Algorithm 1: Distributional Soft Actor-Critic (DSAC)

Input: Initial policy parameters ϕ, quantile value network parameters θ_1, θ_2, target networks $\bar{\phi}, \bar{\theta}_1, \bar{\theta}_2$, polyak coefficient ι, number of quantiles N, Huber loss parameter κ, temperature α, discount γ

Data: Batch of transitions (s, a, r, s')

for *each batch* **do**

  Generate quantile fractions $\tau_i, i = 0, \ldots, N$ and $\hat{\tau}_j, j = 0, \ldots, N$;

  `// Update Quantile Value Networks` $Z(s, a; \theta_k), k = 1, 2$

  Get next actions for target: $a' \sim \pi(\cdot \mid s'; \bar{\phi})$;

  for $i = 0$ **to** $N - 1$ **do**

    for $j = 0$ **to** $N - 1$ **do**

      Compute target quantile:

$$y_i = \min_{k=1,2} Z_{\hat{\tau}_i}(s', a'; \bar{\theta}_k)$$

      Compute TD error:

$$\delta_{ij}^k = r + \gamma \left[y_i - \alpha \log \pi(a' \mid s'; \bar{\phi}) \right] - Z_{\hat{\tau}_j}(s, a; \theta_k), \quad k = 1, 2$$

    end

  end

  Compute quantile regression loss:

$$\mathcal{J}_Z(\theta_k) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (\tau_{i+1} - \tau_i) \varrho_{\hat{\tau}_j}^{\kappa}(\delta_{ij}^k), \quad k = 1, 2$$

  Update quantile network parameters:
  $\theta_k \leftarrow \theta_k - \lambda_Q \nabla \mathcal{J}_Z(\theta_k),\ k = 1, 2$;

  Update target quantile networks: $\bar{\theta}_k \leftarrow \iota \theta_k + (1 - \iota) \bar{\theta}_k,\ k = 1, 2$;

  `// Update Policy Network` $\pi(a \mid s; \phi)$

  Sample reparameterized actions: $\tilde{a} \sim \pi(\cdot \mid s; \phi)$;

  Compute expected Q-value:

$$Q(s, \tilde{a}) = \sum_{i=0}^{N-1} (\tau_{i+1} - \tau_i) \min_{k=1,2} Z_{\hat{\tau}_i}(s, \tilde{a}; \theta_k)$$

  Compute policy loss: $\mathcal{J}_\pi(\phi) = \alpha \log \pi(\tilde{a} \mid s; \phi) - Q(s, \tilde{a})$;

  Update policy parameters: $\phi \leftarrow \phi + \lambda_\pi \nabla \mathcal{J}_\pi(\phi)$;

  Update target policy: $\bar{\phi} \leftarrow \iota \phi + (1 - \iota) \bar{\phi}$;

end