Implicit Distributional Reinforcement Learning: Appendix

A Proof of Lemma[1]

Denote
\[ \mathcal{H} = \mathbb{E}_{a \sim \pi_{\theta}(a|s)} \log \pi_{\theta}(a|s), \]
and
\[ \mathcal{H}_L = \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0)})} \log \frac{1}{L+1} \sum_{\ell=0}^L \pi_{\theta}(a|s, \xi^{(\ell)}), \]
and
\[ \pi_{\theta}(a|s, \xi^{(0):(L)}) = \frac{1}{L+1} \sum_{\ell=0}^L \pi_{\theta}(a|s, \xi^{(\ell)}). \]

Notice that \( \xi \)s are from the same distribution, so we have
\[
\mathcal{H}_L = \frac{1}{L+1} \sum_{i=0}^L \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0)})} \log \frac{1}{L+1} \sum_{\ell=0}^L \pi_{\theta}(a|s, \xi^{(\ell)})
= \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0):(L)})} \log \pi_{\theta}(a|s, \xi^{(0):(L)}),
\]

Use the identity that \( \mathbb{E}_{a \sim \pi_{\theta}(a|s)} = \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0):(L)})} \), we can rewrite \( \mathcal{H} \) as
\[ \mathcal{H} = \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0):(L)})} \log \pi_{\theta}(a|s). \]

Therefore, we have
\[
\mathcal{H}_L - \mathcal{H} = \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0):(L)})} \log \frac{\pi_{\theta}(a|s, \xi^{(0):(L)})}{\pi_{\theta}(a|s)} \geq 0.
\]

To compare between \( \mathcal{H}_L \) and \( \mathcal{H}_{L+1} \), rewrite \( \mathcal{H}_L \) as
\[ \mathcal{H}_L = \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0):(L)})} \log \pi_{\theta}(a|s, \xi^{(0):(L)}) \]
and \( \mathcal{H}_{L+1} \) as
\[ \mathcal{H}_{L+1} = \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0)):\xi^{(L+1)})} \log \pi_{\theta}(a|s, \xi^{(0):(L+1)}) \]
and the difference will be
\[
\mathcal{H}_L - \mathcal{H}_{L+1} = \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)} \sim p(\xi)} \mathbb{E}_{a \sim \pi_{\theta}(a|s, \xi^{(0):(L)})} \left[ \log \pi_{\theta}(a|s, \xi^{(0):(L)}) - \log \pi_{\theta}(a|s, \xi^{(0):(L+1)}) \right]
= \mathbb{E}_{\xi^{(0)}, \ldots, \xi^{(L)}, \xi^{(L+1)} \sim p(\xi)} KL(\pi_{\theta}(a|s, \xi^{(0):(L)}), \pi_{\theta}(a|s, \xi^{(0):(L+1)})) \geq 0.
\]

Finally, we arrive at the conclusion that for any \( \ell \), we have
\[ \mathcal{H} \leq \mathcal{H}_{\ell+1} \leq \mathcal{H}_\ell. \]
**Algorithm 2** Implicit Distributional Actor-Critic (IDAC)

**Require:** Learning rate $\lambda$, batch size $M$, quantile number $K$, action number $J$ and noise number $L$, target entropy $\mathcal{H}_t$.

Initial policy network parameter $\theta$, action-value function network parameter $\omega_1, \omega_2$, entropy parameter $\eta$.

Initial target network parameter $\tilde{\omega}_1 = \omega_1, \tilde{\omega}_2 = \omega_2$.

for the number of environment steps do

Sample $M$ number of transitions $\{s^i_t, a^i_t, r^i_t, s^i_{t+1}\}_{i=1}^M$ from the replay buffer

Sample $\varepsilon_i^{(k)}, e_i^{(k)}, \xi_i^{0,(0)}$ from $\mathcal{N}(0, 1)$ for $i = 1 \cdots M$ and $k = 1 \cdots K$ and $\ell = 0 \cdots L$.

Sample $a_{i+1}^j \sim \pi_\theta(\cdot | s^i_{t+1}, \xi_{i+1}^{0,(0)}) = \mathcal{N}(T^\beta_\theta(s^i_{t+1}, \xi_{i+1}^{0,(0)}), T^\alpha_\theta(s^i_{t+1}, \xi_{i+1}^{0,(0)}))$ for $i = 1 \cdots M$.

Apply Bellman update to create samples (of return distribution)

$$y_{1,i,k} = r^i_t + \gamma G_{\omega_1}(s^i_{t+1}, a^i_{t+1}, e^i_{t+1})$$
# Calculate target values

$$y_{2,i,k} = r^i_t + \gamma G_{\omega_2}(s^i_{t+1}, a^i_{t+1}, e^i_{t+1})$$
# Calculate target values

and let

$$(\tilde{y}_{1,1,i}, \ldots, \tilde{y}_{1,i,K}) = \text{StopGradient}(\text{sort}(y_{1,i,1}, \ldots, y_{1,i,K}))$$
# Obtain target quantile estimation

$$(\tilde{y}_{2,1,i}, \ldots, \tilde{y}_{2,i,K}) = \text{StopGradient}(\text{sort}(y_{2,i,1}, \ldots, y_{2,i,K}))$$
# Obtain target quantile estimation

$$\tilde{y}_{i,k} = \min(\tilde{y}_{1,1,i}, \ldots, \tilde{y}_{i,k})$$
for $i = 1 \cdots M; k = 1 \cdots K$

Generate samples $x_{1,i,k} = G_{\omega_1}(s^i_t, a^i_t, e^i_t)$ and $x_{2,i,k} = G_{\omega_2}(s^i_t, a^i_t, e^i_t)$, and let

$$(\tilde{x}_{1,1,i}, \ldots, \tilde{x}_{1,i,K}) = \text{sort}(x_{1,1,i}, \ldots, x_{1,i,K})$$

$$(\tilde{x}_{2,1,i}, \ldots, \tilde{x}_{2,i,K}) = \text{sort}(x_{2,1,i}, \ldots, x_{2,i,K})$$

Update action-value function parameter $\omega_1$ and $\omega_2$ by minimizing the quantile loss

$$J(\omega_1, \omega_2) = \frac{1}{M} \sum_{i=1}^M \frac{1}{K^2} \sum_{k=1}^K \sum_{k'=1}^K \rho_{k,k'}(\tilde{y}_{i,k} - \tilde{x}_{1,i,k'}) + \frac{1}{M} \sum_{i=1}^M \frac{1}{K^2} \sum_{k=1}^K \sum_{k'=1}^K \rho_{k,k'}(\tilde{y}_{i,k} - \tilde{x}_{2,i,k'})$$

Sample $\Xi_{t,h}^i, \varepsilon_t^{i,(j)}$ from $\mathcal{N}(0, 1)$, for $i = 1 \cdots M; j = 1 \cdots J$ and $h = 0 \cdots L + J$, and form $\xi_{t,0}^i, \varepsilon_t^i$ from $\Xi_{t,h}^i$ by concatenating $L$ of them to the rest of $Js$. Sample $a_{t,0}^{i,(j)} \sim \pi_\theta(\cdot | s^i_t$, $\xi_{t,0}^i)$ = $\mathcal{N}(T^\beta_\theta(s^i_t, \xi_{t,0}^i), T^\alpha_\theta(s^i_t, \xi_{t,0}^i))$ using

$$a_{t,0}^{i,(j)} = T_\theta(s^i_t, \xi_{t,0}^{i,(j)}) + \varepsilon_t^i \odot T^\alpha_\theta(s^i_t, \xi_{t,0}^{i,(j)}),$$

for $i = 1 \cdots M$.

Update the policy function parameter $\theta$ by minimizing

$$J(\theta) = -\frac{1}{M} \sum_{i=1}^M \left\{ \frac{1}{2J} \sum_{j=1}^J G_{\omega_2}(s^i_t, a_{t,0}^{i,(j)}, e_t^i) \right. - \left. \exp(\eta) \sum_{j=1}^J \frac{1}{L} \log \sum_{\ell=0}^L \pi_\theta(a_{t,0}^{i,(j)} | s^i_t, \xi_{t,0}^{i,(j)}) \right\} .$$

We also use stop gradient on $(T^\beta_\theta(s^i_t, \xi_{t,0}^{i,(j)}), T^\alpha_\theta(s^i_t, \xi_{t,0}^{i,(j)}))$ to reduce variance on gradient as mentioned in Eq. (16).

Update the log entropy parameter $\eta$ by minimizing

$$J(\eta) = \frac{1}{M} \sum_{i=1}^M \text{StopGradient}(-\log \frac{\sum_{\ell=0}^L \pi_\theta(a_{t,0}^{i,(0)} | s^i_t, \xi_{t,0}^{i,(0)})}{L + 1} - \mathcal{H}(\eta))$$

end for
### C Hyperparameters of IDAC

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>learning rate</td>
<td>3e-4</td>
</tr>
<tr>
<td>discount</td>
<td>0.99</td>
</tr>
<tr>
<td>replay buffer size</td>
<td>$10^6$</td>
</tr>
<tr>
<td>number of hidden layers (all networks)</td>
<td>2</td>
</tr>
<tr>
<td>number of hidden units per layer</td>
<td>256</td>
</tr>
<tr>
<td>number of samples per minibatch</td>
<td>256</td>
</tr>
<tr>
<td>entropy target</td>
<td>$-\dim(A)$ (e.g., $-6$ for HalfCheetah-v2)</td>
</tr>
<tr>
<td>nonlinearity</td>
<td>ReLU</td>
</tr>
<tr>
<td>target smoothing coefficient</td>
<td>0.005</td>
</tr>
<tr>
<td>target update interval</td>
<td>1</td>
</tr>
<tr>
<td>gradient steps</td>
<td>1</td>
</tr>
<tr>
<td>distribution of $\xi$</td>
<td>$\mathcal{N}(0, I_5)$</td>
</tr>
<tr>
<td>distribution of $\epsilon$</td>
<td>$\mathcal{N}(0, I_5)$</td>
</tr>
<tr>
<td>$J$</td>
<td>51</td>
</tr>
<tr>
<td>$K$</td>
<td>51</td>
</tr>
<tr>
<td>$L$</td>
<td>21</td>
</tr>
</tbody>
</table>

### D Additional ablation study

Additional ablation studies on Ant is shown in Fig. 4a for a thorough comparison. In Ant, the performance of IDAC is on par with that of IDAC-Gaussian, which outperforms the other variants.

Furthermore, we would like to learn the interaction between DGN and SIA by running ablation studies by holding each of them as a control factor; we conduct the corresponding experiments on Walker2d. From Fig. 4b, we can observe that by removing either SIA (resulting in IDAC-Gaussian) or DGN (resulting in IDAC-noDGN) from IDAC in general negatively impacts its performance, which echoes our motivation that we integrate DGN and SIA to allow them to help strengthen each other: (i) Modeling $G$ exploits distributional information to help better estimate its mean $Q$ (note C51, which outperforms DQN by exploiting distributional information, also conducts its argmax operation on $Q$); (ii) A more flexible policy may become more necessary given a better estimated $Q$.

![Figure 4: Additional plots for ablation study](image-url)
E Additional comparison with SDPG

In Fig. 5, we include a thorough comparison with SDPG (implemented based on the stable baselines codebase).

![Training curves on continuous control benchmarks.](image)

Figure 5: Training curves on continuous control benchmarks. The solid line is the average performance over 4 seeds with ± 1 std shaded, and with a smoothing window of length 100.

F Late stage policy visualization

We show in Fig. 6 the visualization of the late stage policy of one seed from Walker2d-v2 environment. We can see that SIA does provide a more flexible policy even in the late stage, where the correlations between action dimensions is clear on plot, and the marginal distributions are more flexible than a Gaussian distribution. Moreover, we randomly choose 1000 states and conduct normality tests as well as correlation tests on them. As a result, all of the tests indicate that the SIA policy captures the non-zero correlations between the selected dimensions, and the marginal distributions for each dimension across different states are significantly non-normal.

![Visualization of the SIA on Walker2d-v2.](image)

Figure 6: Visualization of the SIA on Walker2d-v2. The density contour of 1000 randomly sampled actions at a late training stage, where the x- and y-axis correspond to dimensions 1 and 4, respectively.