics-guided Data Filtering (DGDF). Specifically, we estimate the dynamics discrepancy via the learned classifiers following the prior works [13, 45]. Same as VGDF, we share the source domain transitions whose estimated dynamics difference is smaller than the quantile value. We set the selection ratios as 25%,

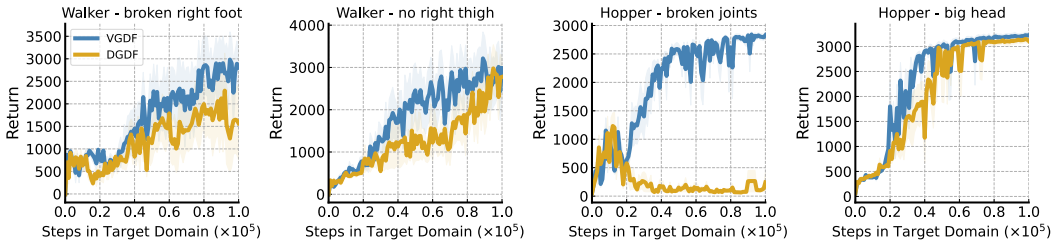

Figure 15: Comparison with the variant performing data filtering based on estimated dynamics discrepancies. The results demonstrate that the original algorithm outperforms the variant using data filtering via Value in four of eight environments, validating the effect of the value consistency.

the same as our implementation. The results demonstrate that the original algorithm outperforms the variant in three out of four environments, validating the superior effect of the value consistency compared to the dynamics discrepancy.