Reward Shaping via Meta-Learning

Haosheng Zou [*1] Tongzheng Ren [*1] Dong Yan [1] Hang Su [1] Jun Zhu [1]

ArXiv 2019

Outline

- Preliminaries
  - Reward Shaping
  - Potential-based Reward Shaping
- Methods
  - Intuition(Meta Learning)
  - Method
- Experiments Results

Reward Shaping

■ The idea

The idea of shaping is to provide the learner with supplemental rewards that encourage progress towards highly rewarding states in the environment.

$$R_F(s, a, s') = R(s, a, s') + F(s, a, s')$$

$$M = \langle S, A, T, \gamma, R \rangle$$

$$\lim_{T \to \infty} \sum_{t=0}^{T-1} \gamma^t R_{t+1}, 0 < \gamma < 1$$

$$M' = \langle S, A, T, \gamma, R' \rangle$$

$$R' = R + F$$

$$F : S \times A \times S \to \mathbb{R}$$

Potential-Based Shaping

Ng, Russell and Harada. "Policy Invariance Under Reward Transformations: Theory And Application To Reward Shaping." ICML, 1999.

$$F(s, s') = \gamma\Phi(s') - \Phi(s)$$

- **Proof of sufficiency**

$$Q^*_M(s,a) = \mathrm{E}_{s' \sim P_{sa}(\cdot)}\left[R(s,a,s') + \gamma \max_{a' \in A} Q^*_M(s',a')\right]$$

$$Q^*_M(s,a) - \Phi(s) = \mathbb{E}_{s' \sim P_{sa}}\left[R(s,a,s') + \gamma \max_{a' \in A} Q^*_M(s',a')\right] - \Phi(s)$$

$$= \mathbb{E}_{s' \sim P_{sa}}\left[R(s,a,s') + \gamma\Phi(s') + \gamma \max_{a' \in A}(Q^*_M(s',a') - \Phi(s'))\right] - \Phi(s)$$

$$= \mathbb{E}_{s' \sim P_{sa}}\left[R(s,a,s') + \gamma\Phi(s') - \Phi(s) + \gamma \max_{a' \in A}\left(Q^*_M(s',a') - \Phi(s')\right)\right]$$

Potential-Based Shaping

■ Proof of sufficiency

define $\hat{Q}_{M'}(s,a) := Q^*_M(s,a) - \Phi(s).$

$$\hat{Q}_{M'}(s,a)$$
$$= \mathrm{E}_{s'}\left[R(s,a,s') + F(s,a,s') + \gamma \max_{a' \in A} \hat{Q}_{M'}(s',a')\right]$$
$$= \mathrm{E}_{s'}\left[R'(s,a,s') + \gamma \max_{a' \in A} \hat{Q}_{M'}(s',a')\right]$$

such. We see that if $\Phi(s) = V^*_M(s)$ (with $\Phi(s_0) = 0$ in the undiscounted case), then Equation (4) tells us that the value function in M' is $V^*_{M'}(s) \equiv 0$ — and this is a particularly easy value function to learn; even

Intuition

■ The Goal

Learn a potential function $\phi(s): S \rightarrow R$ capable of effective reward shaping on tasks sampled from the distribution to accelerate their learning.

■ Optimal Potential Functions

$$V^*_{M'}(s) = V^*_M(s) - \Phi(s),$$

Consequently, if we choose $\Phi(s) = V^*_M(s)$, then $V^*_{M'}(s) \equiv 0$, and "all that would remain to be done would be to learn the non-zero Q-values" (Ng et al., 1999).

$$\begin{aligned} R'(s,a) &= \mathbb{E}_{s'} R'(s,a,s') \\ &= \mathbb{E}_{s'}[R(s,a,s') + \gamma V^*_M(s') - V^*_M(s)] \\ &= Q^*_M(s,a) - \max_a Q^*_M(s,a) \\ &\leq 0, \end{aligned}$$

The Method

■ Meta Learning

meta-training set $\{T_i\}_{i=1}^{N}$ and meta-testing set$\{T_j\}_{j=N+1}^{N+M}$, both drawn from the same task distribution $p(T)$.

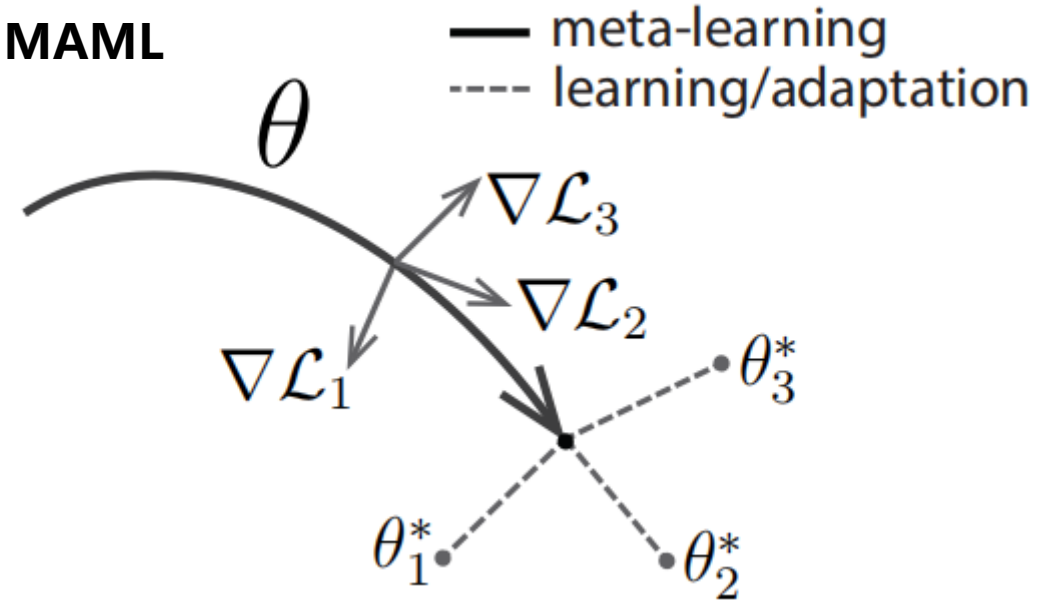
- Model-Agnostic Meta-Learning，which meta-learns an versatile initialization θ of model parameters by:

$$\phi_i \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta),$$

$$\theta \leftarrow \theta - \beta \nabla_\theta \mathbb{E}_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}(f_{\phi_i}).$$

Potential function prior $\Phi(s)$

Task-posterior $\Phi_i(s|\mathcal{T}_i)$



Method

■ Meta-Learning Potential Function Prior

The optimal shaping function $V^*_{M_i}(s)$ is task-specific without a universal optimum for all tasks T. Inspired by MAML' s idea to learn a proper prior capable of fast adaptation to the task-posterior.

$$\min_{\theta} \mathbb{E}_{\mathcal{T}_i} \|\Phi(s;\theta) - V^*_{M_i}(s)\|^2.$$

Dueling DQN $Q^*_{M_i}(s,a) = V^*_{M_i}(s) + A^*_{M_i}(s,a)$

For identifiability of V and A $Q_{\phi_i}(s,a) = V_{\phi_i}(s) + A_{\phi_i}(s,a) - \max_{a'} A_{\phi_i}(s,a').$

$$\mathcal{L}_{\mathcal{T}_i}(Q_{\phi_i}) = \|R_i(s,a,s') + \gamma \max_{a'} Q_{\phi_i}(s',a') - Q_{\phi_i}(s,a)\|^2$$

Method

Algorithm 1 Meta-learning potential function prior

Input: $p(\mathcal{T})$: a distribution over tasks
Input: α, β: step sizes
Output: Learned prior θ
Randomly initialize parameter θ for prior
for `meta_iteration` $= 0, 1, 2...$ **do**
  Sample a batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Initialize replay buffer $\mathcal{D}_i$
    Collect experience $\{s_0, a_0, r_0, \cdots\}$ with ϵ-greedy using $Q_\theta(s, a)$ and add to the replay buffer $\mathcal{D}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(Q_\theta)$ using samples from $\mathcal{D}_i$ ($\mathcal{L}_{\mathcal{T}_i}$ defined in Eqn. (5))
    Compute adapted parameters with gradient descent: $\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(Q_\theta)$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \mathbb{E}_{\mathcal{T}_i} \|Q_\theta(s, a) - Q_{\phi_i}(s, a)\|^2$ with previous samples from all $\mathcal{D}_i$
end for

$$\min_\theta \mathbb{E}_{\mathcal{T}_i} \|Q_\theta(s, a) - Q_{\phi_i}(s, a)\|^2.$$

temporal difference (TD) error

$$\mathcal{L}_{\mathcal{T}_i}(Q_{\phi_i}) = \|R_i(s, a, s') + \gamma \max_{a'} Q_{\phi_i}(s', a') - Q_{\phi_i}(s, a)\|^2$$

Potential function prior $\Phi(s)$

Method

■ Meta-Testing with Potential Function Prior

Find the optimal policy on newly sampled tasks T_j with reward shaping by the learned potential function prior

- **Shaping only** $V_\theta(s)$

- **Adaptation with advantage head**

Jointly adapt $V^*_{M_j}(s)$ to the task-posterior and find the optimal policy efficiently within a few updates, initializing the whole ϕ_j as θ

Method

Algorithm 2 Meta-testing (adaptation with advantage head)

Input: $\mathcal{T}_j$: new task to solve
Input: ϕ_j: task-posterior parameters, initialized as learned prior θ
Output: adapted task-posterior ϕ_j
Initialize replay buffer $\mathcal{D}$
for `gradient_step` $= 0, 1, 2...$ **do**
  Collect experience $\{s_0, a_0, r_0, \cdots\}$ with ϵ-greedy using $A_{\phi_j}(s, a)$ and add to replay buffer $\mathcal{D}$
  Update $A_{\phi_j}(s, a)$ with Eqn. (8) with samples from $\mathcal{D}$ and current potential function $V_{\phi_j}(s)$ for shaping
  Update $V_{\phi_j}(s)$ with Eqn. (9) with samples from $\mathcal{D}$
end for

- Update $A_{\phi_j}(s, a)$ with sampled data from replay buffer:

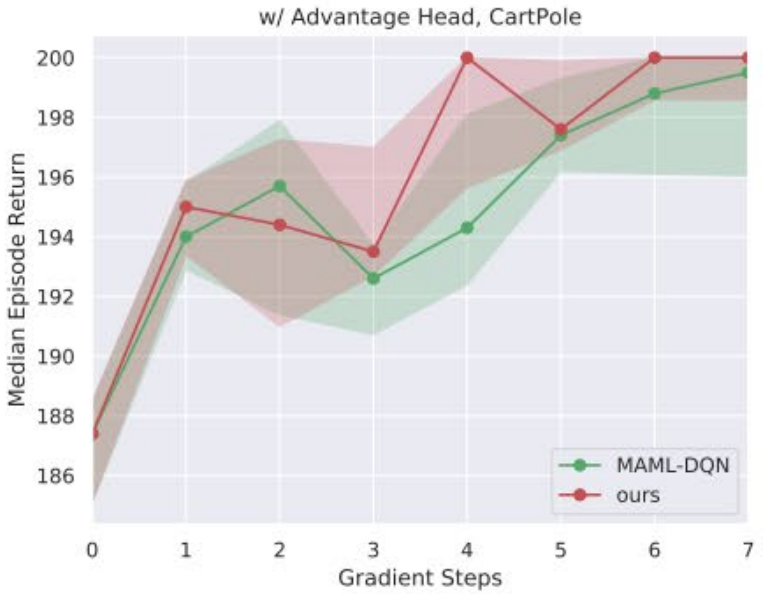
$$\phi_j \leftarrow \phi_j - \alpha \nabla_{\phi_j} \| R'_j(s, a, s') + \gamma \max_{a'} A_{\phi_j}(s', a') - A_{\phi_j}(s, a) \|^2. \quad (8)$$

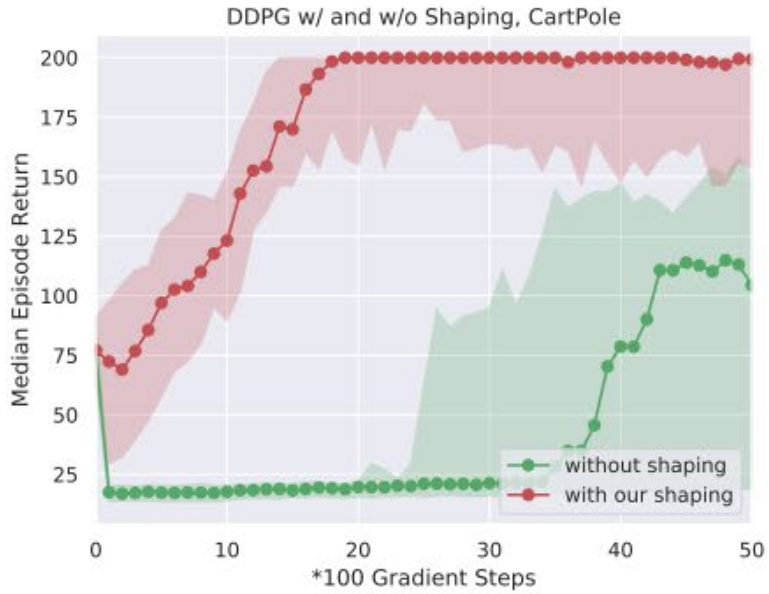
- Update $V_{\phi_j}(s)$ with sampled data from replay buffer:

$$\phi_j \leftarrow \phi_j - \alpha \nabla_{\phi_j} \| V_{\phi_j}(s) - \texttt{stop\_gradient}\big(\max_{a} A_{\phi_j}(s, a) + V_{\phi_j}(s) \big) \|^2. \quad (9)$$

Experiment

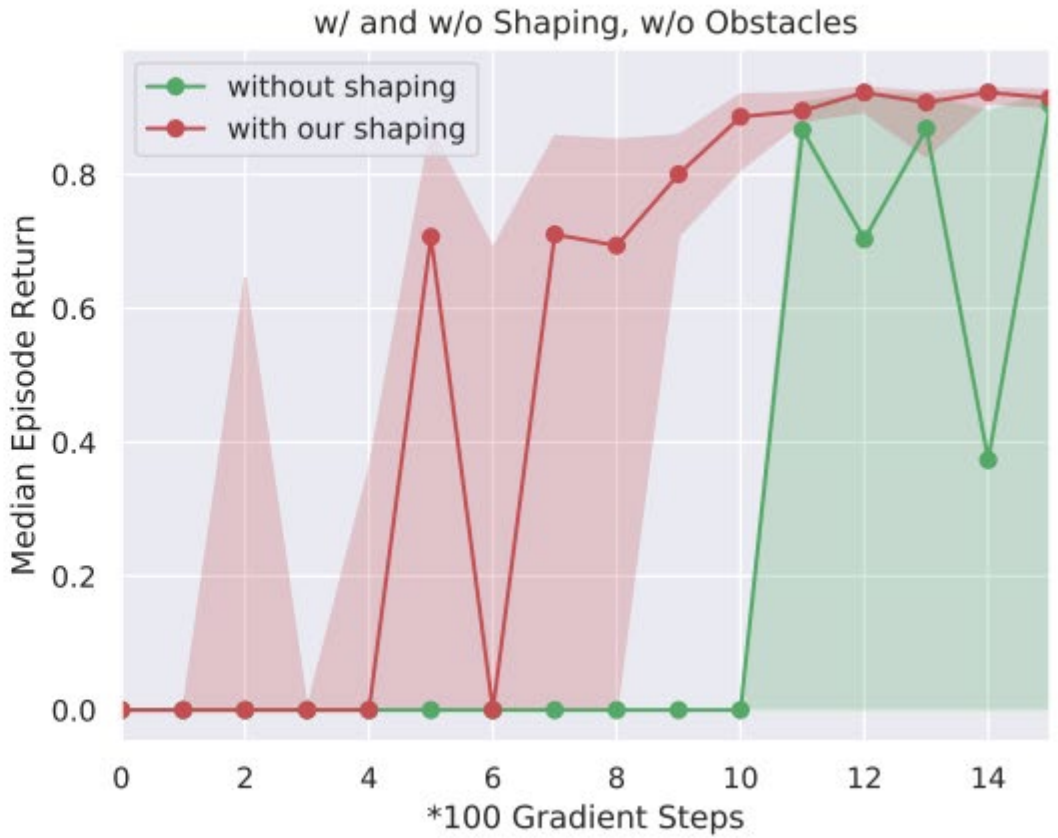
■ Discrete and Continuous CartPoles





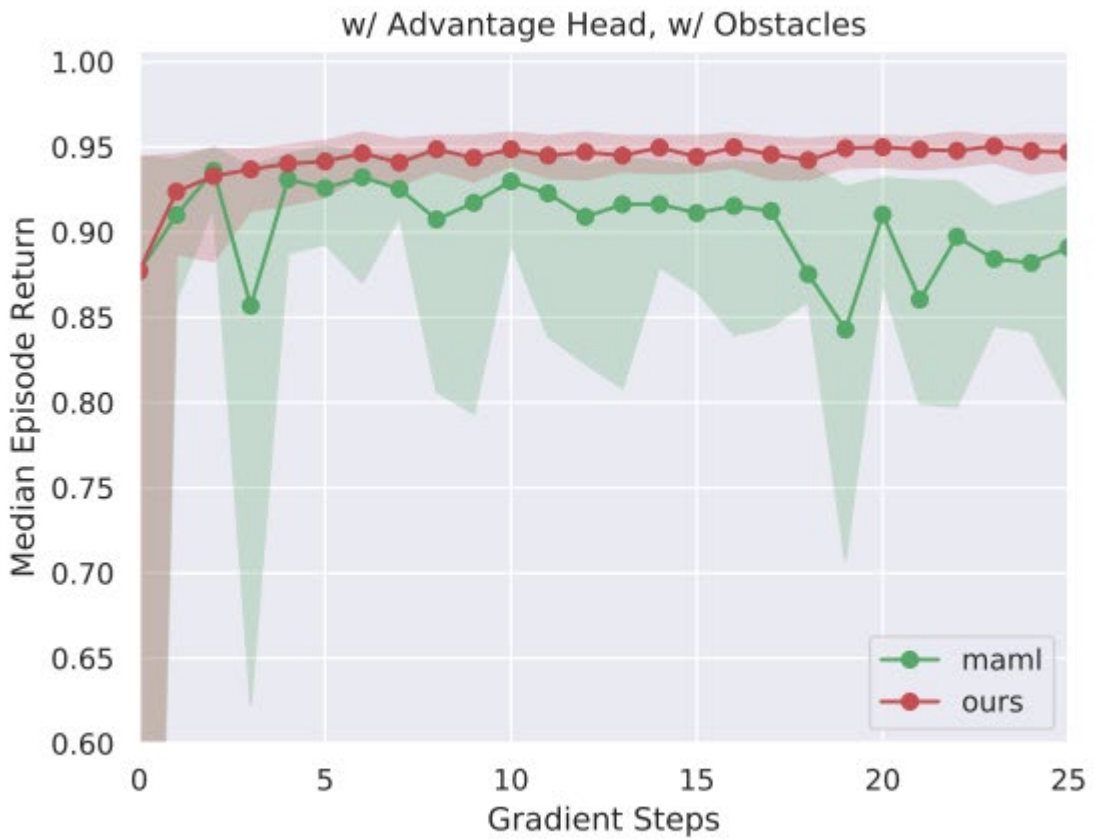
Experiment

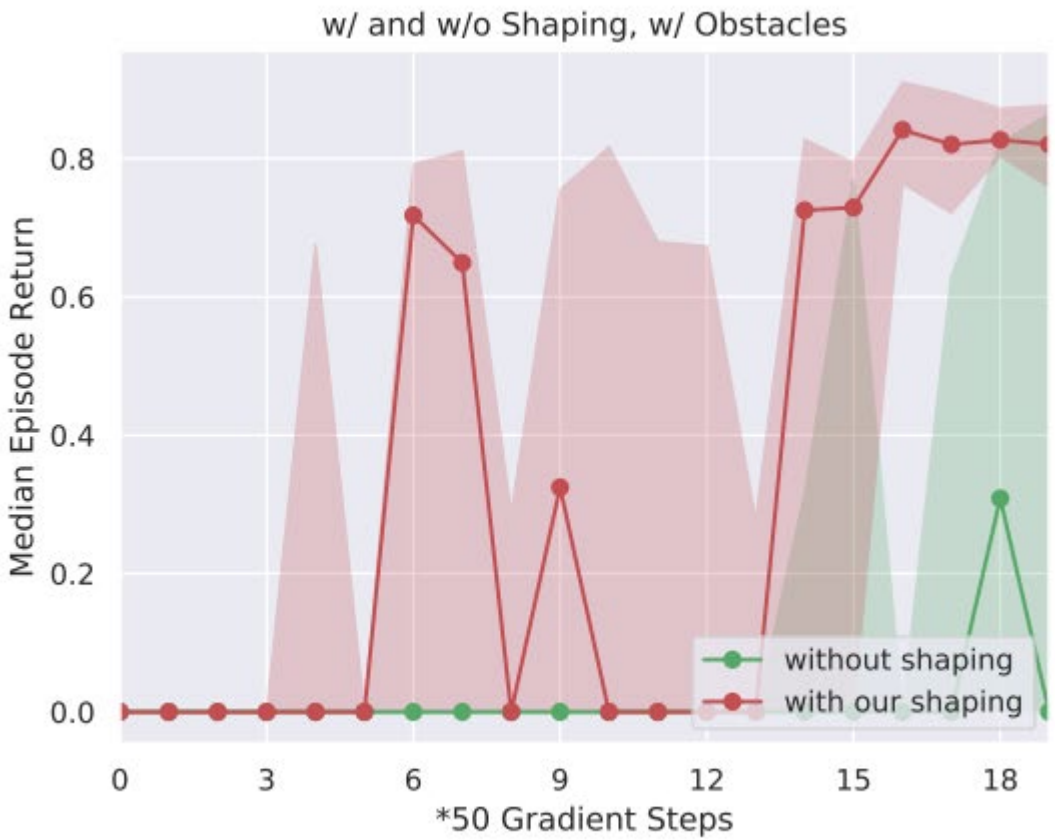
■ Grid Games with Clean Maps



Experiment

■ Grid Games with Obstacles





Q-Value Initialization

Wiewiora. “Potential-based shaping and Q-value initialization are equivalent.” JAIR, 2003

■ He Proved

A reinforcement learner with initial Q-values based on the shaping algorithm’ s potential function make the same updates throughout learning as a learner receiving potential-based shaping rewards.

$$Q(s,a) = Q_0(s,a) \qquad r + F(s,s')$$

$$Q'_0(s,a) = Q_0(s,a) + \Phi(s) \qquad r$$

$\phi(s)$ is Static