深度强化学习与双重 Q 学习

Hado van Hasselt、Arthur Guez 和 David Silver，Google DeepMind

摘要

流行的 Q-learning 算法在某些条件下已知会高估动作价值。此前并不清楚在实践中这种高估是否常见、是否会损害性能，以及是否能够普遍加以避免。本文对这三个问题都给出了肯定的回答。具体而言，我们首先表明，近期的 DQN 算法（将 Q-learning 与深度神经网络结合）在 Atari 2600 领域的某些游戏中存在显著的高估现象。随后我们表明，最初在表格型设定中提出的 Double Q-learning 算法背后的思想，可以推广到适用于大规模函数逼近。我们提出了对 DQN 算法的一种具体改造，并展示所得算法不仅如假设那样减少了观察到的高估，而且这也在多款游戏上带来了显著更好的性能。

强化学习（Sutton 和 Barto，1998）的目标是通过优化累积的未来奖励信号，为序列决策问题学习良好的策略。Q-learning（Watkins，1989）是最流行的强化学习算法之一，但众所周知，它有时会学习到不切实际地高的动作价值，因为它包含一个在估计的动作价值上进行最大化的步骤，而这往往会偏向于被高估的值而非被低估的值。

在以往的研究中，过估计被归因于不够灵活的函数近似（Thrun and Schwartz，1993）以及噪声（van Hasselt，2010，2011）。本文统一了这些观点，并表明当动作价值不准确时就可能出现过估计，而不论近似误差的来源是什么。当然，在学习过程中，不精确的价值估计是常态，这表明过估计可能比先前所认识到的要普遍得多。

在实践中，如果确实发生了过估计，这是否会对性能产生负面影响仍是一个开放性问题。过于乐观的价值估计本身并不一定就是问题。若所有价值都一致地更高，那么相对的动作偏好将得以保持，我们也不会预期由此产生的策略会更差。此外，众所周知，有时保持乐观是有益的：面对不确定性的乐观是一种众所周知的探索技术（Kaelbling 等，1996）。然而，如果这些高估并非均匀分布，且未集中在我们希望进一步学习的状态上，那么它们可能会对所得策略的质量产生负面影响。Thrun 和 Schwartz（1993）给出了具体示例，表明这会导致次优策略，即使在渐近意义下亦然。

为了检验在实践中以及在大规模情况下是否会出现过高估计，我们研究了近期的 DQN 算法（Mnih 等，2015）的表现。DQN 将 Q-learning 与灵活的深度神经网络相结合，并在一组多样且规模很大的确定性 Atari 2600 游戏上进行了测试，在许多游戏上达到了人类水平的表现。从某些方面看，这一设定是 Q-learning 的最佳情形，因为深度神经网络提供了灵活的函数近似，具有较低渐近近似误差的潜力，而环境的确定性也避免了噪声带来的有害影响。令人惊讶的是，我们表明，即使在这种相对有利的设定下，DQN 有时也会显著高估动作的价值。

我们表明，Double Q-learning 算法（van Hasselt, 2010）背后的思想最初是在表格化设定中提出的，但可以推广到任意函数近似方法，包括深度神经网络。我们利用这一点构建了一种新算法，称之为 Double DQN。随后我们展示，该算法不仅能给出更准确的价值估计，而且在多款游戏上取得了显著更高的得分。这表明，DQN 的过度估计确实会导致更差的策略，而降低这种过度估计是有益的。此外，通过对 DQN 的改进，我们在 Atari 领域获得了最先进的结果。

背景

为了解决序贯决策问题，我们可以学习对每个动作最优价值的估计，其定义为：采取该动作并在此后遵循最优策略时，未来奖励的期望累积和。在给定策略 π 下，状态 s 中动作 a 的真实价值为

$$Q_\pi(s,a) \equiv \mathbb{E}\left[R_1 + \gamma R_2 + \ldots \mid S_0 = s, A_0 = a, \pi\right],$$

其中 $\gamma \in [0,1]$ 是一个折扣因子，用于权衡即时奖励与后续奖励的重要性。则最优值为 $Q_*(s,a) = \max_\pi Q_\pi(s,a)$。通过在每个状态中选择价值最高的动作，可以很容易地从最优值导出最优策略。

最优动作价值的估计可以使用 Q-learning（Watkins，1989）来学习，这是一种时序差分学习（Sutton，1988）。大多数有趣的问题规模过大，无法在所有状态中分别学习所有动作价值。相反，我们可以学习一个参数化的价值函数 $Q(s,a;\boldsymbol{\theta}_t)$。在状态 S_t 中采取动作 A_t，并观察到即时奖励 R_{t+1} 以及得到的后继状态 S_{t+1} 之后，用于更新参数的标准 Q-learning 更新规则为

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha(Y_t^{\mathrm{Q}} - Q(S_t, A_t; \boldsymbol{\theta}_t))\nabla_{\boldsymbol{\theta}_t} Q(S_t, A_t; \boldsymbol{\theta}_t)\,. \quad (1)$$

其中 α 是标量步长，目标 Y_t^{Q} 被定义为

$$Y_t^{\mathrm{Q}} \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t)\,. \quad (2)$$

此更新类似于随机梯度下降，将当前值 $Q(S_t, A_t;\ \boldsymbol{\theta}_t)$ 朝着目标值 Y_t^{Q} 进行更新。

深度 Q 网络

深度 Q 网络（DQN）是一个多层神经网络，对于给定的状态 s，输出一个动作价值向量 $Q(s,\cdot;\ \boldsymbol{\theta})$，其中 $\boldsymbol{\theta}$ 是网络的参数。对于一个 n 维的状态空间以及一个包含 m 个动作的动作空间，该神经网络是一个从 $\mathbb{R}^n$ 到 $\mathbb{R}^m$ 的函数。Mnih 等人（2015）提出的 DQN 算法有两个重要组成部分：使用目标网络，以及使用经验回放。目标网络的参数为 $\boldsymbol{\theta}^-$，与在线网络相同，不同之处在于它的参数每隔 τ 步从在线网络复制一次，因此此时 $\boldsymbol{\theta}_t^- = \boldsymbol{\theta}_t$，并在其他所有步骤保持固定。DQN 使用的目标为：

$$Y_t^{\mathrm{DQN}} \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t^-)\,. \quad (3)$$

对于经验回放（Lin，1992），观测到的转移会被存储一段时间，并从该记忆库中以均匀方式采样以更新网络。目标网络和经验回放都显著提升了该算法的性能（Mnih 等，2015）。

双重 Q 学习

标准 Q-learning 和 DQN 中（2）和（3）里的 max 算子，使用相同的值既用于选择动作又用于评估动作。这会使得更可能选择被高估的值，从而导致过于乐观的价值估计。为防止这种情况，我们可以将选择与评估解耦。这就是 Double Q-learning（van Hasselt，2010）背后的思想。

在原始的 Double Q-learning 算法中，通过将每条经验随机分配给两个价值函数之一进行更新来学习两个价值函数，因此存在两组权重，$\boldsymbol{\theta}$ 和 $\boldsymbol{\theta}'$。对于每次更新，一组权重用于确定贪婪策略，另一组用于确定其价值。为便于清晰比较，我们可以先将 Q-learning 中的选择与评估解耦，并将其目标式 (2) 重写为

$$Y_t^{\mathrm{Q}} = R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t)\,.$$

Double Q-learning 误差可以写为

$$Y_t^{\mathrm{DoubleQ}} \equiv R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}'_t)\,. \quad (4)$$

注意，在 argmax 中对动作的选择仍然取决于在线权重 $\boldsymbol{\theta}_t$。这意味着，与 Q-learning 一样，我们仍在根据当前的价值来估计贪婪策略的价值，如 $\boldsymbol{\theta}_t$ 所定义的那样。然而，我们使用第二组权重 $\boldsymbol{\theta}'_t$ 来公平地评估该策略的价值。第二组权重可以通过对称地更新，即交换 $\boldsymbol{\theta}$ 和 $\boldsymbol{\theta}'$ 的角色来进行更新。

由于估计误差导致的过度乐观

Q-learning 的高估最早由 Thrun 和 Schwartz（1993）研究，他们表明，如果动作价值包含在区间 $[-\epsilon, \epsilon]$ 内均匀分布的随机误差，那么每个目标最多会被高估 $\gamma\epsilon\frac{m-1}{m+1}$，其中 m 是动作的数量。此外，Thrun 和 Schwartz 给出了一个具体示例，其中这些高估甚至在渐近意义上会导致次优策略，并展示了在使用函数逼近时，这些高估会在一个小型玩具问题中表现出来。随后 van Hasselt（2010）指出，即使使用表格表示，环境中的噪声也可能导致高估，并提出 Double Q-learning 作为解决方案。

在本节中，我们更一般地说明：任何形式的估计误差都可能导致向上的偏差，无论这些误差是由环境噪声、函数逼近、非平稳性还是任何其他来源引起的。这一点很重要，因为在实践中，任何方法在学习过程中都会产生一些不准确性，仅仅因为真实值在初始阶段是未知的。

Thrun 和 Schwartz（1993）在上文引用的结果为特定设置下的高估给出了一个上界，但也有可能——且可能更有趣——推导出一个下界。

定理 1。*Consider a state s in which all the true optimal action values are equal at $Q_*(s,a) = V_*(s)$ for some $V_*(s)$. Let Q_t be arbitrary value estimates that are on the whole unbiased in the sense that $\sum_a (Q_t(s,a) - V_*(s)) = 0$, but that are not all correct, such that $\frac{1}{m}\sum_a (Q_t(s,a) - V_*(s))^2 = C$ for some $C > 0$, where $m \geq 2$ is the number of actions in s. Under these conditions, $\max_a Q_t(s,a) \geq V_*(s) + \sqrt{\frac{C}{m-1}}$. This lower bound is tight. Under the same conditions, the lower bound on the absolute error of the Double Q-learning estimate is zero. (Proof in appendix.)*

请注意，我们并不需要假设不同动作的估计误差彼此独立。该定理表明，即使价值估计在平均意义上是正确的，来自任何来源的估计误差也可能推动这些估计值上升，并偏离真实的最优值。

定理 1 中的下界会随着动作数量的增加而降低。这是考虑下界所产生的假象，因为下界要求达到非常特定的取值。更典型地，如图 1 所示，过度乐观会随着动作数量的增加而增强。Q-learning 在那里产生的过估计确实会随着动作数量的增加而增加，

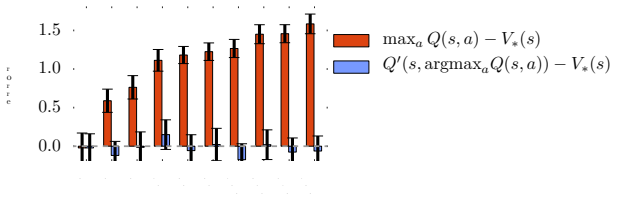

图 1：当动作价值为 $Q(s,a) = V_*(s) + \epsilon_a$ 且误差 $\{\epsilon_a\}_{a=1}^m$ 为相互独立的标准正态随机变量时，橙色柱状条显示了单次 Q-learning 更新中的偏差。用于蓝色柱状条的第二组动作价值 Q' 以相同方式独立生成。所有柱状条均为 100 次重复的平均值。

尽管 Double Q-learning 是无偏的。再举一例，如果对所有动作 $Q_*(s,a) = V_*(s)$ 的估计误差 $Q_t(s,a) - V_*(s)$ 在 $[-1, 1]$ 上均匀随机，则过度乐观为 $\frac{m-1}{m+1}$。（证明见附录。）

我们现在转向函数逼近，并考虑一个实值的连续状态空间，其中每个状态下有 10 个离散动作。为简单起见，本例中的真实最优动作价值只依赖于状态，因此在每个状态下所有动作具有相同的真实价值。这些真实价值在图 2 左列的图中（紫色线）给出，并被定义为 $Q_*(s,a) = \sin(s)$（顶行）或 $Q_*(s,a) = 2\exp(-s^2)$（中行和底行）。左侧图还展示了单个动作的一个近似（绿色线）作为状态的函数，以及该估计所依据的样本（绿色点）。该估计是一个 d 次多项式，通过在采样状态处拟合真实值获得，其中 $d = 6$（顶行和中行）或 $d = 9$（底行）。样本与真实函数完全一致：没有噪声，并且我们假设在这些采样状态上我们拥有动作价值的真实值。对于前两行，即使在采样状态上该近似也是不精确的，因为函数逼近器的灵活性不足。在底行中，函数足够灵活以拟合绿色点，但这会降低在未采样状态上的准确性。注意，在左侧图的左边附近，采样状态之间的间距更大，从而导致更大的估计误差。从许多方面来看，这都是一种典型的学习设置：在每个时间点，我们只有有限的数据。

图 2 中间一列的图展示了所有 10 个动作的估计动作价值函数（绿色线），作为状态的函数，同时还给出了每个状态下的最大动作价值（黑色虚线）。尽管真实价值函数对所有动作都相同，但由于我们提供了不同的采样状态集合，这些近似结果有所不同。[1] 最大值往往高于左侧以紫色显示的真实值。这一点在右侧的图中得到证实，右侧图以橙色显示了黑色曲线与紫色曲线之间的差异。橙色线几乎总是为正，表明存在向上的偏差。右侧图还以蓝色显示了 Double Q-learning 的估计[2]，其平均值更接近于零得多。这表明 Double Q-learning 确实能够成功降低 Q-learning 的过度乐观。

图2中的不同行展示了同一实验的不同变体。顶行与中行之间的差异在于真实价值函数，这表明高估并非某个特定真实价值函数所导致的伪象。中行与底行之间的差异在于函数近似的灵活性。在左中图中，由于函数不够灵活，即使对某些采样到的状态，估计也是不正确的。底左图中的函数更灵活，但这会导致对未见状态的估计误差更高，从而产生更高的高估。这一点很重要，因为灵活的参数化函数近似器常常被用于强化学习（例如，Tesauro 1995；Sallans and Hinton 2004；Riedmiller 2005；Mnih 等 2015）。

与 van Hasselt（2010）不同，我们没有使用统计论证来发现高估；获得图 2 的过程完全是确定性的。与 Thrun 和 Schwartz（1993）不同，我们没有依赖具有不可约渐近误差的僵硬函数近似；底部一行表明，足够灵活以覆盖所有样本的函数会导致很高的高估。这表明，高估现象可能相当普遍地发生。

在上面的示例中，即使假设我们在某些状态下拥有 *true* 动作价值的样本，仍然会出现高估。如果我们从已经过度乐观的动作价值中进行自举，价值估计还会进一步恶化，因为这会使高估在我们的估计中不断传播。尽管 *uniformly* 高估价值可能不会损害最终得到的策略，但在实践中，不同状态和动作的高估误差会有所不同。高估与自举相结合，会产生一种有害的效应：传播关于哪些状态比其他状态更有价值的错误相对信息，从而直接影响所学习策略的质量。

这些过估计不应与面对不确定性时的乐观主义（Sutton, 1990; Agrawal, 1995; Kaelbling et al., 1996; Auer et al., 2002; Brafman and Tennenholtz, 2003; Szita and Lőrincz, 2008; Strehl et al., 2009）相混淆；在后者中，会给价值不确定的状态或动作一个探索奖励。相反，这里讨论的过估计只会在更新之后发生，从而导致在表面确定性的情况下产生过度乐观。这一点已被 Thrun 和 Schwartz（1993）观察到，他们指出，与面对不确定性时的乐观主义不同，这些过估计实际上会阻碍学习到最优策略。我们还将在后续实验中看到这一对策略质量的负面影响得到证实：当我们使用 Double Q-learning 减少过估计时，策略会得到改善。

[1] Each action-value function is fit with a different subset of integer states. States -6 and 6 are always included to avoid extrapolations, and for each action two adjacent integers are missing: for action a_1 states -5 and -4 are not sampled, for a_2 states -4 and -3 are not sampled, and so on. This causes the estimated values to differ.

[2] We arbitrarily used the samples of action a_{i+5} (for $i \leq 5$) or a_{i-5} (for $i > 5$) as the second set of samples for the double estimator of action a_i.

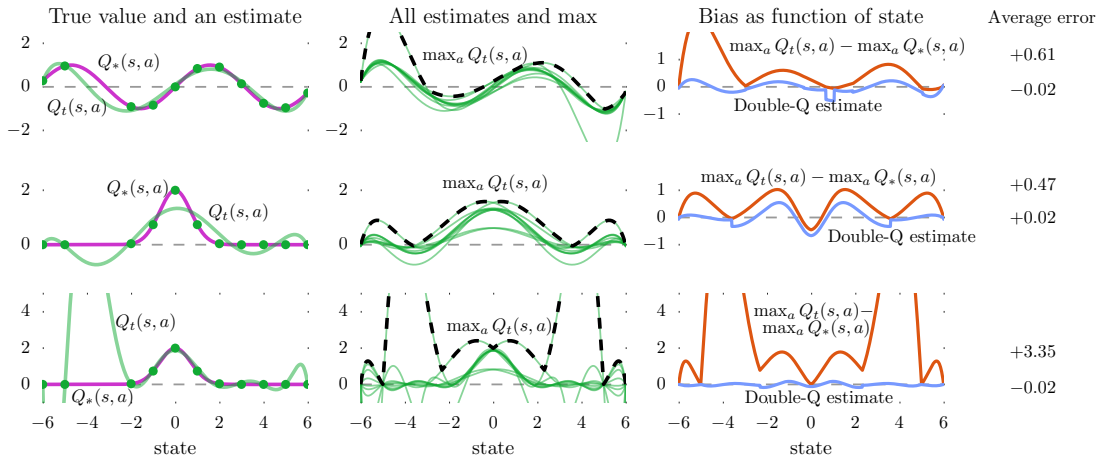

图2：学习过程中高估现象的示意图。在每个状态（x轴）中，有10个动作。左列显示真实值$V_*(s)$ (紫色曲线)。所有真实动作值由$Q_*(s,a) = V_*(s)$定义。绿色曲线显示某个动作的估计值$Q(s,a)$随状态变化的函数，并在若干采样状态（绿色点）处拟合真实值。中列图绘出所有估计值（绿色）以及这些值的最大值（黑色虚线）。该最大值几乎在所有位置都高于真实值（左图中的紫色曲线）。右列图用橙色显示差值。右列图中的蓝色曲线是Double Q-learning使用每个状态的第二组样本得到的估计。蓝色曲线更接近零，表明偏差更小。三行分别对应不同的真实函数（左图，紫色）或拟合函数的容量（左图，绿色）。（细节见正文）

双重 DQN

Double Q-learning 的核心思想是通过将目标中的 max 操作分解为动作选择与动作评估，从而减少过高估计。尽管这种分解并未做到完全解耦，DQN 架构中的目标网络为第二个价值函数提供了一个天然的候选者，而无需引入额外的网络。因此，我们提出：按照在线网络来评估贪心策略，但使用目标网络来估计其价值。结合 Double Q-learning 与 DQN，我们将所得算法称为 Double DQN。其更新方式与 DQN 相同，但将目标 Y_t^{DQN} 替换为

$$Y_t^{\text{DoubleDQN}} \equiv R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t), \boldsymbol{\theta}_t^-) .$$

与 Double Q-learning (4) 相比，在评估当前贪婪策略时，第二个网络 $\boldsymbol{\theta}_t'$ 的权重被替换为目标网络 $\boldsymbol{\theta}_t^-$ 的权重。对目标网络的更新方式与 DQN 保持不变，仍然是周期性地从在线网络复制。

这个版本的 Double DQN 也许是在朝向 Double Q-learning 的方向上对 DQN 所做的最小可能改动。目标是在保持 DQN 算法其余部分不变、以便进行公平比较的同时，获得 Double Q-learning 的大部分收益，并且将额外的计算开销降到最低。

经验结果

在本节中，我们分析 DQN 的高估现象，并表明 Double DQN 相比 DQN 在价值精度和策略质量两方面都有所改进。为进一步检验该方法的鲁棒性，我们还按照 Nair 等人（2015）的建议，使用由专家人类轨迹生成的随机起始状态对这些算法进行额外评估。

我们的测试平台由 Atari 2600 游戏组成，使用 Arcade Learning Environment（Bellemare 等，2013）。该目标是让单一算法在一组固定的超参数下，仅以屏幕像素作为输入，通过交互分别学习玩每一款游戏。这是一个要求很高的测试平台：不仅输入是高维的，游戏画面和游戏机制在不同游戏之间也有很大差异。因此，好的解决方案必须在很大程度上依赖学习算法——仅靠调参来对领域进行过拟合在实践中并不可行。

我们严格遵循 Mnih 等人（2015）所概述的实验设置和网络架构。简而言之，网络架构为卷积神经网络（Fukushima，1988；LeCun 等人，1998），包含 3 个卷积层和一个全连接隐藏层（总计约 150 万个参数）。该网络以最近四帧作为输入，并输出每个动作的动作价值。在每个游戏上，网络在单个 GPU 上训练 2 亿帧，约为 1 周。

过度乐观性的结果

图 3 展示了 DQN 在六款 Atari 游戏中过度估计的示例。DQN 和 Double DQN 都是在 Mnih 等（2015）所描述的完全相同条件下训练的。DQN 对当前贪婪策略的价值始终表现出一致且有时极其夸张的过度乐观，这一点可以通过将顶行图中的橙色学习曲线与橙色直线进行比较看出；后者表示所学习到的最佳策略的实际折扣价值。更精确地说，（平均的）价值估计是在训练过程中定期计算的，计算时会进行长度为 $T = 125,000$ 步的完整评估阶段，如下所示

$$\frac{1}{T}\sum_{t=1}^{T} \operatorname*{argmax}_a Q(S_t, a; \boldsymbol{\theta}) .$$

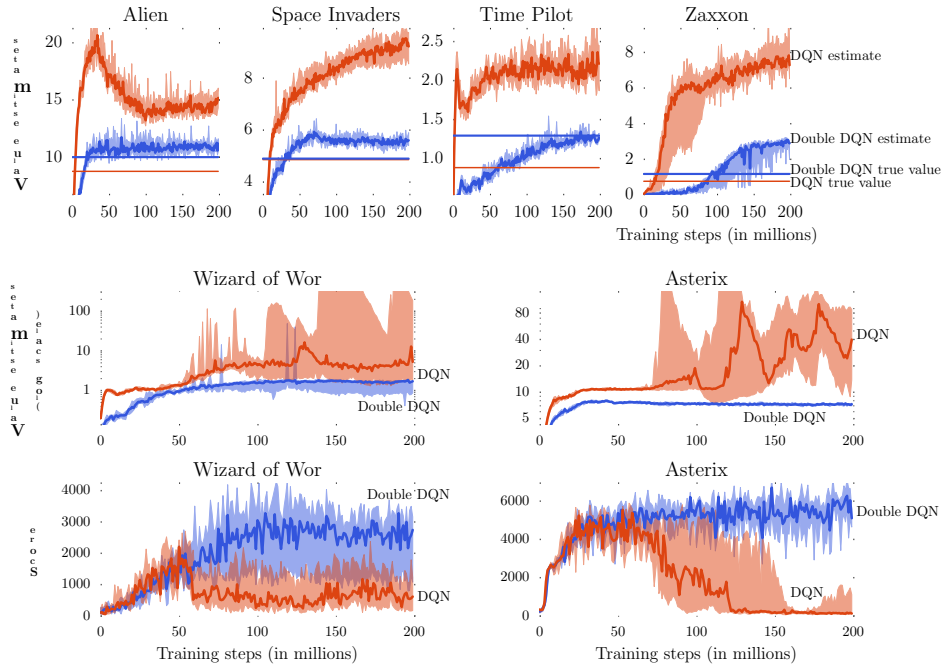

图 3：顶部和中间两行展示了 DQN（橙色）和 Double DQN（蓝色）在六款 Atari 游戏上的价值估计。结果通过在 Mnih 等人（2015）所采用的超参数设置下，使用 6 个不同的随机种子运行 DQN 与 Double DQN 获得。较深的线表示跨种子的中位数，我们对两个极端值取平均以得到阴影区域（即采用线性插值得到的 10% 与 90% 分位数）。顶部一行中水平的橙色（对应 DQN）与蓝色（对应 Double DQN）直线，是在学习结束后运行相应智能体，并对其从每个访问过的状态获得的实际折扣回报取平均来计算的。若不存在偏差，这些直线应当与图中右侧的学习曲线相吻合。中间一行展示了两款游戏中的价值估计（对数刻度），在这两款游戏里 DQN 的过度乐观相当极端。底部一行展示了这种现象对训练期间评估得到的智能体得分的负面影响：当过高估计开始出现时，得分会下降。使用 Double DQN 学习要稳定得多。

地面真实平均值是通过运行学习到的最佳策略若干个回合，并计算实际的累积回报得到的。若不存在过高估计，我们会期望这些量彼此一致（即每个图右侧的曲线与直线重合）。然而，DQN 的学习曲线最终往往显著高于真实值。以蓝色显示的 Double DQN 学习曲线则更接近表示最终策略真实价值的蓝色直线。注意，蓝色直线通常高于橙色直线。这表明 Double DQN 不仅能产生更准确的价值估计，还能得到更好的策略。

更极端的高估出现在中间两幅图中，其中 DQN 在游戏 Asterix 和 Wizard of Wor 上非常不稳定。注意 y 轴上的数值采用对数刻度。底部两幅图展示了这两款游戏对应的得分。注意，中间图中 DQN 的价值估计上升与底部图中得分下降同时发生。再次说明，这些高估正在损害最终策略的质量。若孤立地看这些结果，人们或许会倾向于认为，观察到的不稳定性与使用函数近似进行离策略学习所固有的不稳定性问题有关（Baird, 1995; Tsitsiklis and Van Roy, 1997; Sutton et al., 2008; Maei, 2011; Sutton et al., 2015）。然而，我们看到使用 Double DQN 时学习要稳定得多，

	DQN	Double DQN
Median	93.5%	114.7%
Mean	241.1%	330.3%

表 1：在 49 款游戏中，游玩时长最多 5 分钟时的归一化性能汇总。DQN 的结果来自 Mnih 等（2015）

这表明，这些不稳定性的原因实际上是 Q-learning 的过度乐观。图 3 只展示了少数几个例子，但在所测试的全部 49 款 Atari 游戏中都观察到了 DQN 的高估现象，只是程度各不相同。

学习到的策略质量

过度乐观并不总是会对学习到的策略质量产生不利影响。例如，尽管对策略价值略有高估，DQN 仍能在 Pong 中实现最优行为。尽管如此，减少高估可以显著提升学习的稳定性；我们在图 3 中看到了清晰的例子。我们现在通过在 DQN 所测试的全部 49 款游戏上进行评估，更一般性地考察 Double DQN 在策略质量方面的帮助程度。

如 Mnih 等人（2015）所述，每个评估回合开始时会执行一种特殊的空操作（no-op）动作，该动作不会影响环境，最多重复 30 次，以便为智能体提供不同的起始点。评估期间进行一定程度的探索可提供额外的随机化。对于 Double DQN，我们使用了与 DQN 完全相同的超参数，

	DQN	Double DQN	Double DQN (tuned)
Median	47.5%	88.4%	116.7%
Mean	122.0%	273.1%	475.2%

表 2：在 49 款以人类先手开始的游戏中，游戏进行至 30 分钟以内的归一化性能汇总。DQN 的结果来自 Nair 等人（2015）。

为了进行一个受控实验，仅聚焦于减少高估。学习到的策略在 5 分钟的模拟器时间（18,000 帧）内进行评估，采用一种 ϵ-贪婪策略，其中 $\epsilon = 0.05$。分数在 100 个回合上取平均。Double DQN 与 DQN 之间唯一的差异在于目标，使用 $Y_t^{\text{DoubleDQN}}$ 而不是 Y^{DQN}。这种评估在某种程度上具有对抗性，因为所用的超参数是为 DQN 调优的，而不是为 Double DQN。

为了获得跨游戏的汇总统计数据，我们按如下方式对每个游戏的得分进行归一化：

$$\text{score}_{\text{normalized}} = \frac{\text{score}_{\text{agent}} - \text{score}_{\text{random}}}{\text{score}_{\text{human}} - \text{score}_{\text{random}}}. \qquad (5)$$

“随机”和“人类”得分与 Mnih 等人（2015）使用的相同，并在附录中给出。

表 1 在无操作（no ops）条件下显示，总体而言 Double DQN 相比 DQN 有明显提升。更为详细的比较（见附录）表明，在若干游戏中 Double DQN 相对 DQN 的改进幅度很大。值得注意的例子包括 Road Runner（从 233% 提升到 617%）、Asterix（从 70% 提升到 180%）、Zaxxon（从 54% 提升到 111%）以及 Double Dunk（从 17% 提升到 397%）。

Gorila 算法（Nair 等，2015）是 DQN 的大规模分布式版本，由于其架构和基础设施差异足够大，难以进行直接比较，因此未被纳入该表。为求完整起见，我们指出 Gorila 分别获得了 96% 和 495% 的中位数与平均归一化得分。

对人类起步的鲁棒性

对先前评估的一个担忧是，在具有唯一起始点的确定性游戏中，学习者可能会学会记住动作序列，而几乎不需要进行泛化。尽管这样也能成功，但该解决方案并不特别鲁棒。通过从不同的起始点测试智能体，我们可以检验所发现的解决方案是否具有良好的泛化能力，从而为学习到的策略提供一个具有挑战性的测试平台（Nair 等，2015）。

我们按照 Nair 等人（2015）提出的方法，从人类专家的轨迹中为每个游戏采样获得了 100 个起始点。我们从这些起始点中的每一个开始一个评估回合，并将模拟器运行至多 108,000 帧（以 60Hz 计为 30 分钟，包括起始点之前的轨迹）。每个智能体仅在起始点之后累积的奖励上进行评估。

在本次评估中，我们纳入了一个经过调参的 Double DQN 版本。进行一定的调参是合适的，因为这些超参数是为 DQN 调整的，而 DQN 是一种不同的算法。对于经过调参的 Double DQN 版本，我们将目标网络两次拷贝之间的帧数从 10,000 增加到 30,000，以进一步减少过估计，因为在每次切换之后立即，DQN 和 Double DQN

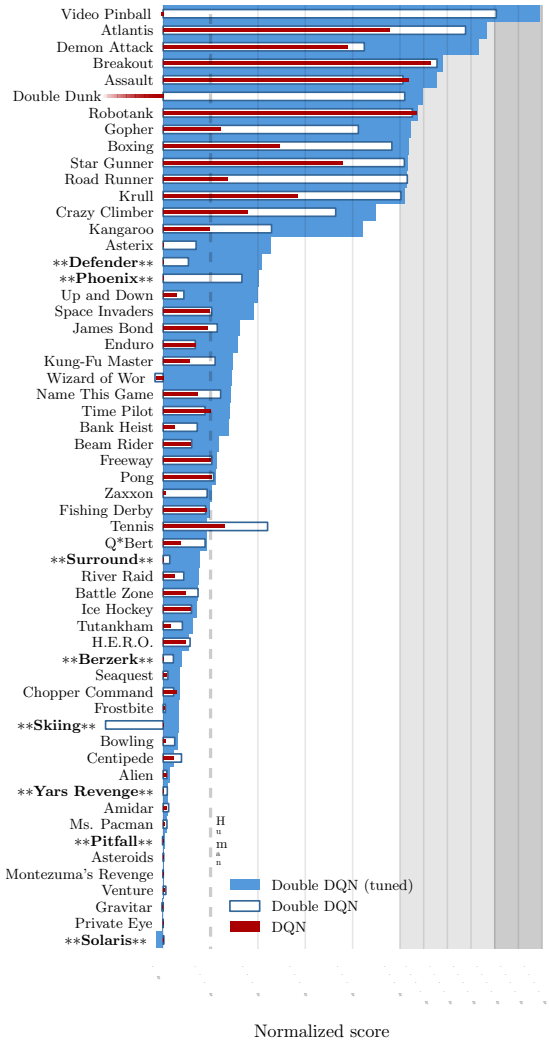

图 4：57 个 Atari 游戏的归一化得分；每个游戏在采用人类起始状态的条件下测试 100 个回合。与 Mnih 等（2015）相比，额外测试了 8 个游戏。这些游戏以星号和加粗字体标示。

两者都恢复为 Q-learning。此外，我们将学习期间的探索从 $\epsilon = 0.1$ 降低到 $\epsilon = 0.01$，并在评估期间使用 $\epsilon = 0.001$。最后，调优版本在网络顶层对所有动作价值使用单个共享偏置。这些更改中的每一项都提升了性能，它们结合在一起带来了明显更好的结果。[3]

表 2 报告了对 Mnih 等（2015）中 49 款游戏进行该评估的汇总统计量。Double DQN 获得了明显更高的中位数和平均得分。同样，表中未包含 Gorila DQN（Nair 等，2015），但为完整起见，注意其取得了 78% 的中位数和 259% 的平均值。更详细的结果，以及另外 8 款游戏的结果，可在图 4 和附录中找到。在若干游戏上，从 DQN 到 Double DQN 的改进十分显著，在某些情况下使得得分更接近于

[3] Except for Tennis, where the lower ϵ during training seemed to hurt rather than help.

人类，甚至超越人类。

Double DQN 在这一更具挑战性的评估中似乎更为稳健，这表明发生了适当的泛化，并且所找到的解决方案并未利用环境的确定性。这一点很有吸引力，因为它表明我们正朝着找到通用解决方案的方向取得进展，而不是依赖一串确定性的步骤序列——那样的方案会更不稳健。

讨论

本文有五项贡献。第一，我们说明了为何在大规模问题中，即使这些问题是确定性的，Q-learning 也可能由于学习过程中固有的估计误差而产生过度乐观。第二，通过分析 Atari 游戏上的价值估计，我们表明这种高估在实践中比先前所承认的更为常见且更为严重。第三，我们展示了 Double Q-learning 可以在大规模场景下用于成功降低这种过度乐观，从而带来更稳定、更可靠的学习。第四，我们提出了一种名为 Double DQN 的具体实现，它沿用 DQN 算法的现有架构与深度神经网络，而无需额外的网络或参数。最后，我们表明 Double DQN 能找到更优的策略，并在 Atari 2600 领域取得了新的最先进结果。

致谢

我们感谢 Tom Schaul、Volodymyr Mnih、Marc Bellemare、Thomas Degris、Georg Ostrovski 和 Richard Sutton 提供的宝贵意见，并感谢 Google DeepMind 的所有人营造了富有建设性的研究环境。

参考文献

R. Agrawal。基于样本均值的索引策略：用于多臂老虎机问题的 O(log n) 级遗憾。*Advances in Applied Probability*，第 1054–1078 页，1995 年。P. Auer、N. Cesa-Bianchi 和 P. Fischer。多臂老虎机问题的有限时间分析。*Machine learning*，47(2-3):235–256，2002 年。L. Baird。残差算法：带函数逼近的强化学习。载于 *Machine Learning: Proceedings of the Twelfth International Conference*，第 30–37 页，1995 年。M. G. Bellemare、Y. Naddaf、J. Veness 和 M. Bowling。街机学习环境：用于通用智能体的评估平台。*J. Artif. Intell. Res. (JAIR)*，47:253–279，2013 年。R. I. Brafman 和 M. Tennenholtz。R-max：一种用于近最优强化学习的通用多项式时间算法。*The Journal of Machine Learning Research*，3:213–231，2003 年。K. Fukushima。Neocognitron：一种能够进行视觉模式识别的分层神经网络。*Neural networks*，1(2):119–130，1988 年。L. P. Kaelbling、M. L. Littman 和 A. W. Moore。强化学习：综述。*Journal of Artificial Intelligence Research*，4:237–285，1996 年。Y. LeCun、L. Bottou、Y. Bengio 和 P. Haffner。用于文档识别的基于梯度的学习。*Proceedings of the IEEE*，86(11):2278–2324，1998 年。L. Lin。基于强化学习、规划与教学的自我改进反应式智能体。*Machine learning*，8(3):293–321，1992 年。H. R. Maei。*Gradient temporal-difference learning algorithms*。博士论文，阿尔伯塔大学，2011。V. Mnih、K. Kavukcuoglu、D. Silver、A. A. Rusu、J. Veness、M. G. Bellemare、A. Graves、M. Riedmiller、A. K. Fidjeland、G. Ostrovski、S. Petersen、C. Beattie、A. Sadik、I. Antonoglou、H. King、D. Kumaran、D. Wierstra、S. Legg 和 D. Hassabis。通过深度强化学习实现人类水平的控制。*Nature*，518(7540):529–533，2015。A. Nair、P. Srinivasan、S. Blackwell、C. Alcicek、R. Fearon、A. D. Maria、V. Panneershelvam、M. Suleyman、C. Beattie、S. Petersen、S. Legg、V. Mnih、K. Kavukcuoglu 和 D. Silver。用于深度强化学习的大规模并行方法。见 *Deep Learning Workshop, ICML*，2015。M. Riedmiller。神经拟合 Q 迭代——对一种数据高效的神经强化学习方法的初步经验。见 J. Gama、R. Camacho、P. Brazdil、A. Jorge 和 L. Torgo（编），*Proceedings of the 16th European Conference on Machine Learning (ECML'05)*，第 317–328 页。Springer，2005。B. Sallans 和 G. E. Hinton。具有分解状态与动作的强化学习。*The Journal of Machine Learning Research*，5:1063–1088，2004。A. L. Strehl、L. Li 和 M. L. Littman。有限 MDP 中的强化学习：PAC 分析。*The Journal of Machine Learning Research*，10:2413–2444，2009。R. S. Sutton。通过时序差分方法学习预测。*Machine learning*，3(1):9–44，1988。R. S. Sutton。基于近似动态规划的学习、规划与反应的一体化架构。见 *Proceedings of the seventh international conference on machine learning*，第 216–224 页，1990。R. S. Sutton 和 A. G. Barto。*Introduction to reinforcement learn- ing*。MIT Press，1998。R. S. Sutton、C. Szepesvári 和 H. R. Maei。一种用于带线性函数逼近的离策略时序差分学习的收敛 O(n) 算法。*Advances in Neural Information Processing Systems 21 (NIPS-08)*，21:1609–1616，2008。R. S. Sutton、A. R. Mahmood 和 M. White。针对离策略时序差分学习问题的强调式方法。*arXiv preprint arXiv:1503.04269*，2015。I. Szita 和 A. Lőrincz。乐观主义的多种面孔：一种统一的方法。见 *Proceedings of the 25th international conference on Machine learning*，第 1048–1055 页。ACM，2008。G. Tesauro。时序差分学习与 TD-Gammon。*Communications of the ACM*，38(3):58–68，1995。S. Thrun 和 A. Schwartz。在强化学习中使用函数逼近的问题。见 M. Mozer、P. Smolensky、D. Touretzky、J. Elman 和 A. Weigend（编），*Proceedings of the 1993 Connectionist Models Summer School*，新泽西州希尔斯代尔，1993。Lawrence Erlbaum。J. N. Tsitsiklis 和 B. Van Roy。带函数逼近的时序差分学习分析。*IEEE Transactions on Automatic Control*，42(5):674–690，1997。H. van Hasselt。双重 Q 学习。*Advances in Neural Information Processing Systems*，23:2613–2621，2010。H. van Hasselt。*Insights in Reinforcement Learning*。博士论文，乌得勒支大学，2011。C. J. C. H. Watkins。*Learning from delayed rewards*。博士论文，英国剑桥大学，1989。

附录

定理 1。*Consider a state s in which all the true optimal action values are equal at $Q_*(s,a) = V_*(s)$ for some $V_*(s)$. Let Q_t be arbitrary value estimates that are on the whole unbiased in the sense that $\sum_a (Q_t(s,a) - V_*(s)) = 0$, but that are not all zero, such that $\frac{1}{m}\sum_a (Q_t(s,a) - V_*(s))^2 = C$ for some $C > 0$, where $m \geq 2$ is the number of actions in s. Under these conditions, $\max_a Q_t(s,a) \geq V_*(s) + \sqrt{\frac{C}{m-1}}$. This lower bound is tight. Under the same conditions, the lower bound on the absolute error of the Double Q-learning estimate is zero.*

Proof of Theorem 1. 将每个动作 a 的误差定义为 $\epsilon_a = Q_t(s,a) - V_*(s)$。假设存在一种 $\{\epsilon_a\}$ 的设定，使得 $\max_a \epsilon_a < \sqrt{\frac{C}{m-1}}$。令 $\{\epsilon_i^+\}$ 为大小为 n 的正 ϵ 的集合，$\{\epsilon_j^-\}$ 为大小为 $m-n$（的严格负 ϵ 的集合，并满足 $\{\epsilon\} = \{\epsilon_i^+\} \cup \{\epsilon_j^-\}$）。如果 $n = m$，则 $\sum_a \epsilon_a = 0 \Longrightarrow \epsilon_a = 0\ \forall a$，这与 $\sum_a \epsilon_a^2 = mC$ 矛盾。因此，必有 $n \leq m-1$。于是，$\sum_{i=1}^n \epsilon_i^+ \leq n \max_i \epsilon_i^+ < n\sqrt{\frac{C}{m-1}}$，并且（利用约束 $\sum_a \epsilon_a = 0$）我们也有 $\sum_{j=1}^{m-n} |\epsilon_j^-| < n\sqrt{\frac{C}{m-1}}$。这意味着 $\max_j |\epsilon_j^-| < n\sqrt{\frac{C}{m-1}}$。由 Hölder 不等式，则

$$\begin{aligned}\sum_{j=1}^{m-n} (\epsilon_j^-)^2 &\leq \sum_{j=1}^{m-n} |\epsilon_j^-| \cdot \max_j |\epsilon_j^-| \\ &< n\sqrt{\frac{C}{m-1}} n\sqrt{\frac{C}{m-1}}\,.\end{aligned}$$

我们现在可以将这些关系结合起来，以计算所有 ϵ_a 的平方和的上界：

$$\begin{aligned}\sum_{a=1}^{m} (\epsilon_a)^2 &= \sum_{i=1}^{n} (\epsilon_i^+)^2 + \sum_{j=1}^{m-n} (\epsilon_j^-)^2 \\ &< n\frac{C}{m-1} + n\sqrt{\frac{C}{m-1}} n\sqrt{\frac{C}{m-1}} \\ &= C\frac{n(n+1)}{m-1} \\ &\leq mC.\end{aligned}$$

这与$\sum_{a=1}^m \epsilon_a^2 < mC$的假设相矛盾，因此对所有满足约束的$\epsilon$设置都有$\max_a \epsilon_a \geq \sqrt{\frac{C}{m-1}}$。我们可以通过对$a = 1, \ldots, m-1$并且$\epsilon_m = -\sqrt{(m-1)C}$时设定$\epsilon_a = \sqrt{\frac{C}{m-1}}$来检验该下界是紧的。这验证了$\sum_a \epsilon 2_a = mC$ 和 $\sum_a \epsilon_a = 0$.

Double Q-学习 $|Q'_t(s, \operatorname{argmax}_a Q_t(s,a)) - V_*(s)|$ 的绝对误差唯一的紧下界是零。这可以看出来，因为我们可以有

$$Q_t(s,a_1) = V_*(s) + \sqrt{C\frac{m-1}{m}}\,,$$

和

$$Q_t(s,a_i) = V_*(s) - \sqrt{C\frac{1}{m(m-1)}}\,, \text{for } i > 1.$$

那么该定理的条件成立。如果进一步我们有 $Q'_t(s,a_1) = V_*(s)$，则误差为零。其余的动作值 $Q'_t(s,a_i)$，对于 $i > 1$，是任意的。 □

定理 2。*Consider a state s in which all the true optimal action values are equal at $Q_*(s,a) = V_*(s)$. Suppose that the estimation errors $Q_t(s,a) - Q_*(s,a)$ are independently distributed uniformly randomly in $[-1, 1]$. Then,*

$$\mathbb{E}\left[\max_a Q_t(s,a) - V_*(s)\right] = \frac{m-1}{m+1}$$

Proof. 定义 $\epsilon_a = Q_t(s,a) - Q_*(s,a)$；这是一个在 $[-1,1]$ 上均匀的随机变量。对于某个 x，$\max_a Q_t(s,a) \leq x$ 的概率等于对所有 a 同时满足 $\epsilon_a \leq x$ 的概率。由于估计误差相互独立，我们可以推导出

$$\begin{aligned}P(\max_a \epsilon_a \leq x) &= P(X_1 \leq x \wedge X_2 \leq x \wedge \ldots \wedge X_m \leq x) \\ &= \prod_{a=1}^{m} P(\epsilon_a \leq x)\,.\end{aligned}$$

函数 $P(\epsilon_a \leq x)$ 是 ϵ_a 的累积分布函数（CDF），这里将其简单定义为

$$P(\epsilon_a \leq x) = \begin{cases} 0 & \text{if } x \leq -1 \\ \frac{1+x}{2} & \text{if } x \in (-1,1) \\ 1 & \text{if } x \geq 1 \end{cases}$$

这意味着

$$\begin{aligned}P(\max_a \epsilon_a \leq x) &= \prod_{a=1}^{m} P(\epsilon_a \leq x) \\ &= \begin{cases} 0 & \text{if } x \leq -1 \\ \left(\frac{1+x}{2}\right)^m & \text{if } x \in (-1,1) \\ 1 & \text{if } x \geq 1 \end{cases}\end{aligned}$$

这给出了随机变量 $\max_a \epsilon_a$ 的累积分布函数（CDF）。其期望可以写成一个积分

$$\mathbb{E}\left[\max_a \epsilon_a\right] = \int_{-1}^{1} x f_{\max}(x)\,\mathrm{d}x\,,$$

其中 $f_{\max}$ 是该变量的概率密度函数，定义为 CDF 的导数：$f_{\max}(x) = \frac{\mathrm{d}}{\mathrm{d}x} P(\max_a \epsilon_a \leq x)$，因此对于 $x \in [-1,1]$ 我们有 $f_{\max}(x) = \frac{m}{2}\left(\frac{1+x}{2}\right)^{m-1}$。对该积分进行计算可得

$$\begin{aligned}\mathbb{E}\left[\max_a \epsilon_a\right] &= \int_{-1}^{1} x f_{\max}(x)\,\mathrm{d}x \\ &= \left[\left(\frac{x+1}{2}\right)^m \frac{mx-1}{m+1}\right]_{-1}^{1} \\ &= \frac{m-1}{m+1}\,.\end{aligned}$$

□

Atari 2600 域的实验细节

我们选择了 49 款游戏，以匹配 Mnih 等人（2015）使用的列表；完整列表见下方表格。每个智能体步由四帧组成（在这些帧期间重复最后一次选择的动作），并且奖励值（来自 Arcade Learning Environment（Bellemare 等人，2013））被裁剪到 -1 与 1 之间。

网络架构

实验中使用的卷积网络与 Mnih 等人（2015）提出的完全一致，这里仅为完整性起见提供细节。网络的输入为一个 84x84x4 的张量，包含最近四帧的缩放后灰度版本。第一层卷积层使用 32 个大小为 8（步幅 4）的滤波器对输入进行卷积，第二层有 64 个大小为 4（步幅 2）的滤波器，最后一层卷积层有 64 个大小为 3（步幅 1）的滤波器。随后接一个包含 512 个单元的全连接隐藏层。所有这些层之间都由修正线性单元（ReLu）分隔。最后，一个全连接线性层将其投影到网络输出，即 Q 值。用于训练网络的优化方法为 RMSProp（动量参数为 0.95）。

超参数

在所有实验中，折扣因子设为 $\gamma = 0.99$，学习率设为 $\alpha = 0.00025$。目标网络更新之间的步数为 $\tau = 10,000$。训练在 50M 步（即 200M 帧）上进行。智能体每 1M 步评估一次，并保留这些评估中表现最好的策略作为学习过程的输出。经验回放存储器的大小为 1M 个元组。每 4 步从存储器中采样一次，以大小为 32 的小批量来更新网络。所使用的简单探索策略是 ϵ-贪婪策略，其中 ϵ 在 1M 步内从 1 线性下降到 0.1。

Atari 2600 领域的补充结果

下表提供了我们在 Atari 领域实验的更详细结果。

Game	Random	Human	DQN	Double DQN
Alien	227.80	6875.40	3069.33	2907.30
Amidar	5.80	1675.80	739.50	702.10
Assault	222.40	1496.40	3358.63	5022.90
Asterix	210.00	8503.30	6011.67	15150.00
Asteroids	719.10	13156.70	1629.33	930.60
Atlantis	12850.00	29028.10	85950.00	64758.00
Bank Heist	14.20	734.40	429.67	728.30
Battle Zone	2360.00	37800.00	26300.00	25730.00
Beam Rider	363.90	5774.70	6845.93	7654.00
Bowling	23.10	154.80	42.40	70.50
Boxing	0.10	4.30	71.83	81.70
Breakout	1.70	31.80	401.20	375.00
Centipede	2090.90	11963.20	8309.40	4139.40
Chopper Command	811.00	9881.80	6686.67	4653.00
Crazy Climber	10780.50	35410.50	114103.33	101874.00
Demon Attack	152.10	3401.30	9711.17	9711.90
Double Dunk	-18.60	-15.50	-18.07	-6.30
Enduro	0.00	309.60	301.77	319.50
Fishing Derby	-91.70	5.50	-0.80	20.30
Freeway	0.00	29.60	30.30	31.80
Frostbite	65.20	4334.70	328.33	241.50
Gopher	257.60	2321.00	8520.00	8215.40
Gravitar	173.00	2672.00	306.67	170.50
H.E.R.O.	1027.00	25762.50	19950.33	20357.00
Ice Hockey	-11.20	0.90	-1.60	-2.40
James Bond	29.00	406.70	576.67	438.00
Kangaroo	52.00	3035.00	6740.00	13651.00
Krull	1598.00	2394.60	3804.67	4396.70
Kung-Fu Master	258.50	22736.20	23270.00	29486.00
Montezuma's Revenge	0.00	4366.70	0.00	0.00
Ms. Pacman	307.30	15693.40	2311.00	3210.00
Name This Game	2292.30	4076.20	7256.67	6997.10
Pong	-20.70	9.30	18.90	21.00
Private Eye	24.90	69571.30	1787.57	670.10
Q*Bert	163.90	13455.00	10595.83	14875.00
River Raid	1338.50	13513.30	8315.67	12015.30
Road Runner	11.50	7845.00	18256.67	48377.00
Robotank	2.20	11.90	51.57	46.70
Seaquest	68.40	20181.80	5286.00	7995.00
Space Invaders	148.00	1652.30	1975.50	3154.60
Star Gunner	664.00	10250.00	57996.67	65188.00
Tennis	-23.80	-8.90	-2.47	1.70
Time Pilot	3568.00	5925.00	5946.67	7964.00
Tutankham	11.40	167.60	186.70	190.60
Up and Down	533.40	9082.00	8456.33	16769.90
Venture	0.00	1187.50	380.00	93.00
Video Pinball	16256.90	17297.60	42684.07	70009.00
Wizard of Wor	563.50	4756.50	3393.33	5204.00
Zaxxon	32.50	9173.30	4976.67	10182.00

表 3：无操作评估条件下的原始得分（模拟器时间 5 分钟）。DQN 如 Mnih 等（2015）所述。

Game	DQN	Double DQN
Alien	42.75 %	40.31 %
Amidar	43.93 %	41.69 %
Assault	246.17 %	376.81 %
Asterix	69.96 %	180.15 %
Asteroids	7.32 %	1.70 %
Atlantis	451.85 %	320.85 %
Bank Heist	57.69 %	99.15 %
Battle Zone	67.55 %	65.94 %
Beam Rider	119.80 %	134.73 %
Bowling	14.65 %	35.99 %
Boxing	1707.86 %	1942.86 %
Breakout	1327.24 %	1240.20 %
Centipede	62.99 %	20.75 %
Chopper Command	64.78 %	42.36 %
Crazy Climber	419.50 %	369.85 %
Demon Attack	294.20 %	294.22 %
Double Dunk	17.10 %	396.77 %
Enduro	97.47 %	103.20 %
Fishing Derby	93.52 %	115.23 %
Freeway	102.36 %	107.43 %
Frostbite	6.16 %	4.13 %
Gopher	400.43 %	385.66 %
Gravitar	5.35 %	-0.10 %
H.E.R.O.	76.50 %	78.15 %
Ice Hockey	79.34 %	72.73 %
James Bond	145.00 %	108.29 %
Kangaroo	224.20 %	455.88 %
Krull	277.01 %	351.33 %
Kung-Fu Master	102.37 %	130.03 %
Montezuma's Revenge	0.00 %	0.00 %
Ms. Pacman	13.02 %	18.87 %
Name This Game	278.29 %	263.74 %
Pong	132.00 %	139.00 %
Private Eye	2.53 %	0.93 %
Q*Bert	78.49 %	110.68 %
River Raid	57.31 %	87.70 %
Road Runner	232.91 %	617.42 %
Robotank	508.97 %	458.76 %
Seaquest	25.94 %	39.41 %
Space Invaders	121.49 %	199.87 %
Star Gunner	598.09 %	673.11 %
Tennis	143.15 %	171.14 %
Time Pilot	100.92 %	186.51 %
Tutankham	112.23 %	114.72 %
Up and Down	92.68 %	189.93 %
Venture	32.00 %	7.83 %
Video Pinball	2539.36 %	5164.99 %
Wizard of Wor	67.49 %	110.67 %
Zaxxon	54.09 %	111.04 %

表 4：无操作评估条件下的归一化结果（模拟器时间 5 分钟）。

Game	Random	Human	DQN	Double DQN	Double DQN (tuned)
Alien	128.30	6371.30	570.2	621.6	1033.4
Amidar	11.80	1540.40	133.4	188.2	169.1
Assault	166.90	628.90	3332.3	2774.3	6060.8
Asterix	164.50	7536.00	124.5	5285.0	16837.0
Asteroids	871.30	36517.30	697.1	1219.0	1193.2
Atlantis	13463.00	26575.00	76108.0	260556.0	319688.0
Bank Heist	21.70	644.50	176.3	469.8	886.0
Battle Zone	3560.00	33030.00	17560.0	25240.0	24740.0
Beam Rider	254.60	14961.00	8672.4	9107.9	17417.2
Berzerk	196.10	2237.50		635.8	1011.1
Bowling	35.20	146.50	41.2	62.3	69.6
Boxing	-1.50	9.60	25.8	52.1	73.5
Breakout	1.60	27.90	303.9	338.7	368.9
Centipede	1925.50	10321.90	3773.1	5166.6	3853.5
Chopper Command	644.00	8930.00	3046.0	2483.0	3495.0
Crazy Climber	9337.00	32667.00	50992.0	94315.0	113782.0
Defender	1965.50	14296.00		8531.0	27510.0
Demon Attack	208.30	3442.80	12835.2	13943.5	69803.4
Double Dunk	-16.00	-14.40	-21.6	-6.4	-0.3
Enduro	-81.80	740.20	475.6	475.9	1216.6
Fishing Derby	-77.10	5.10	-2.3	-3.4	3.2
Freeway	0.10	25.60	25.8	26.3	28.8
Frostbite	66.40	4202.80	157.4	258.3	1448.1
Gopher	250.00	2311.00	2731.8	8742.8	15253.0
Gravitar	245.50	3116.00	216.5	170.0	200.5
H.E.R.O.	1580.30	25839.40	12952.5	15341.4	14892.5
Ice Hockey	-9.70	0.50	-3.8	-3.6	-2.5
James Bond	33.50	368.50	348.5	416.0	573.0
Kangaroo	100.00	2739.00	2696.0	6138.0	11204.0
Krull	1151.90	2109.10	3864.0	6130.4	6796.1
Kung-Fu Master	304.00	20786.80	11875.0	22771.0	30207.0
Montezuma's Revenge	25.00	4182.00	50.0	30.0	42.0
Ms. Pacman	197.80	15375.00	763.5	1401.8	1241.3
Name This Game	1747.80	6796.00	5439.9	7871.5	8960.3
Phoenix	1134.40	6686.20		10364.0	12366.5
Pit Fall	-348.80	5998.90		-432.9	-186.7
Pong	-18.00	15.50	16.2	17.7	19.1
Private Eye	662.80	64169.10	298.2	346.3	-575.5
Q*Bert	183.00	12085.00	4589.8	10713.3	11020.8
River Raid	588.30	14382.20	4065.3	6579.0	10838.4
Road Runner	200.00	6878.00	9264.0	43884.0	43156.0
Robotank	2.40	8.90	58.5	52.0	59.1
Seaquest	215.50	40425.80	2793.9	4199.4	14498.0
Skiing	-15287.40	-3686.60		-29404.3	-11490.4
Solaris	2047.20	11032.60		2166.8	810.0
Space Invaders	182.60	1464.90	1449.7	1495.7	2628.7
Star Gunner	697.00	9528.00	34081.0	53052.0	58365.0
Surround	-9.70	5.40		-7.6	1.9
Tennis	-21.40	-6.70	-2.3	11.0	-7.8
Time Pilot	3273.00	5650.00	5640.0	5375.0	6608.0
Tutankham	12.70	138.30	32.4	63.6	92.2
Up and Down	707.20	9896.10	3311.3	4721.1	19086.9
Venture	18.00	1039.00	54.0	75.0	21.0
Video Pinball	20452.0	15641.10	20228.1	148883.6	367823.7
Wizard of Wor	804.00	4556.00	246.0	155.0	6201.0
Yars Revenge	1476.90	47135.20		5439.5	6270.6
Zaxxon	475.00	8443.00	831.0	7874.0	8593.0

表 5：人类起始条件下的原始得分（模拟器时间 30 分钟）。DQN 如 Nair 等人（2015）所述。

Game	DQN	Double DQN	Double DQN (tuned)
Alien	7.08%	7.90%	14.50%
Amidar	7.95%	11.54%	10.29%
Assault	685.15%	564.37%	1275.74%
Asterix	-0.54%	69.46%	226.18%
Asteroids	-0.49%	0.98%	0.90%
Atlantis	477.77%	1884.48%	2335.46%
Bank Heist	24.82%	71.95%	138.78%
Battle Zone	47.51%	73.57%	71.87%
Beam Rider	57.24%	60.20%	116.70%
Berzerk		21.54%	39.92%
Bowling	5.39%	24.35%	30.91%
Boxing	245.95%	482.88%	675.68%
Breakout	1149.43%	1281.75%	1396.58%
Centipede	22.00%	38.60%	22.96%
Chopper Command	28.99%	22.19%	34.41%
Crazy Climber	178.55%	364.24%	447.69%
Defender		53.25%	207.17%
Demon Attack	390.38%	424.65%	2151.65%
Double Dunk	-350.00%	600.00%	981.25%
Enduro	67.81%	67.85%	157.96%
Fishing Derby	91.00%	89.66%	97.69%
Freeway	100.78%	102.75%	112.55%
Frostbite	2.20%	4.64%	33.40%
Gopher	120.42%	412.07%	727.95%
Gravitar	-1.01%	-2.63%	-1.57%
H.E.R.O.	46.88%	56.73%	54.88%
Ice Hockey	57.84%	59.80%	70.59%
James Bond	94.03%	114.18%	161.04%
Kangaroo	98.37%	228.80%	420.77%
Krull	283.34%	520.11%	589.66%
Kung-Fu Master	56.49%	109.69%	145.99%
Montezuma's Revenge	0.60%	0.12%	0.41%
Ms. Pacman	3.73%	7.93%	6.88%
Name This Game	73.14%	121.30%	142.87%
Phoenix		166.25%	202.31%
Pit Fall		-1.32%	2.55%
Pong	102.09%	106.57%	110.75%
Private Eye	-0.57%	-0.50%	-1.95%
Q*Bert	37.03%	88.48%	91.06%
River Raid	25.21%	43.43%	74.31%
Road Runner	135.73%	654.15%	643.25%
Robotank	863.08%	763.08%	872.31%
Seaquest	6.41%	9.91%	35.52%
Skiing		-121.69%	32.73%
Solaris		1.33%	-13.77%
Space Invaders	98.81%	102.40%	190.76%
Star Gunner	378.03%	592.85%	653.02%
Surround		13.91%	76.82%
Tennis	129.93%	220.41%	92.52%
Time Pilot	99.58%	88.43%	140.30%
Tutankham	15.68%	40.53%	63.30%
Up and Down	28.34%	43.68%	200.02%
Venture	3.53%	5.58%	0.29%
Video Pinball	-4.65%	2669.60%	7220.51%
Wizard of Wor	-14.87%	-17.30%	143.84%
Yars Revenge		8.68%	10.50%
Zaxxon	4.47%	92.86%	101.88%

表 6：人类起始条件的归一化得分（模拟器时间 30 分钟）。