离线强化学习的乐观视角

摘要

使用固定的离线数据集记录互动的异策略强化学习（RL）是实际应用中的一个重要考虑因素。本文研究了使用 DQN 重播数据集的离线 RL，该数据集包含 DQN 算法在 60 个雅达利 2600 游戏中的整个重播经验。我们展示了即使仅在这个固定数据集上训练，最近的 off 深度 RL 算法也能胜过完全训练过的 DQN 算法。为了增强离线设置中的泛化能力，我们提出了随机集成混合（REM），这是一种强化 Q-learning 算法，对多个 Q 值估计的随机凸组合强制执行最优 Bellman 一致性。在 DQN 重播数据集上训练的离线 REM 超越了强 RL 基线。消融研究突显了我们积极结果中离线数据集的大小和多样性以及算法选择的作用。总体而言，这里的结果呈现了一个乐观的观点，即在足够大且多样化的离线数据集上使用强化 RL 算法可以获得高质量的策略。

关键词：强化学习；离线强化学习；雅达利；DQN

1 引言

深度学习成功的主要原因之一是具有大量和多样化的数据集，例如 ImageNet [1]，用于训练表达能力强的深度神经网络。相比之下，大多数强化学习算法 [2] 假设智能体与在线环境或模拟器进行交互，并从自己收集的经验中学习。这限制了在线强化学习在复杂的现实世界问题中的适用性，因为主动数据收集意味着每次实验都需要从头开始收集大量多样化的数据，这可能昂贵、不安全，或者需要一个难以构建的高保真模拟器 [3]。

离线强化学习关注的是从一个固定的轨迹数据集中学习策略的问题，而不需要与环境进行进一步的交互。这种设置可以利用大量已记录的互动，解决现实世界的决策问题，如机器人技术 [4] [5]，自动驾驶 [6]，推荐系统 [7] [8]，以及医疗保健 [9]。对这些数据集的有效利用不仅可以使现实世界的强化学习更加实用，还可以通过整合多样化的先前经验实现更好的泛化。

在离线强化学习中，智能体不会从在线环境中接收任何新的纠正性反馈，并且在评估期间需要从固定的互动数据集中推广到新的在线互动。原则上，异策略算法可以从任何策略收集的数据中学习，然而，最近的研究 [10] [11] [12] [13] 呈现了一个令人沮丧的观点，即标准的异策略深度 RL 算法在离线设置中会发散或表现不佳。这些论文通过使学到的策略规范化以保持接近离线轨迹的训练数据集提出了补救方法。此外，Zhang 和 Sutton（2017）[14] 声称，大型重放缓冲区甚至可能由于其” 异策略性” 而损害异策略算法的性能。

相比之下，本文提出了一个对离线强化学习的乐观视角，即在足够大且多样化的数据集的情况下，具有鲁棒性的强化学习算法，即使没有显式校正分布不匹配，也可以导致高质量的策略。本文的贡献可以总结如下：

- 提出了一种用于评估雅达利 2600 游戏算法的离线强化学习设置 [15]，基于 DQN 算法 [16] 记录的重播数据，其中每个游戏包含 5000 万个（观察，动作，奖励，下一个观察）元组。这种设置大大减少了实验的计算成本，并通过使用固定数据集标准化训练，有助于提高可重现性。DQN 重播数据集和我们的代码已经发布，以便在共同的基础上进行离线 RL 算法的优化。

- 与最近的研究相反，我们展示了最新的异策略 RL 算法仅在离线数据上训练就可以取得成功。例如，仅在 DQN 重播数据集上训练的离线 QR-DQN [17] 优于 DQN 重播数据集中的最佳策略。这种差异归因于离线数据集的大小和多样性以及 RL 算法的选择。

- 提出了一种强大的 Q 学习算法，称为随机集成混合（REM），它通过对多个 Q 值估计的随机凸组合强制执行最优 Bellman 一致性。离线 REM 在离线设置中展现出强大的泛化性能，并优于离线 QR-DQN。与在线 C51 [18] 的比较，这是一个强大的 RL 基线，说明了使用 REM 开发已记录的 DQN 数据的相对增益的规模。

2 相关工作

2.1 异策略强化学习

强化学习中的交互式环境通常被描述为马尔可夫决策过程（MDP）(S, A, R, P, γ) [19]，其中包括状态空间 S，动作空间 A，随机奖励函数 $R(s,a)$，转移动态 $P(s'|s,a)$，以及折扣因子 $\gamma \in [0,1)$。随机策略 $\pi(\cdot|s)$ 将每个状态 $s \in S$ 映射到一个动作分布。

对于按照策略　运行的代理，动作值函数（记作 $Q^\pi(s,a)$）被定义为累积折扣未来奖励的期望，即，

$$Q^\pi(s,a) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right], \tag{1}$$
$$s_0 = s, a_0 = a, s_t \sim P(\cdot|s_{t-1}, a_{t-1}), a_t \sim \pi(\cdot|s_t)$$

RL 的目标是找到一个最优策略 π^*，该策略能够获得最大的期望回报，即对于所有的 π、s 和 a，都有 $Q_{\pi^*}(s,a) \geq Q_\pi(s,a)$。贝尔曼最优性方程 [20] 通过最优的 Q-值，表示为 $Q^* = Q_{\pi^*}$，来刻画最优策略，具体表达式为：

$$Q^*(s,a) = \mathbb{E}R(s,a) + \gamma \mathbb{E}_{s' \sim P} \max_{a' \in A} Q^*(s', a') \tag{2}$$

要从与环境的交互中学习策略，Q-learning [21] 通过迭代地改进对 Q^* 的近似估计，表示为 Q_θ，通过反复地将（2）的左侧进行回归到由（2）的右侧样本定义的目标值。对于大而复杂的状态空间，可以使用神经网络作为函数逼近器来获取近似的 Q 值。为了进一步稳定优化过程，可以使用具有冻结参数的目标网络 $Q_{\theta'}$ 来计算学习目标 [22]。目标网络的参数 θ' 在固定数量的时间步长之后更新为当前的 Q 网络参数　。

DQN [16] [22] 使用卷积神经网络 [23] 对 Q_θ 进行参数化，并在遵循对于数据收集而言是 ϵ-贪心的 Q_θ 的情况下，使用目标网络进行 Q-learning。DQN 通过最小化迭代训练期间收集

的经验回放缓冲区 $\mathcal{D}$ [24] 中的代理过去经验元组 $(s,\ a,\ r,\ s')$ 的小批量上的时间差（TD）误差 Δ_θ 来学习。

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'\sim\mathcal{D}}\left[l_\lambda(\Delta_\theta(s,a,r,s'))\right], \tag{3}$$
$$\Delta_\theta(s,a,r,s') = Q_\theta(s,a) - r - \gamma \max_{a'} Q_{\theta'}(s',a')$$

Q-learning 是一种异策略算法 [2]，因为可以计算学习目标，而无需考虑经验是如何生成的。

最近一系列离策略深度强化学习算法作为本文的强基线，包括分布式强化学习方法 [18] [25]。这些算法对于每个状态-动作对估计返回的密度，表示为 Z^π(s, a)，而不是直接估计均值 Q^π(s, a)。因此，可以表达一种分布式贝尔曼最优性，如下：

$$Z^*(s,a) \overset{D}{=} r + \gamma Z^*(s', argmax_{a'\in\mathcal{A}} Q^*(s',a')), \tag{4}$$
$$where \quad r \sim R(s,a), s' \sim P(\cdot|s,a).$$

在上面的表达中，$\overset{D}{=}$ 表示分布等价，而 Q^*(s', s') 是通过对 Z^*(s', a') 取期望来估计的。C51 [18] 通过使用对预先指定的锚点集合的分类分布来近似 Z^*(s, a)。而分布式 QR-DQN [17] 通过使用 K 个狄拉克 δ 函数的均匀混合来近似回报密度，即

$$Z_\theta(s,a) := \frac{1}{K}\sum_{i=1}^{K} \delta_{\theta_i(s,a)}, \quad Q_\theta(s,a) := \frac{1}{K}\sum_{i=1}^{K} \theta_i(s,a).$$

QR-DQN 在 Atari 2600 游戏中的表现优于 C51 和 DQN，获得了最先进的结果，尤其是在不利用 n 步更新 [26] 和优先回放 [27] 的代理之间。本文避免使用 n 步更新和优先回放，以使经验研究保持简单，并专注于深度 Q 学习算法。

2.2 离线强化学习

现代的异策略深度强化学习算法（如上所述）在常见基准测试中表现出色，例如 Atari 2600 游戏 [15] 和连续控制 MuJoCo 任务 [28]。这些离策略算法被认为是“在线”的，因为它们在优化策略和使用该策略收集更多数据之间交替进行。通常，这些算法在有限的回放缓冲区 [24] 中保留最近的经验，丢弃陈旧的数据以纳入最新的（在策略上的）经验。

与在线强化学习相反，离线强化学习描述了在使用固定数据集的完全离策略学习环境中的设置，而不需要与环境进行任何进一步的交互。我们主张使用离线强化学习来帮助分离强化学习算法利用经验和泛化的能力与其有效探索的能力。离线强化学习设置消除了与回放缓冲区和探索相关的设计选择；因此，与在线设置相比，它更容易进行实验和复现。

离线强化学习被认为具有挑战性，因为当前策略和离线数据收集策略之间存在分布不匹配，即，当正在学习的策略采取与数据收集策略不同的行动时，我们不知道应该提供什么奖励。本文重新审视离线强化学习，并探讨了仅使用离线数据训练的异策略深度强化学习代理是否可以在不进行分布不匹配修正的情况下取得成功。

3 本文方法

3.1 集成 DQN

Ensemble-DQN 是 DQN 的一个简单扩展，通过一组参数化的 Q 函数 [29] [30] [31] 来近似 Q 值。每个 Q 值估计，表示为 Q_θ^k(s,a)，都针对自己的目标 $Q_{\theta'}^k$(s,a) 进行训练，类似于 Bootstrapped-DQN [32]。这些 Q 函数使用相同顺序的相同小批量进行优化，从不同的参数初始化开始。损失函数 $\mathcal{L}(\theta)$ 的形式为：

$$\mathcal{L}(\theta) = \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{s,a,r,s'\sim\mathcal{D}}\left[l_\lambda(\Delta_\theta^k(s,a,r,s'))\right], \tag{5}$$
$$\Delta_\theta^k(s,a,r,s') = Q_\theta^k(s,a) - r - \gamma\max_{a'}Q_{\theta'}^k(s',a')$$

在上述公式中，l_λ 是 Huber 损失。尽管 Bootstrapped-DQN 在每个情节中使用一个 Q 值估计来改进探索，但在离线设置中，我们只关心 Ensemble-DQN 利用更好的能力，并使用 Q 值估计的均值进行评估。

3.2 随机集成混合 (REM)

增加用于集成的模型数量通常会提高监督学习模型的性能 [33]。这引发了一个问题，即是否可以以计算效率的方式使用指数数量的 Q 值估计进行集成。受到 dropout 的启发 [34]，我们提出了用于离策略强化学习的随机集成混合（REM）。

Random Ensemble Mixture (REM) 使用多个参数化的 Q 函数来估计 Q 值，类似于 Ensemble-DQN。REM 背后的关键见解是，可以将多个 Q 值估计的凸组合看作是一个 Q 值估计本身。这在固定点特别成立，即所有 Q 值估计都收敛到相同的 Q 函数。利用这一见解，我们训练了一个由混合概率定义的 Q 函数逼近器族，混合概率定义在 (K−1)-simplex 上。

具体而言，对于每个小批次，我们随机抽取一个分类分布 ，该分布定义了一个凸组合，用于逼近最优 Q 函数的 K 个估计。这个逼近器针对其相应的目标进行训练，以最小化时序差错。损失函数 $\mathcal{L}(\theta)$ 的形式为：

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'\sim D}\left[\mathbb{E}_{\alpha\sim P_\Delta}\left[l_\lambda\Delta_\theta^\alpha(s,a,r,s')\right]\right], \tag{6}$$
$$\Delta_\theta^\alpha = \sum_k \alpha_k Q_\theta^k(s,a) - r - \gamma\max_{a'}\sum_k \alpha_k Q_{\theta'}^k(s',a')$$

在上述公式中，P_Δ 代表标准 K−1 单纯形 $\Delta_{K-1} = \{\alpha\in\mathbb{R}^K : \alpha_1+\alpha_2+...+\alpha_K = 1, \alpha_k \geq 0, k = 1,...,K\}$ 上的概率分布。

REM 将 Q 学习看作基于 Bellman 最优性约束（2）的约束满足问题，而 $\mathcal{L}(\theta)$ 可以被视为与不同混合概率分布相对应的无穷多约束的集合。对于动作选择，REM 使用 K 个值估计的平均值作为 Q 函数，即 $Q(s,a) = \frac{1}{K}\sum_k Q_k^\theta(s,a)$。REM 易于实现和分析（见命题 1），可以看作是用于基于值的强化学习的简单正则化技术。在我们的实验中，我们使用一个非常简单的分布 P^Δ: 我们首先从均匀分布 (Uniform) 中独立同分布 (i.i.d.) 抽取一组 K 个值，范围在 U(0,1)，然后对它们进行归一化，得到一个有效的分类分布，即 $\alpha_k' \sim U(0,1)$，然后 $\alpha_k = \frac{\alpha_k'}{\sum\alpha_i'}$。

命题 1. 考虑以下假设：(a) 分布 P_Δ 在整个 K−1 单纯形上具有完全支持。(b) 仅有有限数量的不同 Q 函数在全局范围内最小化了式 (3) 中的损失。(c) Q^* 是根据数据分布 $\mathcal{D}$ 引起

的 MDP（马尔可夫决策过程）的术语定义的。(d)Q^* 位于我们函数逼近的族中。那么，在多头 Q 网络的 $\mathcal{L}(\theta)$（式 (7)）全局最小值处：

(i) 在假设 (a) 和 (b) 下，所有的 Q-heads 表示相同的 Q 函数。

(ii) 在假设 (a)–(d) 下，共同的全局解是 Q^*。

(ii) 的证明来自 (i) 以及（式 (7)）被 Q^* 达到的 TD 误差的下界这一事实。(i) 部分的证明可以在附录中找到。

3.3 损失函数定义

4 复现细节

4.1 DQN 重播数据集（记录的 DQN 数据）

DQN 重播数据集的收集方式如下：我们首先在所有 60 款 Atari 2600 游戏中训练一个 DQN 代理，并启用了 200 亿帧（标准协议）的粘性操作，并保存了所有经验元组（观察、行动、奖励、下一次观察）（约 50 万）在训练期间遇到。

此记录的 DQN 数据可以在公共 GCP 存储桶中找到，该存储桶可以使用 gsutil 下载。安装 gsutil，按照 gs://atari-replay-datasets 的说明进行操作。

安装 gsutil 后，运行以下命令以复制 Arari 2600 的 Pong 数据集：

```
gsutil -m cp -R gs://atari-replay-datasets/dqn/Pong ./
```

4.2 依赖软件包

该代码在 MAC 下进行了测试，并使用以下软件包：

- tensorflow-gpu>=1.13
- absl-py
- atari-py
- gin-config
- opencv-python
- gym
- numpy

4.3 配置

安装下面的依赖项:

```
brew install cmake zlib
pip install absl-py atari-py gin-config
gym opencv-python tensorflow
```

接着安装 dopamine:

```
pip install git+https://github.com/google/dopamine.git
```

最后下载批量的 RL 源代码:

```
git clone https://github.com/google-research/batch_rl.git
```

4.4 在 DQN 数据上训练批处理代理

标准 Atari 2600 实验的入口是 batch_rl/fixed_replay/train.py。使用以下命令运行批处理代理:

```
python -um batch_rl.fixed_replay.train \
--base_dir=/tmp/batch_rl \
--replay_dir=$DATA_DIR/Pong/1 \
--gin_files='batch_rl/fixed_replay/configs/dqn.gin'
```

5 实验结果分析

通过在 60 个 Atari 2600 游戏上分别对几个实例的 DQN (Nature) 代理进行培训 [16], 创建了 DQN Replay 数据集, 每个游戏培训 200 百万帧, 帧跳过 4 (标准协议) 并采用粘性动作 [35] (以 25% 的概率, 代理的先前动作被执行而不是当前动作)。在每个游戏中, 我们使用随机初始化训练 5 个不同的代理, 并将训练过程中遇到的所有 (观察, 动作, 奖励, 下一个观察) 元组存储到每个游戏的 5 个重放数据集中, 总共产生了 300 个数据集。

每个游戏的重放数据集约为 ImageNet [36] 的 3.5 倍大, 并包含在线 DQN 代理优化过程中看到的所有中间 (多样的) 策略的样本。

实验设置。DQN Replay 数据集用于在训练过程中与环境没有任何交互的情况下, 对 RL 代理进行离线训练。尽管游戏重放数据集包含了 DQN 代理随着训练的进行而不断改善的数据, 我们将离线代理的性能与训练后获得的最佳性能的代理进行比较 (即完全训练的 DQN)。对离线代理的评估是在线进行的, 以每 100 万个训练帧的间隔进行有限次数。对于每个游戏, 我们评估 5 个离线代理 (每个数据集一个), 使用在线回报报告 5 个代理的平均最佳性能。

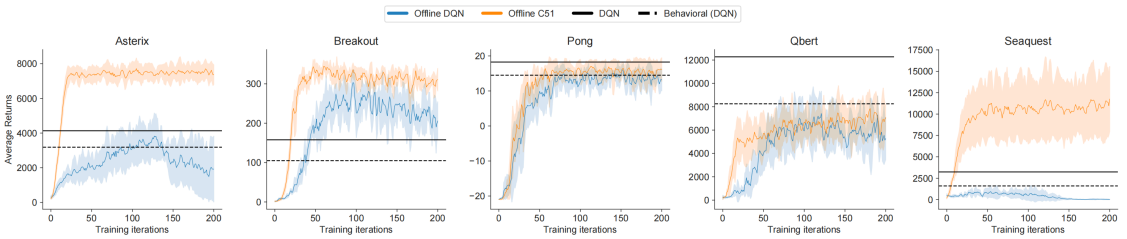

图 1. DQN 重播数据集上的离线代理

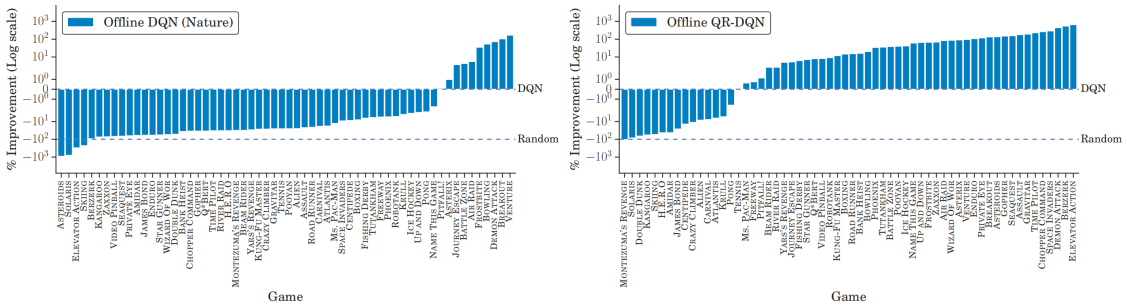

图 2. 离线 QR-DQN 与 DQN（自然）

5.1 无环境交互的标准离策略强化学习算法能否成功？

在给定 DQN 代理的记录回放数据的情况下，自然而然地会问，仅使用这个数据集进行训练的离线版本的 DQN 会表现得如何？此外，比起离线 DQN，更近期的离线策略算法是否能够更有效地利用 DQN Replay 数据集？为了调查这些问题，我们在 DQN 回放数据集上离线训练了 DQN（Nature）和 QR-DQN 代理，训练步数与在线 DQN 相同的梯度更新次数。

图 2 显示，离线 DQN 在所有游戏上的表现都不如完全训练的在线 DQN，除了在少数几个游戏中它的得分比在线 DQN 高得多，且使用相同数量的数据和梯度更新。另一方面，离线 QR-DQN 在大多数游戏上表现优于离线 DQN 和在线 DQN。使用 DQN Replay 数据集进行训练的离线 C51 在很大程度上也改进了离线 DQN 的性能（图 1）。离线 QR-DQN 的性能优于离线 C51。

Offline agent	Median	>DQN
DQN(Nature)	83.4%	17
DQN(Adam)	111.9%	41
Ensemble-DQN	111.0%	39
Averaged Ensemble-DQN	112.1%	43
QR-DQN	118.9%	45
REM	123.8%	49

表 1. 离线智能体的渐进性能

这些结果表明，通过在 DQN Replay 数据集上使用标准的深度强化学习算法，可以在离线环境中优化强大的 Atari 代理，而不需要限制学到的策略紧密匹配离线轨迹的训练数据集。

此外，离线 QR-DQN/C51 与 DQN（Nature）性能之间的差异表明它们在利用离线数据方面的能力不同。

5.2 离线 RL 代理的渐近性能

在监督学习中，渐近性能比在梯度更新的固定预算内的性能更为重要。同样，在给定样本复杂度的情况下，我们更倾向于那些只要梯度更新的数量可行就能表现最佳的强化学习算法。由于离线数据集的样本效率是固定的，我们选择对离线代理进行 5 倍于 DQN 的梯度更新的训练。

与 QR-DQN 的比较。QR-DQN 修改了 DQN（Nature）的架构，使用多头 Q 网络为每个动作输出 K 个值，并用 Adam [37] 替换了 RMSProp [38] 进行优化。为了与 QR-DQN 进行公平比较，我们使用了与 QR-DQN 相同的多头 Q 网络，其中 $K = 200$，每个头代表对 REM 和 Ensemble-DQN 的 Q 值估计。我们同样使用 Adam 进行优化。

结果。表 1 显示了使用 REM 和 Ensemble-DQN 的基线的比较。令人惊讶的是，使用 Adam 的 DQN 填补了 QR-DQN 和 DQN（Nature）在离线环境中渐近性能之间的差距。离线 Ensemble-DQN 并没有改进这个强大的 DQN 基线，表明其天真的集成方法是不足够的。此外，Averaged Ensemble-DQN 的性能仅略优于 Ensemble-DQN。

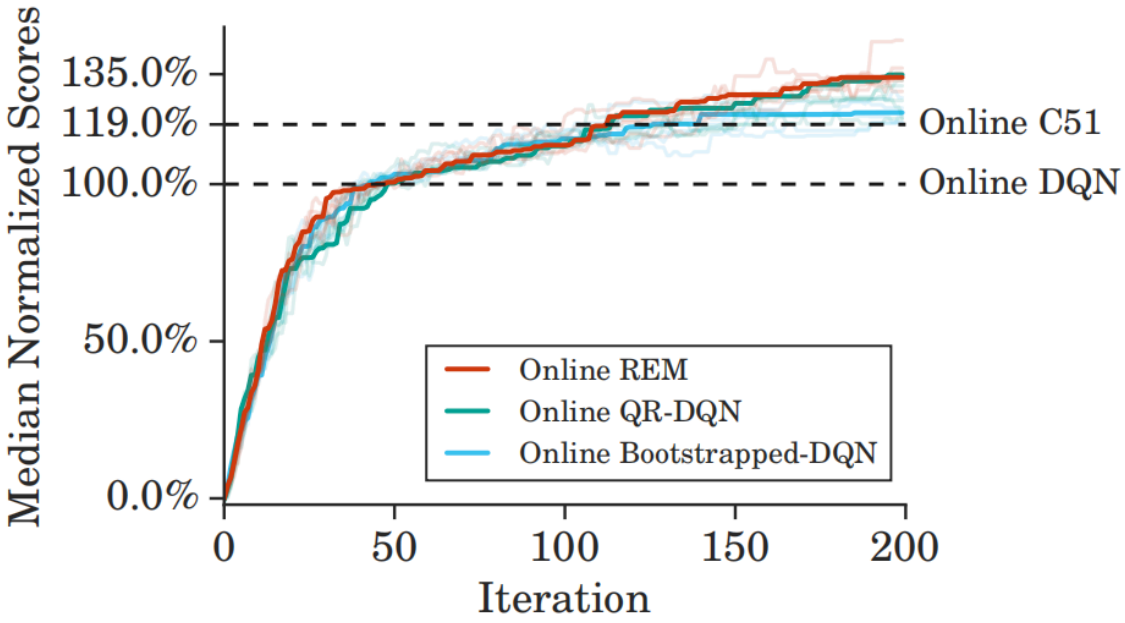

图 3. 在线 REM 与基线

5.3 REM 在在线环境中有效吗？

在在线强化学习中，学习和数据生成紧密耦合，即学得更快的代理也会收集更多相关的数据。我们在在线环境中运行了具有 4 个独立 Q 网络的在线 REM，因为在离线环境中，它比多头 REM 具有更好的收敛速度。对于数据收集，我们使用 epsilon-greedy 策略，每一集都从单纯形上随机采样一个 Q 估计，类似于 Bootstrapped DQN。我们遵循 Atari 上的标准在线 RL 协议，并使用一个固定的重放缓冲区，包含 100 万帧。

为了评估 REM 目标（式 6）在在线环境中的收益，我们还评估了具有相同修改（例如，独立 Q 网络）的 Bootstrapped-DQN，类似于在线 REM。图 3 显示，REM 的性能与 QR-DQN 相当，并且明显优于 Bootstrapped-DQN。这表明我们可以利用在离线环境中获得的见解，通过适当的设计选择（例如，探索、回放缓冲区），创建有效的在线方法。

6 总结与展望

这篇论文研究了基于一个 DQN 代理的离线强化学习（RL）在 Atari 2600 游戏上的应用，基于该代理的记录经验。论文表明，标准的 RL 方法可以从 DQN Replay 数据集中学习玩 Atari 游戏，表现优于数据集中最佳行为。这与现有的研究形成对比，后者声称标准方法在离线设置中失败。DQN Replay 数据集可以作为离线 RL 的基准。这些结果呈现了一个积极的观点，即可以开发出强大的 RL 算法，能够从大规模的离线数据集中有效地学习。REM 通过展示即使是简单的集成方法在离线环境中也能够有效，加强了这种乐观的观点。总体而言，该论文表明离线 RL 有潜力创建一个基于数据的 RL 范式，其中可以在进一步通过探索收集新数据之前，使用大量现有的多样数据集对 RL 代理进行预训练，从而创建能够在现实世界中部署并持续学习的样本高效代理。

参考文献

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[2] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[3] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

[4] Serkan Cabi, Sergio Gómez Colmenarejo, Alexander Novikov, Ksenia Konyushkova, Scott Reed, Rae Jeong, Konrad Zolna, Yusuf Aytar, David Budden, Mel Vecerik, et al. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv preprint arXiv:1909.12200*, 2019.

[5] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[6] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, Trevor Darrell, et al. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.

[7] Alex Strehl, John Langford, Lihong Li, and Sham M Kakade. Learning from logged implicit exploration data. *Advances in neural information processing systems*, 23, 2010.

[8] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(11), 2013.

[9] Susan M Shortreed, Eric Laber, Daniel J Lizotte, T Scott Stroup, Joelle Pineau, and Susan A Murphy. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine learning*, 84:109–136, 2011.

[10] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[11] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[12] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[13] Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

[14] Shangtong Zhang and Richard S Sutton. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*, 2017.

[15] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[17] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[18] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pages 449–458. PMLR, 2017.

[19] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[20] RJNJ Bellman. Dynamic programming princeton university press princeton. *New Jersey Google Scholar*, pages 24–73, 1957.

[21] Christopher JC Watkins. H.; dayan, p. q-learning. *Machine learning*, 8(3):279–292, 1992.

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[24] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8:293–321, 1992.

[25] Stratton C Jaquette. Markov decision processes with a new optimality criterion: Discrete time. *The Annals of Statistics*, 1(3):496–505, 1973.

[26] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.

[27] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[28] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[29] Stefan Faußer and Friedhelm Schwenker. Neural network ensembles in reinforcement learning. *Neural Processing Letters*, 41:55–69, 2015.

[30] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[31] Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *International conference on machine learning*, pages 176–185. PMLR, 2017.

[32] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[33] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[35] Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

[36] Jia Deng. A large-scale hierarchical image database. *Proc. of IEEE Computer Vision and Pattern Recognition, 2009*, 2009.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *COURSERA Neural Networks Mach. Learn*, 17, 2012.