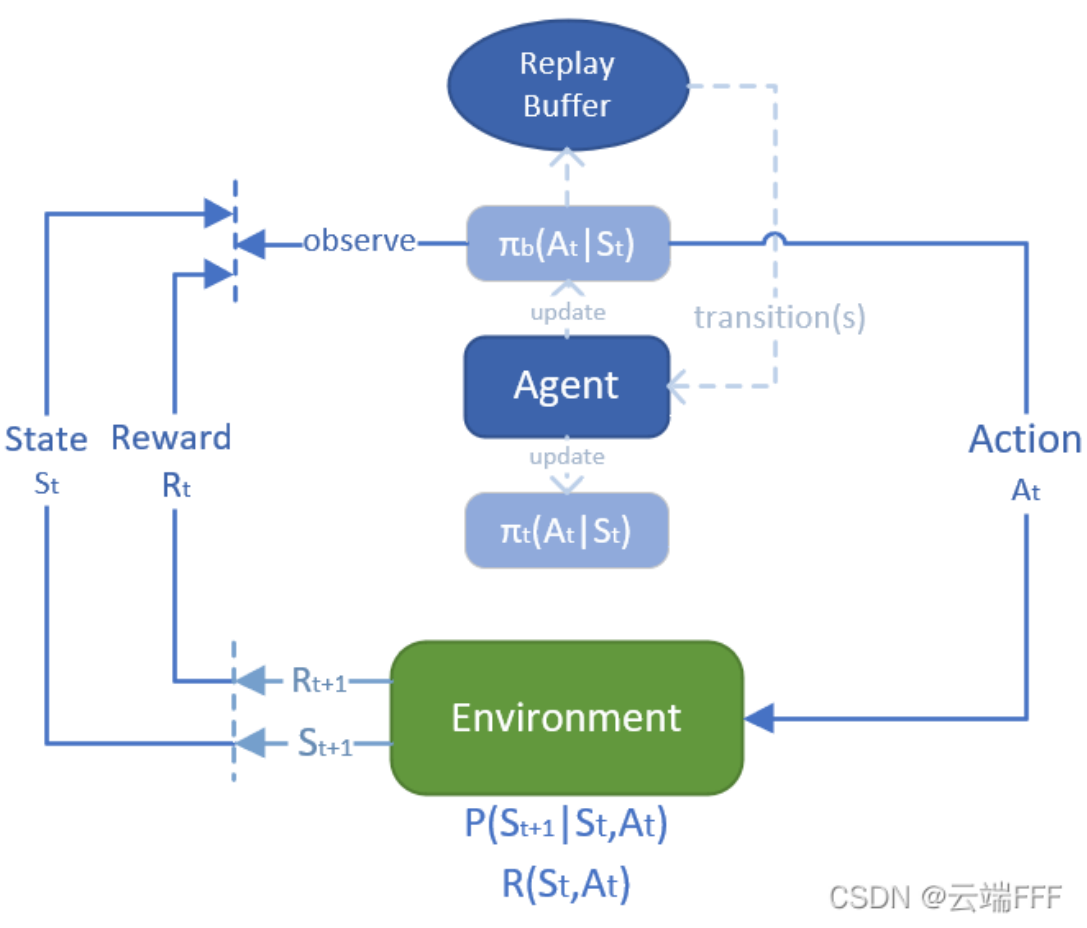
PRIORITIZED EXPERIENCE REPLAY

Tom Schaul, John Quan, Ioannis Antonoglou and David Silver
Google DeepMind
{schaul,johnquan,ioannisa,davidsilver}@google.com

ICLR 2016

Experience Replay



- eliminate circular dependencies
- higher data efficiency
- better data distribution (i.i.d)

Prioritized Experience Repaly

- Key Idea: RL agent can learn more effectively from some transitions than from others

- Measured criterion of transition importance
  - The amount that agent can learn from a transition → not directly accessible
  - TD error → how 'surprising' the transition is

- The sample process need to be stochastic
  - transitions with low TD error are rarely be sampled
  - focus on small subset of the experience (over-fitting)
  - sensitive to noise spike

Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay[J]. arXiv preprint arXiv:1511.05952, 2015.

Stochastic Prioritization

the priority of transition i

$$p(i) = \frac{|\delta(i)|^{\alpha} + \epsilon}{\sum_j (|\delta(j)|^{\alpha} + \epsilon)}$$

between uniform sampling and greedy sampling

$$p(i) = \frac{(\frac{1}{\text{rank(i)}})^{\alpha}}{\sum_j (\frac{1}{\text{rank(j)}})^{\alpha}}$$

power-law distribution with exponent α (more robust)

Annealing the Bias

Tabular method (Q-learning)

$$
\begin{aligned}
Q_*(s,a) &= \max \mathbb{E}_{\pi_*}[G_t | S_t = s, A_t = a] \\
&= \max \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] \\
&= \mathbb{E}[R_{t+1} + \gamma \max_{a'} Q_*(S_{t+1}, a') | S_t = s, A_t = a] \\
&= \sum_{s',r} p(s', r | s, a)[r + \gamma \max_{a'} Q_*(s', a')]
\end{aligned}
$$

NN method (DQN)

$$\delta(i) = Q_\theta(i) - y(i) = Q_\theta(i) - (r + \gamma Q_{\theta'}(s,a))$$

$$\mathcal{L}_i = \mathbb{E}_{s,a\sim\rho(\cdot)}\left[(y_i - Q(s,a,\theta_i)^2)\right]$$

$$\nabla_{\theta_i}\mathcal{L}_i = \mathbb{E}_{s,a\sim\rho(\cdot),s'\sim\text{env}}\left[\left(r + \gamma \max_{a'} Q(s',a',\theta_{i-1}) - Q(s,a,\theta_i)\right)\nabla_{\theta_i} Q(s,a,\theta_i)\right]$$

$$\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \delta}\frac{\partial \delta}{\partial Q}\frac{\partial Q}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial Q}\frac{\partial Q}{\partial \theta} = \nabla_Q \mathcal{L} \cdot \nabla_\theta Q$$

Annealing the Bias

$$\mathcal{L}_{\text{PER}} = w(i)\mathcal{L}(\delta(i))$$

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{p(i)}\right)^{\beta}$$

annealing to 1

$$w(i) = \frac{w_i}{\max_j w_j}$$

for stability reasons

Pseudo Code

Algorithm 1 Double DQN with proportional prioritization

Input: minibatch k, step-size η, replay period K and size N, exponents α and β, budget T.
Initialize replay memory $\mathcal{H} = \emptyset$, $\Delta = 0$, $p_1 = 1$
Observe S_0 and choose $A_0 \sim \pi_\theta(S_0)$
for $t = 1$ **to** T **do**
  Observe S_t, R_t, γ_t
  Store transition $(S_{t-1}, A_{t-1}, R_t, \gamma_t, S_t)$ in $\mathcal{H}$ with maximal priority $p_t = \max_{i<t} p_i$
  if $t \equiv 0 \mod K$ **then**
    for $j = 1$ **to** k **do**
      Sample transition $j \sim P(j) = p_j^\alpha / \sum_i p_i^\alpha$
      Compute importance-sampling weight $w_j = (N \cdot P(j))^{-\beta} / \max_i w_i$
      Compute TD-error $\delta_j = R_j + \gamma_j Q_{\text{target}}(S_j, \arg\max_a Q(S_j, a)) - Q(S_{j-1}, A_{j-1})$
      Update transition priority $p_j \leftarrow |\delta_j|$
      Accumulate weight-change $\Delta \leftarrow \Delta + w_j \cdot \delta_j \cdot \nabla_\theta Q(S_{j-1}, A_{j-1})$
    end for
    Update weights $\theta \leftarrow \theta + \eta \cdot \Delta$, reset $\Delta = 0$
    From time to time copy weights into target network $\theta_{\text{target}} \leftarrow \theta$
  end if
  Choose action $A_t \sim \pi_\theta(S_t)$
end for

An Equivalence between Loss Functions and Non-Uniform Sampling in Experience Replay

Scott Fujimoto, David Meger, Doina Precup
Mila, McGill University
scott.fujimoto@mail.mcgill.ca

NIPS 2020

General Result

Any loss function evaluated with non-uniformly sampled data can be transformed into another uniformly sampled loss function with the same expected gradien

PER can be replaced entirely by this new loss function without impact to empirical performance

This relationship suggests a new branch of improvements to PER by correcting its uniformly sampled loss function equivalent

Preliminaries

$$\delta(i) = Q(i) - y(i), \quad y(i) = r + \gamma Q_{\theta'}(s', a').$$

$$\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \delta}\frac{\partial \delta}{\partial Q}\frac{\partial Q}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial Q}\frac{\partial Q}{\partial \theta} = \boxed{\nabla_Q \mathcal{L}} \cdot \nabla_\theta Q$$

1. L_1 损失：$\mathcal{L}_{L1}(\delta(i)) = |\delta(i)|$，梯度为 $\nabla_Q \mathcal{L}_{L1}(\delta(i)) = \text{sign}(\delta(i))$
2. MSE 损失：$\mathcal{L}_{\text{MSE}}(\delta(i)) = 0.5\delta(i)^2$，梯度为 $\nabla_Q \mathcal{L}_{\text{MSE}}(\delta(i)) = \delta(i)$
3. Huber 损失：

$$\mathcal{L}_{\text{Huber}} = \begin{cases} 0.5\delta(i)^2 & if \ \ |\delta(i)| \le k, \\ k(|\delta(i)| - 0.5k) & \text{otherwise} \end{cases} \tag{2}$$

通常设置 $k = 1$，这样根据 $|\delta(i)|$ 的取值，Huber 损失的梯度等价与 MSE 或 L_1 损失。这种损失函数可以看作 0 附近更平滑的 L_1 损失

Sampling and Loss Functions

importance sampling ratio

$$\underbrace{\mathbb{E}_{i\sim\mathcal{D}_1}[\nabla_Q \mathcal{L}_1(\delta(i))]}_{\text{expected gradient of } \mathcal{L}_1 \text{ under } \mathcal{D}_1} = \mathbb{E}_{i\sim\mathcal{D}_2}\left[\frac{p_{\mathcal{D}_1}(i)}{p_{\mathcal{D}_2}(i)}\nabla_Q \mathcal{L}_1(\delta(i))\right].$$

$$\nabla_Q \mathcal{L}_2(\delta(i)) = \frac{p_{\mathcal{D}_1}(i)}{p_{\mathcal{D}_2}(i)}\nabla_Q \mathcal{L}_1(\delta(i))$$

$$\mathbb{E}_{\mathcal{D}_1}[\nabla_Q \mathcal{L}_1(\delta(i))] = \mathbb{E}_{\mathcal{D}_2}[\nabla_Q \mathcal{L}_2(\delta(i))]$$

Sampling and Loss Functions

使用优先采样重放的 L1 损失 和 使用均匀采样重放的的 MSE 损失 具有相同的期望梯度方向

$$p(i) = \frac{|\delta(i)|}{\sum_{j \in \mathcal{B}} |\delta(j)|}$$

$$\underbrace{\mathbb{E}_{\mathcal{U}}[\nabla_Q \mathcal{L}_{\text{MSE}}(\delta(i))]}_{\text{expected gradient of MSE under } \mathcal{U}} = \underbrace{\mathbb{E}_{\mathcal{D}_2}\left[\frac{\sum_j \delta(j)}{N|\delta(i)|}\delta(i)\right]}_{\text{by Equation (5)}} \propto \mathbb{E}_{\mathcal{D}_2} \underbrace{[\text{sign}(\delta(i))]}_{\nabla_Q \mathcal{L}_{\text{L1}}(\delta(i))} = \underbrace{\mathbb{E}_{\mathcal{D}_2}[\nabla_Q \mathcal{L}_{\text{L1}}(\delta(i))]}_{\text{expected gradient of L1 under } \mathcal{D}_2} \tag{6}$$

$$\frac{p_{\mathcal{D}_1}(i)}{p_{\mathcal{D}_2}(i)} \nabla_Q \mathcal{L}_{\text{MSE}}(\delta(i)) = \frac{1}{N} \frac{\sum_j |\delta(i)|}{|\delta(i)|}\delta(i) = \frac{\sum_j |\delta(i)|}{N|\delta(i)|}\delta(i) = \frac{\sum_j |\delta(i)|}{N}\text{sign}(\delta(i))$$

Theorem 1

Theorem 1：给定大小为 N 的数据集 $\mathcal{B}$，损失 $\mathcal{L}_1, \mathcal{L}_2$ 和某种优先级分布 pr，若 $\nabla_Q \mathcal{L}_1(\delta(i)) = \frac{1}{\lambda} pr(i) \nabla_Q \mathcal{L}_2(\delta(i))$，其中 $\lambda = \frac{\sum_j pr(j)}{N}$，则从 $\mathcal{B}$ 中均匀采样的样本 i 对应的 $\mathcal{L}_1(\delta(i))$ 的期望梯度，和从 $\mathcal{B}$ 中按 pr 优先采样的样本 i 对应的 $\mathcal{L}_2(\delta(i))$ 的期望梯度相等

- Proof:

$$
\begin{aligned}
\mathbb{E}_{i\sim\mathcal{B}}[\nabla_Q \mathcal{L}_1(\delta(i))] &= \frac{1}{N} \sum_i \nabla_Q \mathcal{L}_1(\delta(i)) \\
&= \frac{1}{N} \sum_i \frac{N}{\sum_j pr(j)} pr(i) \nabla_Q \mathcal{L}_2(\delta(i)) \qquad \text{(假设条件)} \\
&= \sum_i \frac{pr(i)}{\sum_j pr(j)} \nabla_Q \mathcal{L}_2(\delta(i)) \\
&= \mathbb{E}_{i\sim pr}[\nabla_Q \mathcal{L}_2(\delta(i))]
\end{aligned}
$$

Corollary 1

Corollary 1：若 $\mathcal{L}_1(\delta(i)) = \frac{1}{\lambda}|pr(i)|_{\times}\mathcal{L}_2(\delta(i))$ for all i，其中 $\lambda = \frac{\sum_j pr(j)}{N}$，$|\cdot|_{\times}$ 是停止梯度符号，则 Theorem 1 对任意均匀采样的损失 $\mathcal{L}_1$ 和按任意按 pr 非均匀采样的损失 $\mathcal{L}_2$ 均成立

- proof：

$$\begin{aligned}\nabla_Q \mathcal{L}_1(\delta(i)) &= \nabla_Q \frac{1}{\lambda}|pr(i)|_{\times}\mathcal{L}_2(\delta(i)) \\ &= \frac{1}{\lambda}pr(i)\nabla_Q \mathcal{L}_2(\delta(i))\end{aligned}$$

满足 Theorem 1 成立条件，得证

Corollary 2

Corollary 2：若 $\text{sign}(\nabla_Q \mathcal{L}_1(\delta(i))) = \text{sign}(\nabla_Q \mathcal{L}_2(\delta(i)))$ 且 $pr(i) = \frac{\nabla_Q \mathcal{L}_1(\delta(i))}{\nabla_Q \mathcal{L}_2(\delta(i))}$ for all i，则 Theorem 1 对任意均匀采样的损失 $\mathcal{L}_1$ 和任意按 pr 非均匀采样的损失 $\lambda\mathcal{L}_2$ 均成立，其中 $\lambda = \frac{\sum_j pr(j)}{N}$

- 由于采样优先级不能是负的，这里 $pr(i)$ 必须设计为非负的，即 $\text{sign}(pr(i)) = 1$，因此需要满足条件 $\text{sign}(\nabla_Q \mathcal{L}_1(\delta(i))) = \text{sign}(\nabla_Q \mathcal{L}_2(\delta(i)))$。通常，所有旨在最小化 Q 输出与给定目标间距离的损失函数都满足此条件

  > 因为有同样的目标，优化方向是相同的，梯度方向也应大致相同

  这时非均匀采样的作用类似于重要性采样，它对 $\mathcal{L}_2$ 做重加权来匹配 $\mathcal{L}_1$ 的期望梯度。

- proof：给定 $\text{sign}(\nabla_Q \mathcal{L}_1(\delta(i))) = \text{sign}(\nabla_Q \mathcal{L}_2(\delta(i)))$

$$\begin{aligned}\frac{1}{\lambda} pr(i) \nabla_Q \lambda \mathcal{L}_2(\delta(i)) &= \frac{\lambda}{\lambda} \frac{\nabla_Q \mathcal{L}_1(\delta(i))}{\nabla_Q \mathcal{L}_2(\delta(i))} \nabla_Q \mathcal{L}_2(\delta(i)) \\ &= \nabla_Q \mathcal{L}_1(\delta(i))\end{aligned}$$

  满足 Theorem 1 成立条件，得证

Theorem 2

$$pr(i) = \frac{\nabla_Q \mathcal{L}_1(\delta(i))}{\nabla_Q \mathcal{L}_2(\delta(i))} \text{ for all } i,$$

Theorem 2：给定大小为 N 的数据集 $\mathcal{B}$ 和损失函数 $\mathcal{L}_1$，样本 $i \in \mathcal{B}$ 依照优先级 pr 进行采样，并有 $\lambda = \frac{\sum_j pr(j)}{N}$，考察损失 $\lambda\mathcal{L}_2$，使得 Theorem 1 成立（即期望梯度相等，且有 $\frac{1}{\lambda} pr(i)\nabla_Q \lambda\mathcal{L}_2(\delta(i)) = pr(i)\nabla_Q \mathcal{L}_2(\delta(i)) = \nabla_Q \mathcal{L}_1(\delta(i)))$ 。**当** $\mathcal{L}_2 = \mathcal{L}_{L_1}$ **且** $pr(i) = |\nabla_Q \mathcal{L}_1(\delta(i))|$ **时，** $\nabla_Q \lambda\mathcal{L}_2(\delta(i)$ **的方差最小**

这些结果表明，通过使用 L_1 损失和相应的优先级采样方案，可以在保持期望梯度不变的情况，减少任何损失函数的方差

Observation 1：给定大小为 N 的数据集 $\mathcal{B}$ 和损失函数 $\mathcal{L}_1$，样本 i 以优先级 $pr(i) = |\nabla_Q \mathcal{L}_1(\delta(i))|$ 进行优先采样，有 $\lambda = \frac{\sum_j pr(j)}{N}$，则 $\lambda\mathcal{L}_{L_1}(\delta(i))$ 的梯度的方差小于等于均匀采样的 $\mathcal{L}_1(\delta(i))$ 的梯度的方差

Corrections to PER

Theorem 3：当与 PER 一起使用时，损失 $\frac{1}{\tau}|\delta(i)|^\tau$ 的期望梯度（其中 $\tau > 0$）等于使用均匀采样重放时以下损失的期望梯度

$$\mathcal{L}^\tau_{\mathrm{PER}}(\delta(i)) = \frac{\eta N}{\tau + \alpha - \alpha\beta}|\delta(i)|^{\tau+\alpha-\alpha\beta}, \quad \eta = \frac{\min_j |\delta(j)|^{\alpha\beta}}{\sum_j |\delta(j)|^\alpha} \tag{7}$$

- proof：根据 PER 定义，有

$$p(i) = \frac{|\delta(i)|^\alpha + \epsilon}{\sum_j (|\delta(j)|^\alpha + \epsilon)}, w(i) = \frac{(\frac{1}{N} \cdot \frac{1}{p(i)})^\beta}{\max_j (\frac{1}{N} \cdot \frac{1}{p(j)})^\beta}$$

下面考察使用 PER 时损失函数 $\frac{1}{\tau}|\delta(i)|^\tau$ 的期望梯度

$$\begin{aligned}
\mathbb{E}_{i\sim\mathrm{PER}}\left[\nabla_Q w(i)\frac{1}{\tau}|\delta(i)|^\tau\right] &= \sum_{i\in\mathcal{B}} w(i)p(i)\nabla_Q \frac{1}{\tau}|\delta(i)|^\tau \\
&= \sum_{i\in\mathcal{B}} \frac{\left(\frac{1}{N}\cdot\frac{1}{p(i)}\right)^\beta}{\max_{j\in\mathcal{B}}\left(\frac{1}{N}\cdot\frac{1}{p(j)}\right)^\beta} \frac{|\delta(i)|^\alpha}{\sum_{j\in\mathcal{B}}|\delta(j)|^\alpha}\mathrm{sign}(\delta(i))|\delta(i)|^{\tau-1} \\
&= \frac{1}{\max_{j\in\mathcal{B}}\frac{1}{|\delta(j)|^{\alpha\beta}}\sum_{j\in\mathcal{B}}|\delta(j)|^\alpha} \sum_{i\in\mathcal{B}} \frac{|\delta(i)|^{\tau+\alpha-1}\mathrm{sign}(\delta(i))}{|\delta(i)|^{\alpha\beta}} \\
&= \eta\sum_{i\in\mathcal{B}}\mathrm{sign}(\delta(i))|\delta(i)|^{\tau+\alpha-\alpha\beta-1}.
\end{aligned} \tag{11}$$

现在考虑 $\mathcal{L}^\tau_{\mathrm{PER}}(\delta(i))$ 的期望梯度

$$\begin{aligned}
\mathbb{E}_{i\sim\mathcal{B}}\left[\nabla_Q \mathcal{L}^\tau_{\mathrm{PER}}(\delta(i))\right] &= \frac{1}{N}\sum_{i\in\mathcal{B}} \frac{\eta N}{\tau+\alpha-\alpha\beta}\nabla_Q|\delta(i)|^{\tau+\alpha-\alpha\beta} \\
&= \eta\sum_{i\in\mathcal{B}}\mathrm{sign}(\delta(i))|\delta(i)|^{\tau+\alpha-\alpha\beta-1}.
\end{aligned} \tag{12}$$

二者相等，证毕

Corollary 3

- 注意到 DQN 传统上使用 Huber 损失，对于 DQN 的使用的 PER 优先采样，可以将其改写为以下均匀采样的损失函数的形式，具有相同的期望梯度

Corollary 3：当与 PER 一起使用时，Huber 损失的期望梯度等于使用**均匀采样重放时，以下损失的期望梯度**（Theorem 3 带入 $\tau = 1$ 和 $\tau = 2$ 即可）

$$\mathcal{L}_{\text{PER}}^{\text{Huber}}(\delta(i)) = \frac{\eta N}{\tau + \alpha - \alpha\beta}|\delta(i)|^{\tau+\alpha-\alpha\beta}, \qquad \tau = \begin{cases} 2 & \textit{if } |\delta(i)| \leq 1, \\ 1 & \textit{otherwise,} \end{cases} \qquad \eta = \frac{\min_j |\delta(j)|^{\alpha\beta}}{\sum_j |\delta(j)|^{\alpha}}. \tag{8}$$

Corollary 3

- 为了理解 Corollary 3 的意义及其对 PER 目标的描述，首先考虑以下两个关于 MSE 和 L_1 的观察现象

1. **Observation 2** (MSE)：$\mathcal{B}(s,a) \subset \mathcal{B}$ 是包含 (s,a) 的 transition 的子集，$\delta(i) = Q(i) - y(i)$，若 $\nabla_Q \mathbb{E}_{i\sim\mathcal{B}(s,a)}[0.5\delta(i)^2] = 0$，则 $Q(s,a) = \text{mean}_{i\in\mathcal{B}(s,a)} y(i)$

Proof.

$$
\begin{aligned}
& \mathbb{E}_{i\sim\mathcal{B}(s,a)}[\nabla_Q 0.5|\delta(i)|^2] = 0 \\
\Rightarrow\ & \mathbb{E}_{i\sim\mathcal{B}(s,a)}[\delta(i)] = 0 \\
\Rightarrow\ & \frac{1}{N}\sum_{i\in\mathcal{B}(s,a)} Q(s,a) - y(i) = 0 \\
\Rightarrow\ & Q(s,a) - \frac{2c}{N}\sum_{i\in\mathcal{B}(s,a)} y(i) = 0 \\
\Rightarrow\ & Q(s,a) = \frac{1}{N}\sum_{i\in\mathcal{B}(s,a)} y(i).
\end{aligned}
\tag{14}
$$

2. **Observation 3** (L_1)：$\mathcal{B}(s,a) \subset \mathcal{B}$ 是包含 (s,a) 的 transition 的子集，$\delta(i) = Q(i) - y(i)$，若 $\nabla_Q \mathbb{E}_{i\sim\mathcal{B}(s,a)}[|\delta(i)|] = 0$，则 $Q(s,a) = \text{median}_{i\in\mathcal{B}(s,a)} y(i)$

Proof.

$$
\begin{aligned}
& \mathbb{E}_{i\sim\mathcal{B}(s,a)}[\nabla_Q|\delta(i)|] = 0 \\
\Rightarrow\ & \mathbb{E}_{i\sim\mathcal{B}(s,a)}[\text{sign}(\delta(i))] = 0 \\
\Rightarrow\ & \sum_{i\in\mathcal{B}(s,a)} \mathbb{1}\{Q(s,a) \le y(i)\} = \sum_{i\in\mathcal{B}(s,a)} \mathbb{1}\{Q(s,a) \ge y(i)\} \\
\Rightarrow\ & Q(s,a) = \text{median}_{i\in\mathcal{B}(s,a)} y(i).
\end{aligned}
\tag{15}
$$

Corollary 3

存在这种可能性：介于 MSE 和 L_1 之间的损失函数可以在“鲁棒性”和“正确性”之间取得平衡。从等式7可见，这就是 PER 和 L_1 损失结合时的作用，因为损失中有幂次 $1+\alpha-\alpha\beta \in [1,2]$，其中 $\alpha \in (0,1], \beta \in [0,1]$。然而，当结合 MSE 时，若 $\beta < 1$，则当 $\alpha \in (0,1]$ 时 $2+\alpha-\alpha\beta > 2$，这意味着**虽然 MSE 单独使用时通过均值来最小化损失，但当与PER结合时，损失将通过一些有利于异常值的表达式最小化。**这种偏差解释了 PER 在使用 MSE 的连续控制任务中使用标准算法时性能不佳

使用 L_1 损失，优先级设为 $pr(i) = |\delta(i)|^{\alpha}$

Theorem 1 $\nabla_Q \mathcal{L}_1(\delta(i)) = \frac{1}{\lambda} pr(i) \nabla_Q \mathcal{L}_2(\delta(i)) = \frac{1}{\lambda} |\delta(i)|^{\alpha} \cdot |\delta(i)| \propto |\delta(i)|^{\alpha+1}$

然而，在实践中，L1 损失可能并不可取，因为每次更新都步进一个固定大小的步长，如果学习率太高，可能会超出目标（overstepping the target）

LAP & PAL

$$p(i) = \frac{\max(|\delta(i)|^\alpha, 1)}{\sum_j \max(|\delta(j)|^\alpha, 1)}, \qquad \mathcal{L}_{\text{Huber}}(\delta(i)) = \begin{cases} 0.5\delta(i)^2 & \text{if } |\delta(i)| \le 1, \\ |\delta(i)| & \text{otherwise.} \end{cases}$$

1. $|\delta(i)| > 1$ 时：L_1 损失，优先级 $pr(i) = |\delta(i)|^\alpha$
2. $|\delta(i)| \le 1$ 时：MSE 损失，均匀采样

Corollary 1 ↓ : $\mathcal{L}_{\text{PLA}}(\delta(i)) = \frac{1}{\lambda}|pr(i)|_\times \mathcal{L}_{\text{Huber}}(\delta(i))$

$$\mathcal{L}_{\text{PAL}}(\delta(i)) = \frac{1}{\lambda}\begin{cases} 0.5\delta(i)^2 & \text{if } |\delta(i)| \le 1, \\ \frac{|\delta(i)|^{1+\alpha}}{1+\alpha} & \text{otherwise,} \end{cases} \qquad \lambda = \frac{\sum_j \max(|\delta(j)|^\alpha, 1)}{N}.$$

1. $|\delta(i)| \le 1$ 时，$pr(i) = 1$，$\mathcal{L}_{\text{PLA}}(\delta(i)) = \frac{1}{\lambda} \cdot 1 \cdot 0.5\delta(i)^2 = \frac{1}{\lambda} \cdot 0.5\delta(i)^2$
2. $|\delta(i)| > 1$ 时，$pr(i) = |\delta(i)|^\alpha$，$\mathcal{L}_{\text{PLA}}(\delta(i)) = \frac{1}{\lambda}||\delta(i)|^\alpha|_\times |\delta(i)|$，对它求梯度是$\nabla_Q \mathcal{L}_{\text{PLA}} = \frac{1}{\lambda}|\delta(i)|^\alpha \nabla_Q |\delta(i)|$，为了表示简便，这里停止梯度的操作改为等效的允许梯度操作，即 $\nabla_Q \frac{1}{\lambda}\frac{|\delta(i)|^{1+\alpha}}{1+\alpha} = \frac{1}{\lambda}|\delta(i)|^\alpha \nabla_Q |\delta(i)|$

Experiments

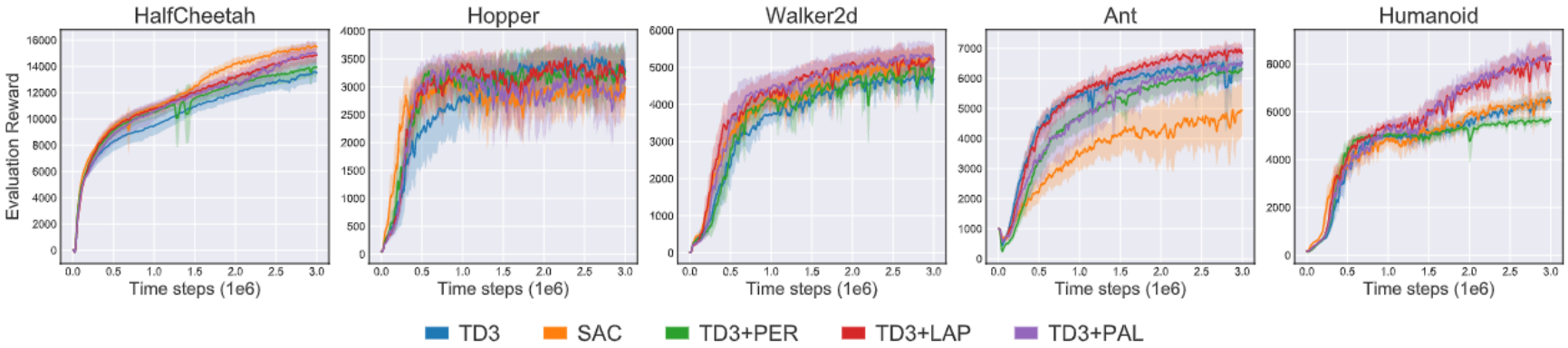


Figure 1: Learning curves for the suite of OpenAI gym continuous control tasks in MuJoCo. Curves are averaged over 10 trials, where the shaded area represents a 95% confidence interval over the trials.

Table 1: Average performance over the last 10 evaluations and 10 trials. ± captures a 95% confidence interval. Scores are bold if the confidence interval intersects with the confidence interval of the highest performance, except for Hopper and Walker2d where all scores satisfy this condition.

	TD3	SAC	TD3 + PER	TD3 + LAP	TD3 + PAL
HalfCheetah	13570.9 ± 794.2	**15511.6 ± 305.2**	13927.8 ± 683.9	**14836.5 ± 532.2**	**15012.2 ± 885.4**
Hopper	3393.2 ± 381.9	2851.6 ± 417.4	3275.5 ± 451.8	3246.9 ± 463.4	3129.1 ± 473.5
Walker2d	4692.4 ± 423.6	5234.4 ± 346.1	4719.1 ± 492.0	5230.5 ± 368.2	5218.7 ± 422.6
Ant	6469.9 ± 200.3	4923.6 ± 882.3	6278.7 ± 311.3	**6912.6 ± 234.4**	**6476.2 ± 640.2**
Humanoid	6437.5 ± 349.3	6580.9 ± 296.6	5629.3 ± 174.4	**7855.6 ± 705.9**	**8265.9 ± 519.0**

Experiments

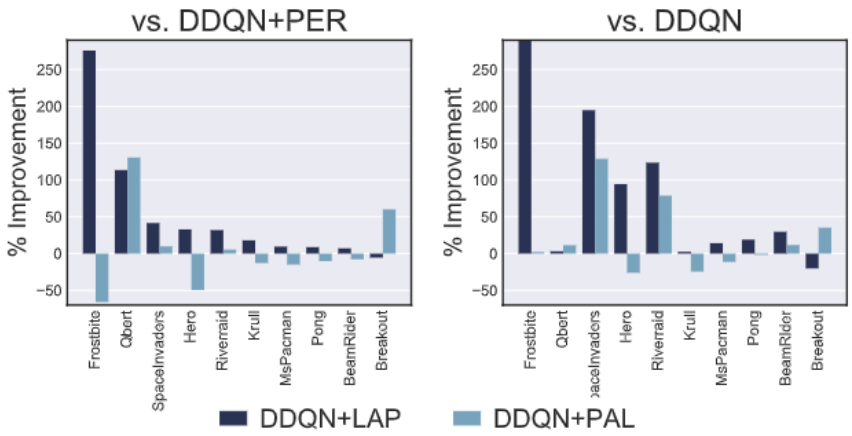


Figure 2: We display the percentage improvement of final scores achieved by DDQN + PAL and DDQN + LAP when compared to DDQN + PER (left) and DDQN (right). Some extreme values are visually clipped.

Table 2: Mean and median percentage improvement of final scores achieved over DDQN and DDQN + PER across 10 Atari games.

	Mean % Gain	Median % Gain
	vs. DDQN + PER	
DDQN	-8.06%	-13.00%
DDQN + LAP	**+53.38%**	**+24.98%**
DDQN + PAL	+4.50%	-8.96%
	vs. DDQN	
DDQN + PER	+37.65%	+15.24%
DDQN + LAP	**+148.16%**	**+24.35%**
DDQN + PAL	+20.46%	+6.79%

The End

thanks