Robot Learning

Maximum Entropy Reinforcement Learning

Prof. Songhwai Oh
ECE, SNU

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine, "**Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor**", in Proc. of the International Conference on Machine Learning (ICML), Aug, 2018.

SOFT ACTOR CRITIC

Soft Actor Critic (SAC)

- Actor-critic method
  - ▶ Continuous action space
- Maximum entropy reinforcement learning
  - ▶ Better exploration, stable faster learning
  - ▶ Stochastic policy, multiple modes
- Off-policy learning method
  - ▶ Sample efficiency

- Notation
  - MDP $(\mathcal{S}, \mathcal{A}, p, r)$
  - Transition probability, $\boldsymbol{p} : \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}} \rightarrow [\mathbf{0}, \infty)$
  - Reward, $\boldsymbol{r} : \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}} \rightarrow [\boldsymbol{r}_{\mathbf{min}}, \boldsymbol{r}_{\mathbf{max}}]$
  - $\boldsymbol{\rho_{\pi}}(\boldsymbol{s_t}) = Prob(\text{state at time t} = s_t)$
  - $\boldsymbol{\rho_{\pi}}(\boldsymbol{s_t}, \boldsymbol{a_t}) = Prob(state\ at\ time\ t = s_t, \text{action}\ at\ time\ t = a_t)$

Maximum Entropy Reinforcement Learning

Objective function

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi(\cdot|\mathbf{s}_t)) \right].$$

entropy

Soft policy evaluation

$$\mathcal{T}^\pi Q(\mathbf{s}_t, \mathbf{a}_t) \triangleq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \, \mathbb{E}_{\mathbf{s}_{t+1} \sim p} \left[V(\mathbf{s}_{t+1}) \right] \qquad V(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi} \left[Q(\mathbf{s}_t, \mathbf{a}_t) - \log \pi(\mathbf{a}_t|\mathbf{s}_t) \right]$$

Soft policy improvement

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} \mathrm{D}_{\mathrm{KL}} \left(\pi'(\cdot|\mathbf{s}_t) \,\middle\|\, \frac{\exp\left(Q^{\pi_{\text{old}}}(\mathbf{s}_t, \cdot)\right)}{Z^{\pi_{\text{old}}}(\mathbf{s}_t)} \right)$$

softmax policy

Soft Policy Iteration

- Soft Bellman Operator

$$\mathcal{T}^{\pi}Q(s,a) \coloneqq r(s,a) + \gamma \mathbb{E}_{s' \sim P}[V(s')]$$
$$V(s') \coloneqq \mathbb{E}_{a' \sim \pi}[Q(s',a') - \alpha \log \pi(a'|s')]$$

  - $-\log \pi(a'|s')$ comes from Shannon entropy
- Soft Policy Evaluation
  - Evaluation of quality of a given policy π
    - Random initial Q_0
    - Repeatedly apply soft Bellman operator

$$Q_{k+1} = \mathcal{T}^{\pi}Q_k$$

- Convergence

Theorem 1. For fixed π, consider the soft Bellman operator $\mathcal{T}^{\pi}$ and define $Q_{k+1} = \mathcal{T}^{\pi}Q_k$ for an arbitrary initial function F_0 over $S \times A$. Then, Q_k converges to soft action value Q^{π} as k goes infinity

Soft Policy Iteration

- Soft Policy Improvement
  - Update a policy distribution based on soft Q value which computed from soft policy evaluation

$$\forall s, \qquad \pi_{k+1}(\cdot\,|s) = \arg\min_{\pi} D_{kl}\left(\pi(\cdot\,|s)\middle|\frac{\exp Q^{\pi_k}(s,\cdot)}{Z(s)}\right)$$

  - Z indicates the partition function, which is often intractable
  - Minimize Kullback-Leibler divergence between a policy and a softmax of Q values
- Soft Policy Improvement Theorem

Theorem 2. Let π_{k+1} be updated by soft policy update rule, then, $Q^{\pi_{k+1}}(\mathrm{s},\mathrm{a}) \geq Q^{\pi_k}(\mathrm{s},\mathrm{a})$ for all state and action pairs.

SAC Algorithm

- Value network, V_ψ
- Target value network, $V_{\bar{\psi}}$
- Q-function network, Q_θ
  - $Q_{\theta_1}, Q_{\theta_2}$
- Policy network, π_ϕ

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors $\psi, \bar{\psi}, \theta, \phi$.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  end for
end for

Learning

Value network

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\frac{1}{2} \left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t \sim \pi_\phi} \left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) \right] \right)^2 \right]$$

$$\hat{\nabla}_\psi J_V(\psi) = \nabla_\psi V_\psi(\mathbf{s}_t) \left(V_\psi(\mathbf{s}_t) - Q_\theta(\mathbf{s}_t, \mathbf{a}_t) + \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) \right)$$

Q-function network

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}} \left[\frac{1}{2} \left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \hat{Q}(\mathbf{s}_t, \mathbf{a}_t) \right)^2 \right] \qquad \hat{Q}(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \, \mathbb{E}_{\mathbf{s}_{t+1} \sim p} \left[V_{\bar{\psi}}(\mathbf{s}_{t+1}) \right]$$

$$\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(\mathbf{a}_t, \mathbf{s}_t) \left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - r(\mathbf{s}_t, \mathbf{a}_t) - \gamma V_{\bar{\psi}}(\mathbf{s}_{t+1}) \right)$$

Policy network

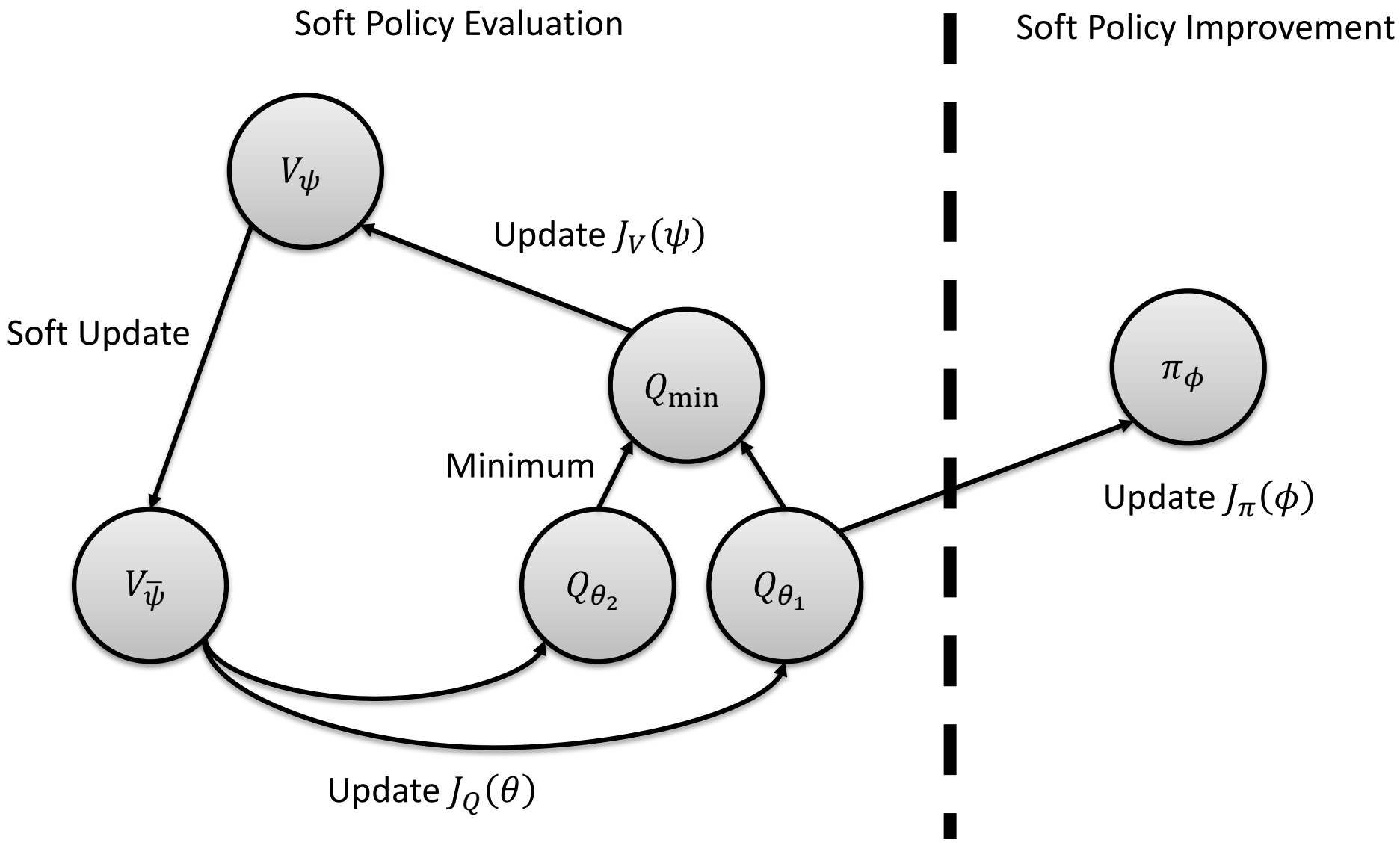
$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\mathrm{D}_{\mathrm{KL}} \left(\pi_\phi(\cdot | \mathbf{s}_t) \,\middle\|\, \frac{\exp\left(Q_\theta(\mathbf{s}_t, \cdot) \right)}{Z_\theta(\mathbf{s}_t)} \right) \right]$$

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}, \epsilon_t \sim \mathcal{N}} \left[\log \pi_\phi(f_\phi(\epsilon_t; \mathbf{s}_t) | \mathbf{s}_t) - Q_\theta(\mathbf{s}_t, f_\phi(\epsilon_t; \mathbf{s}_t)) \right]$$

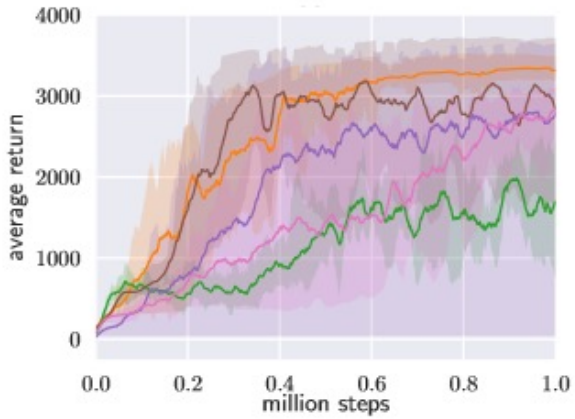
$\mathbf{a}_t = f_\phi(\epsilon_t; \mathbf{s}_t)$, ϵ_t: Gaussian noise (Reparameterization Trick)

$$\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) + \left(\nabla_{\mathbf{a}_t} \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) - \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t) \right) \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t)$$

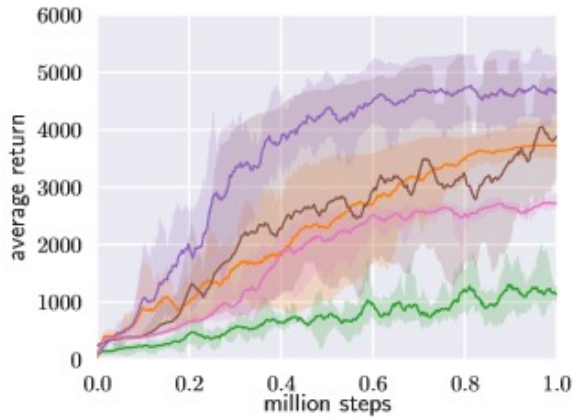
SAC


Soft Policy Evaluation
Soft Policy Improvement
V_ψ
Update $J_V(\psi)$
Soft Update
$Q_{\min}$
π_ϕ
Minimum
Update $J_\pi(\phi)$
$V_{\bar{\psi}}$
Q_{θ_2}
Q_{θ_1}
Update $J_Q(\theta)$

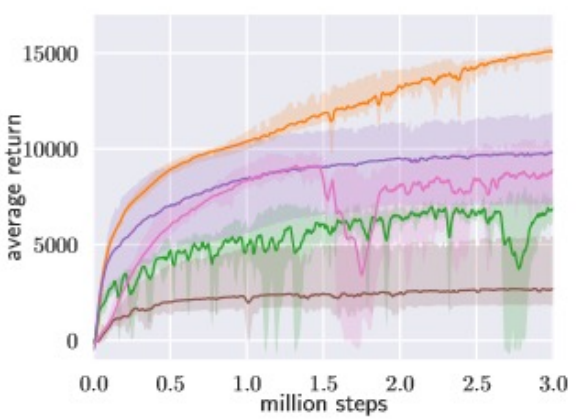
Experiments



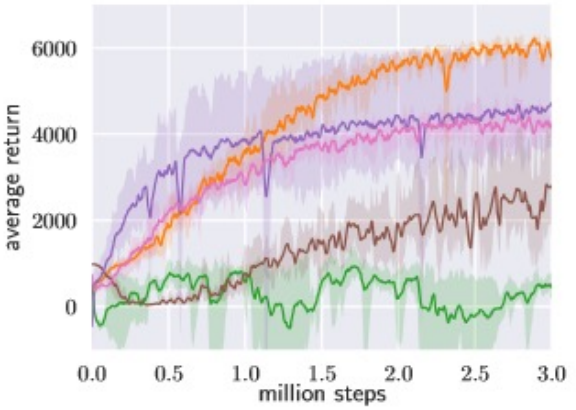
(a) Hopper-v1



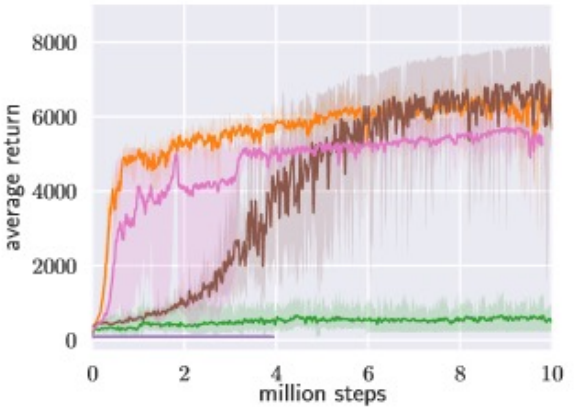
(b) Walker2d-v1



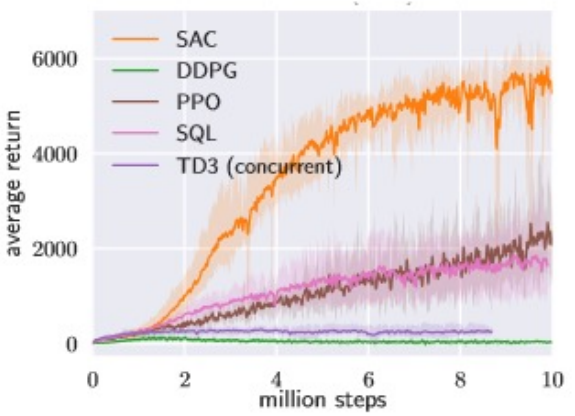
(c) HalfCheetah-v1



(d) Ant-v1



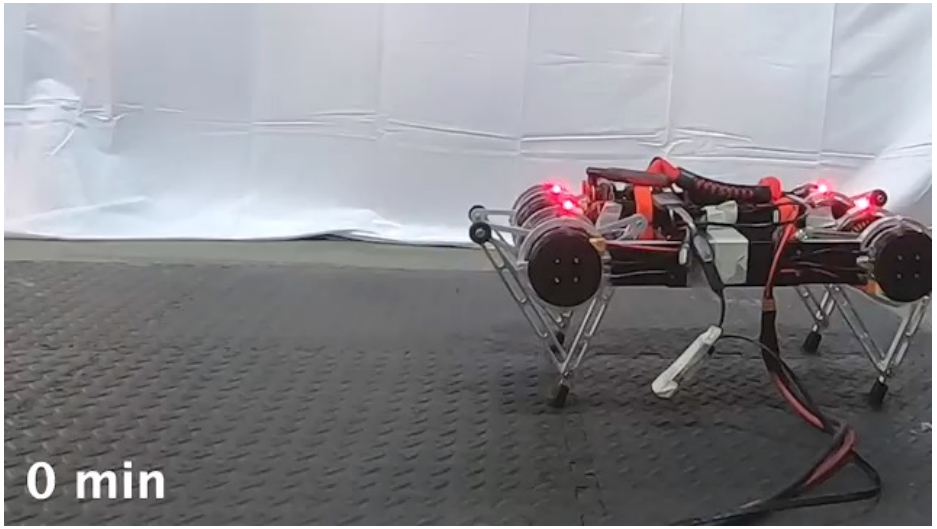
(e) Humanoid-v1



(f) Humanoid (rllab)

SAC in Action

Training



Test

Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, Mineui Hong, Jaein Kim, Yong-Lae Park, and Songhwai Oh, "**Generalized Tsallis Entropy Reinforcement Learning and Its Application to Soft Mobile Robots**," in Proc. of Robotics: Science and Systems (RSS), Jul. 2020

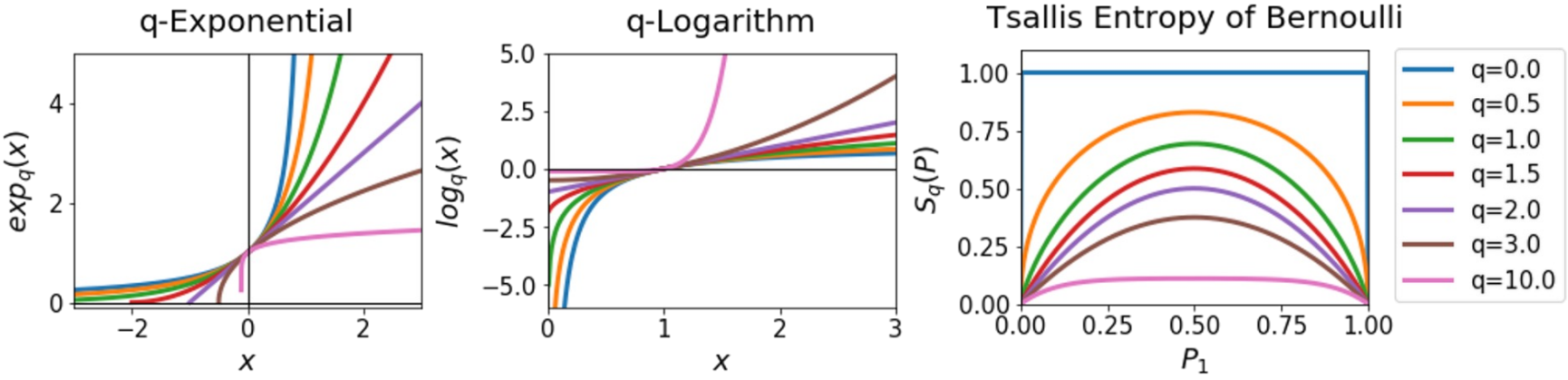
TSALLIS REINFORCEMENT LEARNING

Tsallis Reinforcement Learning

- Main Ideas
  - A unified framework which generalizes Shannon entropy maximization to Tsallis entropy maximization in reinforcement learning
  - The entropic index controls the exploration-exploitation trade-off
    - Different entropic index shows different exploration tendency
  - An off-policy maximum Tsallis entropy actor-critic algorithm
    - Tsallis actor critic with a proper entropic index outperforms existing actor-critic methods

Background: Tsallis Entropy

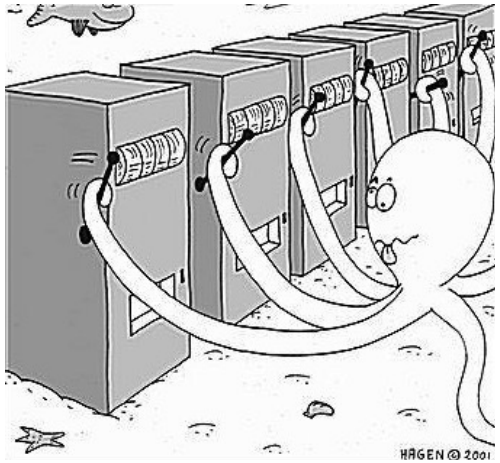
- q-Exponential : $\exp_q(x) \coloneqq \begin{cases} \exp(x), & \text{if } q = 1 \\ [1 + (q-1)x]_+^{\frac{1}{q-1}}, & \text{if } q \neq 1 \end{cases}$
- q-Logarithm : $\log_q(x) \coloneqq \begin{cases} \log(x), & \text{if } q = 1 \\ \frac{x^{q-1}-1}{q-1}, & \text{if } q \neq 1 \end{cases}$
- Tsallis entropy : $S_q(P) = \mathbb{E}_{X\sim P}\left[-\log_q P(X)\right]$



Multi Armed Bandit with Tsallis Entropy

- Multi Armed Bandit (A, R)
  - The agent selects an action: $a_t \in A$
  - The environment generates a reward: $R_t \sim P(\cdot\,|a_t)$
  - Reward function: $r(a_t) = \mathbb{E}[R_t|a_t]$
  - The goal is to maximize expected rewards: $\max_{\pi} \mathbb{E}_{a\sim\pi}[r(a)]$

- Tsallis Entropy
  - $\max_{\pi} \mathbb{E}_{a\sim\pi}[r(a)] + \alpha S_q(\pi)$
  - Apply Tsallis entropy to the bandit problem
  - A simpler problem than MDPs
  - Many properties of the multi-armed bandit are analogous to that of an MDP



Multi Armed Bandit with Tsallis Entropy

- The optimal policy

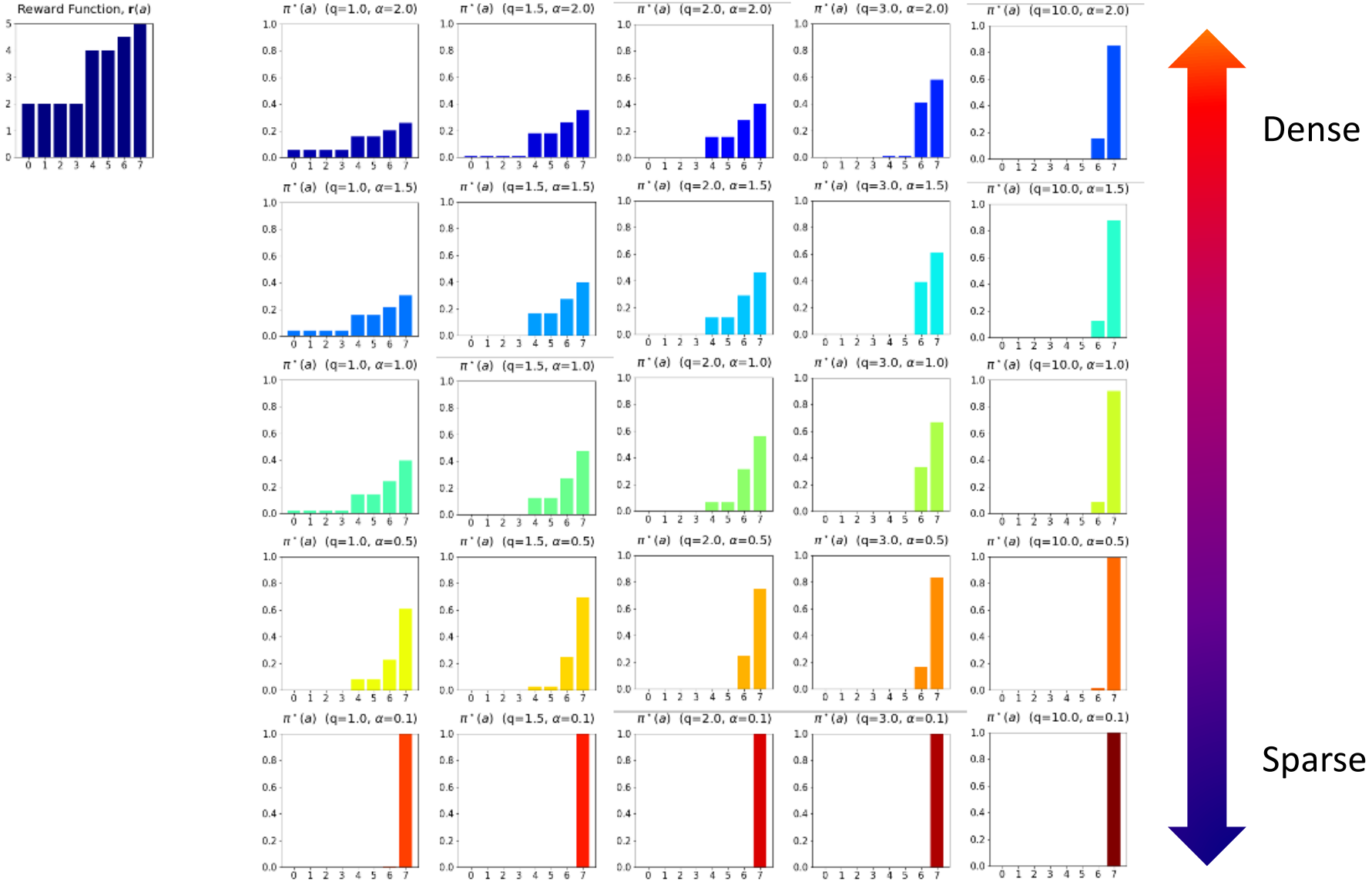
$$\pi_q^\star(a) = \exp_q\left(\frac{r(a)}{q\alpha} - \psi_q\left(\frac{r}{q\alpha}\right)\right)$$

  - If $q = 1$, $\pi_q^\star$ becomes a softmax distribution
  - If $q = 2$, $\pi_q^\star$ becomes a sparsemax distribution
  - If $q \to \infty$, $\pi_q^\star$ becomes a greedy policy !

- q-Potential function
  - ψ_q is called q-potential function, which satisfies the normalization constraint

$$\sum_a \pi_q^\star(a) = \sum_a \exp_q\left(\frac{r(a)}{q\alpha} - \psi_q\left(\frac{r}{q\alpha}\right)\right) = 1$$

Multi Armed Bandit with Tsallis Entropy



Tsallis Markov Decision Processes

$$\max_{\pi} \ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] + \alpha S_q(\pi)$$

$$\text{subject to} \quad \forall\, s, a \qquad \int \pi(a|s)da = 1, \qquad \pi(a|s) \geq 0$$

- Tsallis entropy regularization :

$$S_q(\pi) = \mathbb{E}\left[-\sum_{t=0}^{\infty} \gamma^t \log_q(\pi(a_t|s_t))\right]$$

- Special Cases:
  - If $q = 1$: Shannon Gibbs entropy ▶ Soft MDPs
  - If $q = 2$: Sparse Tsallis entropy ▶ Sparse MDPs
  - If $q \to \infty : S_q \to 0$ ▶ Original MDPs without regularization
- Different entropic indices induce different optimal policies

Tsallis Bellman Equation

Theorem 1. For q > 0, an optimal policy $\pi_q^\star$ and optimal value $V_q^\star$ sufficiently and necessarily satisfy the following **Tsallis-Bellman optimality (TBO)** equations:

$$Q_q^\star(s,a) = \mathbb{E}_{s' \sim P}\left[r(s,a,s') + \gamma V_q^\star(s') \mid s,a\right]$$

$$V_q^\star(s) = \operatorname*{qmax}_a \left(Q_q^\star(s,a)\right)$$

$$\pi_q^\star(a|s) = \exp_q\left(\frac{Q_q^\star(s,a)}{q} - \frac{\psi_q\left(Q_q^\star(s,\cdot)\right)}{q}\right)$$

Where ψ_q is a q-potential function

- q-maximum operator
  - For a function $f(x)$, q-maximum is defined as follows:

$$\operatorname*{qmax}_x f(x) \coloneqq \max_P \mathbb{E}_{X \sim P}[f(X)] + S_q(P)$$

- The optimal condition of Tsallis MDPs
- This theorem holds for every positive entropic indices

Tsallis Dynamic Programming

- Similar to original MDPs, we can derive two dynamic programming methods
  - **Tsallis policy iteration**
    - Compute both policy and state action value under Tsallis MDP settings
    - Update a policy based on its state action value
    - Policy converges to an optimal solution
  - **Tsallis value iteration**
    - Compute optimal state action values under Tsallis MDP settings
    - Tsallis Bellman optimality is used

Tsallis Dynamic Programming

- Tsallis Policy Iteration
  - Generalization of soft policy iteration
- Tsallis Policy Evaluation
  - Q_q^π : expected return of a given policy including Tsallis entropy bonus
  - Tsallis Bellmen Expectation (TBE) Operator

$$\mathcal{T}_q^\pi F(s,a) \coloneqq r(s,a) + \gamma \mathbb{E}_{s' \sim P}[V_F(s')]$$

$$V_F(s') \coloneqq \mathbb{E}_{a' \sim \pi}\left[F(s',a') - \alpha \log_q \pi(a'|s')\right]$$

  - Q_q^π of a fixed policy π is obtained by repeatedly applying TBE operator

Theorem 2. For fixed π and $q > 0$, consider the TBE operator $\mathcal{T}_q^\pi$ and define $F_{k+1} = \mathcal{T}_q^\pi F_k$ for an arbitrary initial function F_0 over $S \times A$. Then, F_k converges to Q_q^π as k goes infinity

Tsallis Dynamic Programming

- Tsallis Policy Iteration (cont.)
  - Generalization of soft policy iteration
- Tsallis Policy Improvement

$$\forall s, \qquad \pi_{k+1}(\cdot\,|s) = \arg\max_{\pi} \mathbb{E}_{a\sim\pi}\left[Q_q^{\pi_k}(s,a)\right] + S_q(\pi)$$

Theorem 3. For $q > 0$, let π_{k+1} be updated by Tsallis policy update rule, then, $Q_q^{\pi_{k+1}}(\mathrm{s},\mathrm{a}) \geq Q_q^{\pi_k}(\mathrm{s},\mathrm{a})$ for all state and action pairs.

  - The policy obtained by Tsallis policy improvement has performance no worse than the previous policy
  - It converges to optimal policy for all entropic indices

Theorem 4. For $q > 0$, let π_k be updated by Tsallis policy iteration. Then, π_k converges to the optimal policy.

Tsallis Dynamic Programming

- Tsallis Value Iteration
  - Tsallis Bellmen Optimality (TB0) Operator

$$\mathcal{T}_q^\star F(s,a) \coloneqq r(s,a) + \gamma \mathbb{E}_{s' \sim P}[V_F(s')]$$

$$V_F(s') \coloneqq \operatorname*{qmax}_a \big(F(s,a)\big)$$

  - Since TBO operator is a contraction mapping, TBO operator has a unique fixed point and it is an optimal state action value
  - $Q_q^\star$ is obtained by repeatedly applying TBO operator for all entropic indices

Theorem 5. For $q > 0$, consider the TBO operator $\mathcal{T}_q^\star$ and define $F_{k+1} = \mathcal{T}_q^\star F_k$ for an arbitrary initial function F_0 over $S \times A$. Then, F_k converges to optimal state action value, $Q_q^\star$ as k goes infinity

Tsallis Dynamic Programming

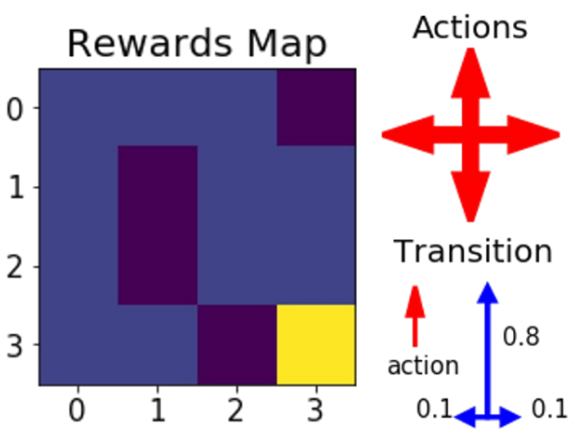
- Performance Error Bounds

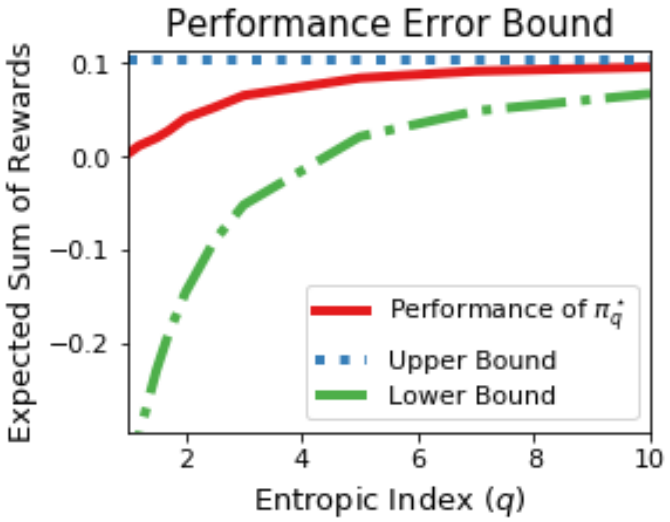
Theorem 6. Following inequalities hold:

$$\mathbb{E}_{\pi^\star}[r(s,a)] - \frac{\alpha}{1-\gamma}\log_q \frac{1}{|\mathcal{A}|} \leq \mathbb{E}_{\pi_q^\star}[r(s,a)] \leq \mathbb{E}_{\pi^\star}[r(s,a)]$$

where $\pi^\star$ and $\pi_q^\star$ are the optimal policy obtained by the original MDP and Tsallis MDP, respectively, and $|\mathcal{A}|$ is the number of action.

- Performance gap converges to zero as q goes to infinity





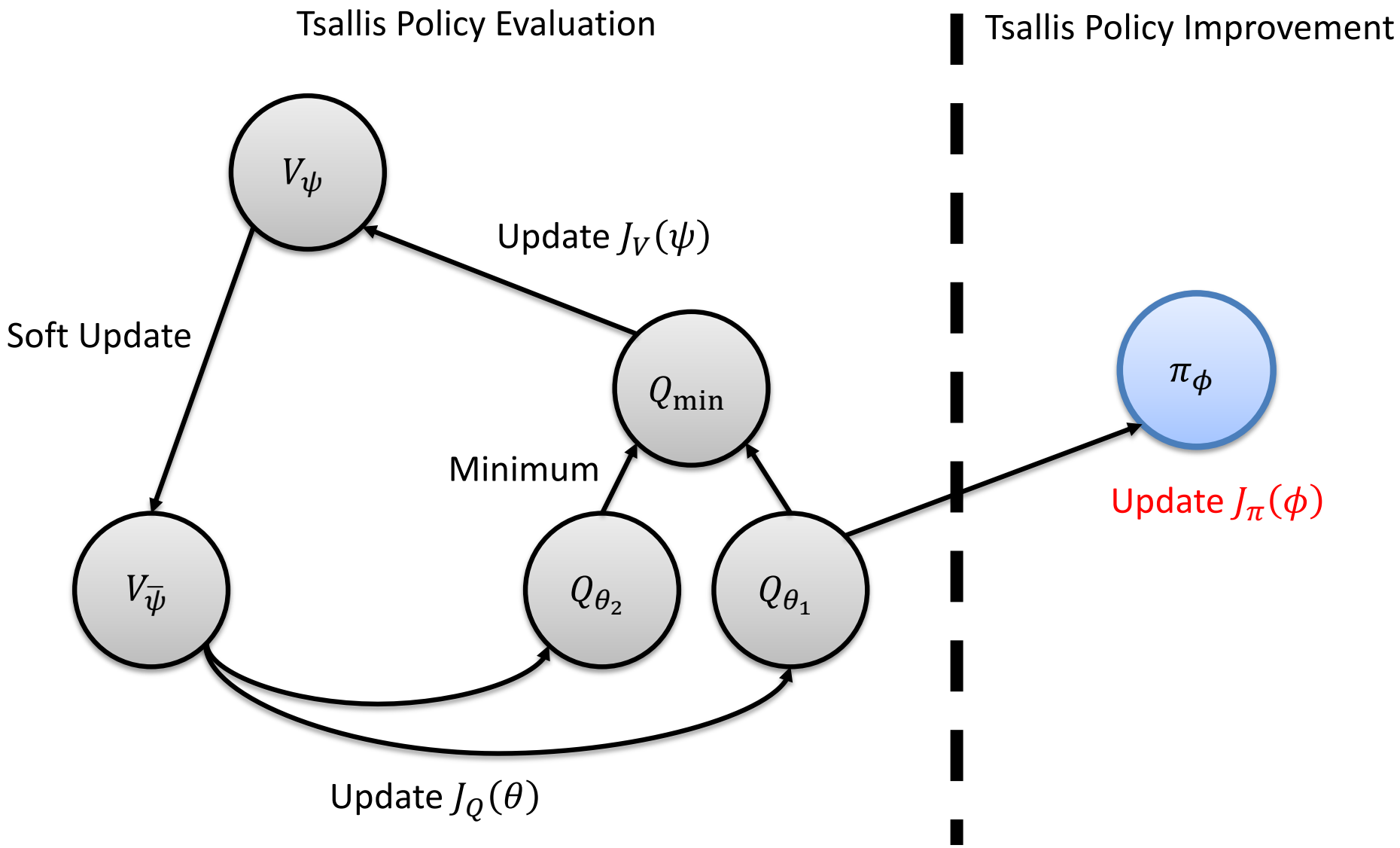
Tsallis Actor Critic (TAC)

- Extension of Tsallis policy iteration
  - Similar to soft actor-critic, five networks are trained
    - π_ϕ : Policy network (Gaussian policy) outputs $\mu_\phi(s)$ and $\sigma_\phi(s)$
    - $Q_{\theta_1}, Q_{\theta_2}$: state action values, same architecture with different parameters θ_1 and θ_2
    - $V_\psi, V_{\bar{\psi}}$: state values and target state values
  - $Q_{\theta_1}, Q_{\theta_2}$, and V_ψ are updated using the Tsallis Bellman expectation operator
  - For updating π_ϕ

$$J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D},\epsilon\sim\mathcal{N}}\left[\alpha \log_q \mathcal{N}\left(a_\epsilon; \mu_\phi(s), \sigma_\phi(s)\right) - Q_{\theta_1}(a_\epsilon, s)\right]$$

    - Reparameterization trick: $a_\epsilon = \mu_\phi(s) + \sigma_\phi(s)\epsilon$
    - Logarithm is changed to q-Logarithm

TAC


Tsallis Policy Evaluation
Tsallis Policy Improvement
V_ψ
Update $J_V(\psi)$
Soft Update
$Q_{\min}$
π_ϕ
Minimum
Update $J_\pi(\phi)$
$V_{\bar{\psi}}$
Q_{θ_2}
Q_{θ_1}
Update $J_Q(\theta)$

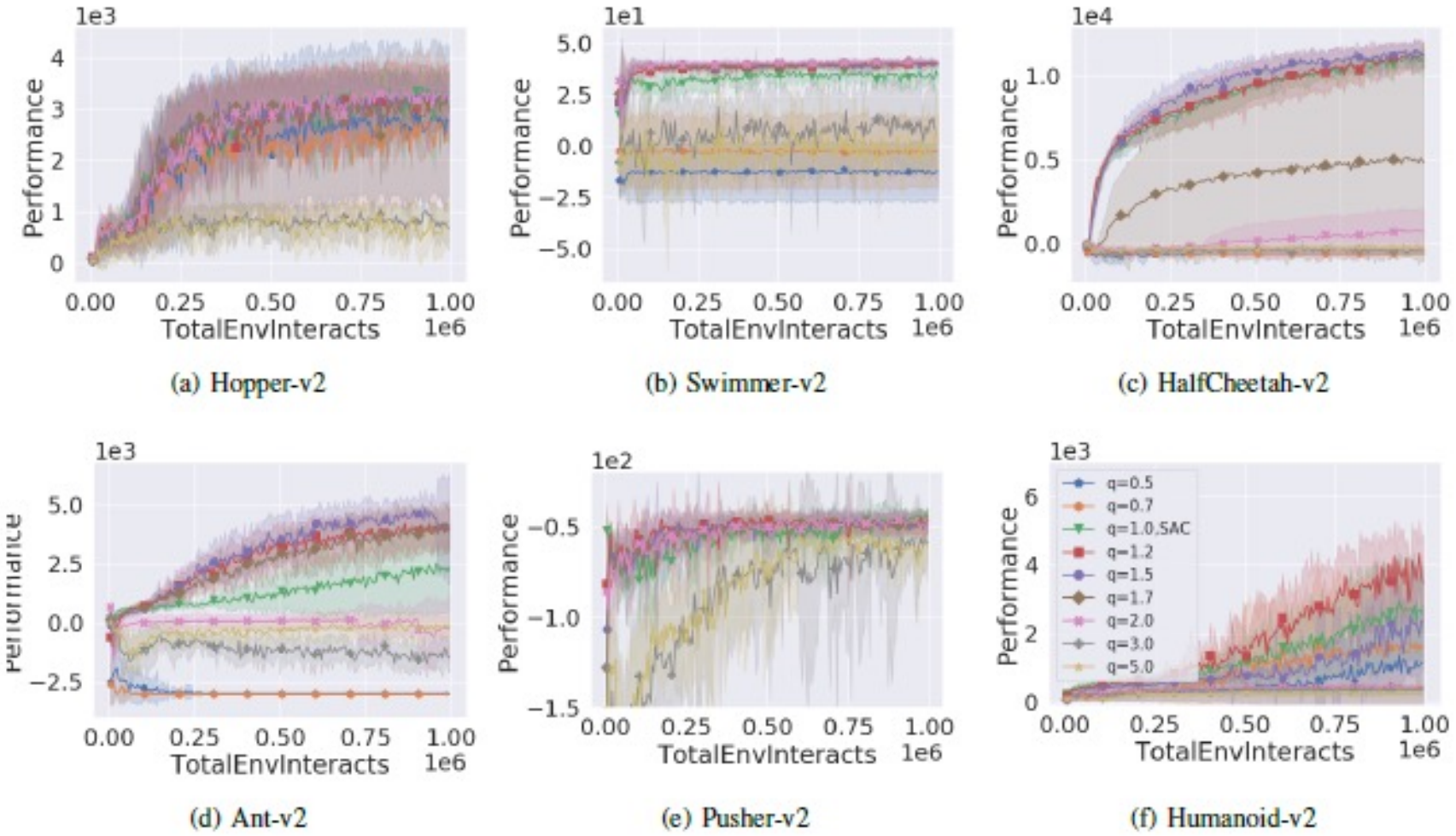
TAC

Algorithm 1 Tsallis Actor Critic (TAC)

Input: Total time steps $t_{\max}$, Max episode length $l_{\max}$, Memory size N, Entropy coefficient α, Entropic index q, Moving average ratio τ, Environment *env*
Initialize: $\psi, \psi^{-}, \theta_1, \theta_2, \phi, \mathcal{D}$: Queue with size N, $t = 0$, $t_e = 0$
while $t \leq t_{\max}$ **do**
  $a_t \sim \pi_\phi$ and $\mathbf{r}_{t+1}, s_{t+1}, d_{t+1} \sim env$ where d_{t+1} is a terminal signal.
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, \mathbf{r}_{t+1}, s_{t+1}, d_{t+1})\}$
  $t_e = t_e + 1, t = t + 1$
  if $d_{t+1} =$ True or $t_e = l_{\max}$ **then**
    for $i = 1$ to t_e **do**
      Randomly sample a mini-batch $\mathcal{B}$ from $\mathcal{D}$
      Minimize J_ψ, J_{θ_1}, J_{θ_2}, and J_ϕ using a stochastic gradient descent
      Update J_{ψ^-} with τ
    end for
    Reset *env*, $t_e = 0$
  end if
end while

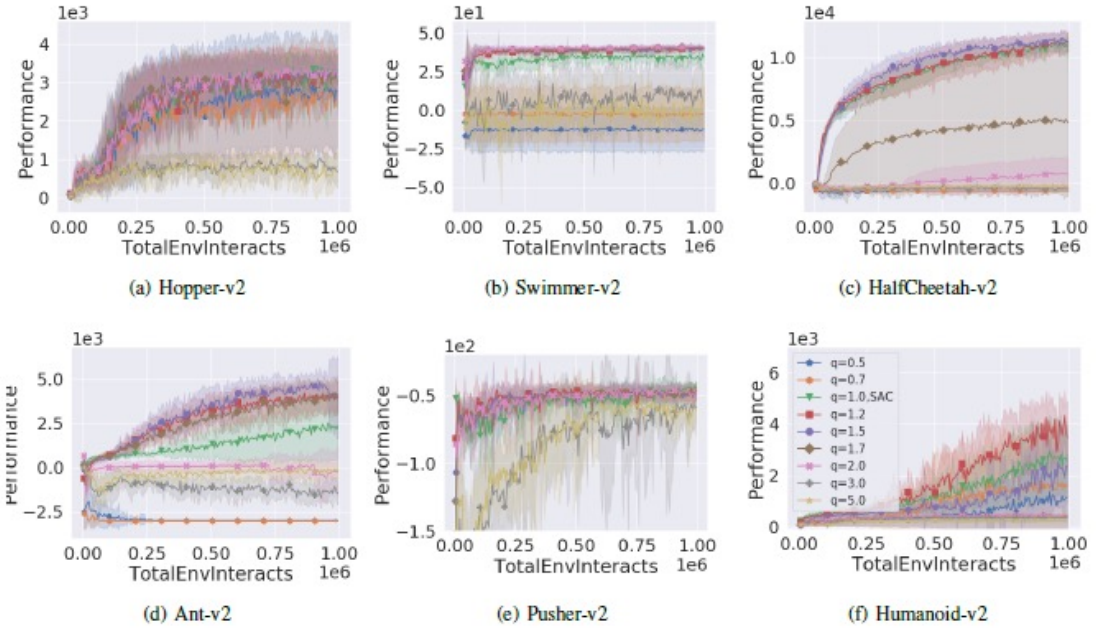
Experimental Results: TAC

- Different *q* values



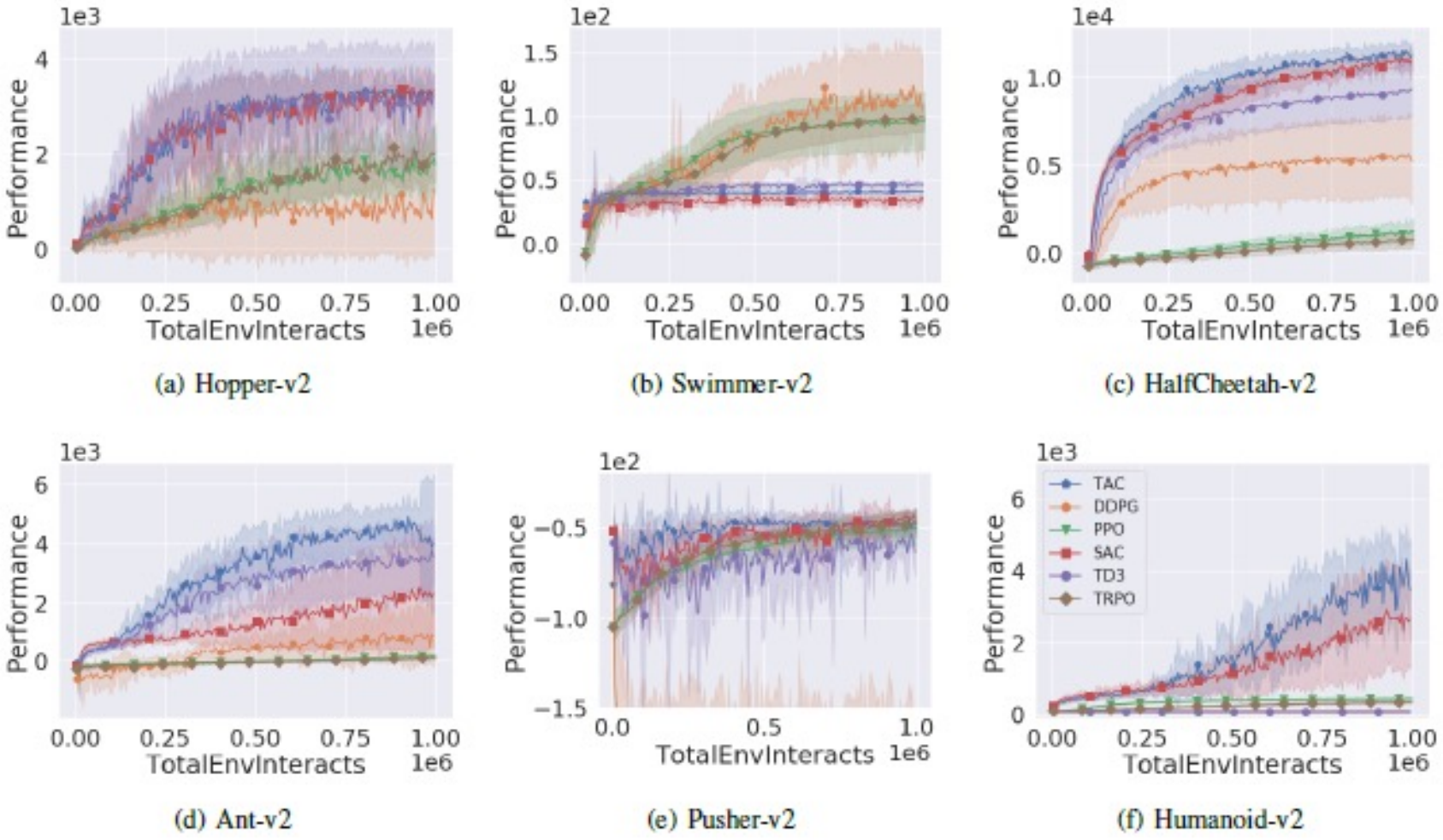
Experimental Results: TAC

- Different *q* values
  - TAC performs better when 1 < q < 2 than when 0 < q < 1 and q > 2, in terms of stable convergence and final total average returns
  - Using 0 < q < 1 generally shows poor performance since it hinders exploitation more strongly than the SG entropy.
  - For q > 2, the action with smaller probability is less encouraged to be explored



Experimental Results: TAC

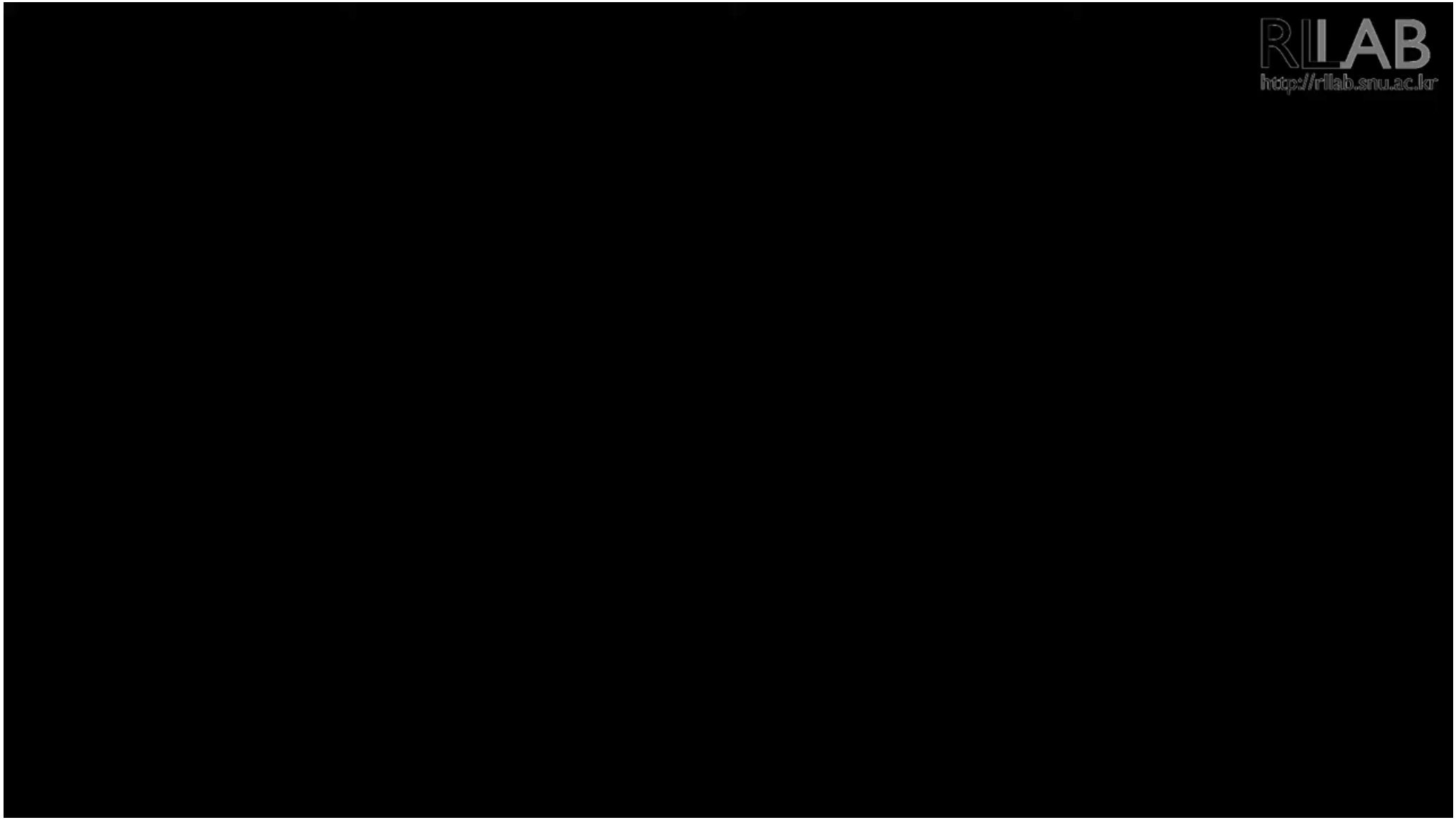
- Compared to existing methods



Summary: Tsallis Actor Critic

- Tsallis MDPs are a unified framework for maximum entropy RL problems, which allows a diverse range of different entropies
- Tsallis MDP (model is known) : Tsallis dynamic programming: convergence and optimality are guaranteed
- Model free RL (model is unknown) : TAC can handle a continuous state action space for model-free RL problems.
- There exists a suitable entropic index for a different RL problem and TAC with a specific entropic index outperforms all compared actor-critic methods

Application: Soft Robot Control



Jae In Kim, Mineui Hong, Kyungjae Lee, DongWook Kim, Yong-Lae Park, and Songhwai Oh, "**Learning to Walk a Tripod Mobile Robot Using Nonlinear Soft Vibration Actuators with Entropy Adaptive Reinforcement Learning**," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2317-2324, Apr. 2020.