APPENDICES: Adaptive Trade-Offs in Off-Policy Learning

A Proofs

Proposition 2.1. Consider the task of evaluation of a policy π under behaviour μ, and consider an update rule $\hat{T}$ which stochastically approximates the application of an operator $\widetilde{T}$, with contraction rate Γ and fixed point $\widetilde{Q}$, to an initial estimate Q. Then we have the following decomposition:

$$\mathbb{E}\left[\|\hat{T}Q - Q^\pi\|_\infty\right] \leq \underbrace{\mathbb{E}\left[\|\hat{T}Q - \widetilde{T}Q\|_2^2\right]^{1/2}}_{\text{(Root) variance}} + \underbrace{\Gamma\|Q - \widetilde{Q}\|_\infty}_{\text{Contraction}} + \underbrace{\|\widetilde{Q} - Q^\pi\|_2}_{\text{Fixed-point bias}} .$$

Proof. Note that by the triangle inequality:

$$\mathbb{E}\left[\|\hat{T}Q - Q^\pi\|_\infty\right] \leq \mathbb{E}\left[\|\hat{T}Q - \widetilde{T}Q\|_\infty\right] + \|\widetilde{T}Q - \widetilde{Q}\|_\infty + \|\widetilde{Q} - Q^\pi\|_\infty .$$

Now, observing $\|\widetilde{T}Q - \widetilde{Q}\|_\infty = \|\widetilde{T}Q - \widetilde{T}\widetilde{Q}\|_\infty \leq \Gamma\|Q - \widetilde{Q}\|_\infty$ yields the second term on the right-hand side of the stated bound. Using the inequality $\|\cdot\|_\infty \leq \|\cdot\|_2$ and Jensen's inequality yields the remaining terms. □

Theorem 2.2. Consider an update rule $\hat{T}$ with corresponding operator T, and consider the collection $\mathcal{M} = \mathcal{M}(\mathcal{X}, \mathcal{A}, P, \gamma, R_{\max})$ of MDPs with common state space, action space, transition kernel, and discount factor (but varying rewards, with maximum immediate reward bounded in absolute value by $R_{\max}$). Fix a target policy π, and a random variable Z, the set of transitions used by the operator $\hat{T}$; these could be transitions encountered in a trajectory following the behaviour μ, or i.i.d. samples from the discounted state-action visitation distribution under μ. We denote the mismatch between π and Z at level $\delta \in (0, 1)$ by

$$D(Z, \pi, \delta) \stackrel{\text{def}}{=} \max\{d_{(x,a),\pi}(\Omega) \mid \Omega \subseteq \mathcal{X} \times \mathcal{A} \text{ s.t. } \mathbb{P}(Z \cap \Omega \neq \emptyset) \leq \delta\,, (x, a) \in \mathcal{X} \times \mathcal{A}\} ,$$

where $d_{(x,a),\pi}$ is the discounted state-action visitation distribution for trajectories initialised with (x, a), following π thereafter. Denoting the variance, contraction rate, and fixed-point bias of $\hat{T}$ for a particular MDP $M \in \mathcal{M}$ by $\mathbb{V}(M)$, $\Gamma(M)$ and $B(M)$ respectively, we have

$$\sup_{M \in \mathcal{M}} \left[\sqrt{\mathbb{V}(M)} + \frac{2R_{\max}}{1 - \gamma}\Gamma(M) + B(M)\right] \geq \sup_{\delta \in (0,1)} (1 - \delta) D(Z, \pi, \delta) R_{\max}/(1 - \gamma) .$$

Proof. The high-level approach to the proof is to exhibit two MDPs $M_0, M_1 \in \mathcal{M}$ which with high probability under the data used by $\hat{T}$, cannot be distinguished. This yields a high probability lower bound on the evaluation error that the operator $\hat{T}$ achieves on the two MDPs. This in turn implies that the mean-squared error quantity of Proposition 2.1 cannot be uniformly low across M_0 and M_1, and this yields a lower bound for the quantity on the right-hand side of the bound appearing in Proposition 2.1, as required.

Using the notation introduced in the statement of the theorem, for a given $\delta \in (0, 1)$, let $(x^*, a^*) \in \mathcal{X} \times \mathcal{A}$, $\Omega^* \subseteq \mathcal{X}$ be quantities achieving the maximum in the definition of $D(Z, \pi, \delta)$. Thus, with probability at least $1 - \delta$, none of the state-action pairs used by the algorithm $\hat{T}$ are contained in Ω^*.

Now define two MDPs M_0, M_1 with common state space $\mathcal{X}$, action space $\mathcal{A}$, transition kernel P, and discount factor γ, with reward functions $r_0, r_1 : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ defined by

$$r_i(x, a) = \begin{cases} 0 & \text{if } x \notin \Omega^* \\ (-1)^i R_{\max} & \text{if } x \in \Omega^* . \end{cases}$$

Now, the Q-functions associated with these two MDPs can be calculated by $(I-\gamma P^\pi)^{-1}r_0$ and $(I-\gamma P^\pi)^{-1}r_1$, and so we can read off their difference in the (x^*,a^*) coordinate as

$$\sum_{(x,a)\in\Omega^*}\frac{1}{1-\gamma}d_{(x^*,a^*),\pi}(x,a)2R_{\max}=\frac{2D(Z,\pi,\delta)R_{\max}}{1-\gamma}\,.$$

Thus, with probability $1-\delta$, the algorithm must make an error of at least $D(Z,\pi,\delta)R_{\max}/(1-\gamma)$ on one of the MDPs M_0 and M_1, as measured the L^∞ norm. This implies that the expected L^∞ error appearing on the left-hand side of the bound in Proposition 2.1 is at least $(1-\delta)D(Z,\pi,\delta)R_{\max}/(1-\gamma)$ for one of the MDPs M_0 and M_1. Thus, the statement of the theorem follows by taking a supremum over $\delta\in(0,1)$. □

Proposition 3.2. The operator associated with the α-Retrace update for evaluating π given behaviour μ has a state-action-dependent contraction rate of

$$C(\alpha|x,a)\stackrel{\text{def}}{=}1-(1-\gamma)\times \tag{5}$$
$$\mathbb{E}_\mu\left[\sum_{t=0}^{\infty}\gamma^t\prod_{s=1}^{t}((1-\alpha)+\alpha\bar{\rho}_s)\middle|(X_0,A_0)=(x,a)\right],$$

for each $(x,a)\in\mathcal{X}\times\mathcal{A}$. Viewed as a function of $\alpha\in[0,1]$, this contraction rate is continuous, monotonically increasing, with minimal value 0, and maximal value no greater than γ. Further, the contraction rate is *strictly* monotonic iff π and μ are (x,a)-distinguishable under μ.

Proof. The α-Retrace operator for evaluation of π given behaviour μ corresponds to the standard Retrace operator for evaluation of $\pi^\alpha=\alpha\pi+(1-\alpha)\mu$ given behaviour μ. Thus, from the analysis of Munos et al. [2016], the contraction rate of the α-Retrace operator specific to a particular state-action pair $(x,a)\in\mathcal{X}\times\mathcal{A}$ may be immediately written down as

$$\begin{aligned}&1-(1-\gamma)\mathbb{E}_\mu\left[\sum_{t\geq0}\gamma^t\prod_{s=1}^{t}\min\left(1,\frac{\alpha\pi(A_t|X_t)+(1-\alpha)\mu(A_t|X_t)}{\mu(A_t|X_t)}\right)\middle|X_0=x,A_0=a\right]\\ =&1-(1-\gamma)\mathbb{E}_\mu\left[\sum_{t\geq0}\gamma^t\prod_{s=1}^{t}\left((1-\alpha)+\alpha\min\left(1,\frac{\pi(A_t|X_t)}{\mu(A_t|X_t)}\right)\right)\middle|X_0=x,A_0=a\right].\end{aligned}$$

To see that this is a continuous function of $\alpha\in[0,1]$, we note that the integrand of the expectation above is clearly a continuous function of α, and is uniformly dominated by the constant function equal to $(1-\gamma)^{-1}$. By the dominated convergence theorem, continuity of the above expression follows. Since the integrand is non-negative and bounded above by $(1-\gamma)^{-1}$, the contraction rate must lie in the interval $[0,\gamma]$ for all $\alpha\in[0,1]$. For monotonicity, we show that each term

$$\mathbb{E}_\mu\left[\prod_{s=1}^{t}\left((1-\alpha)+\alpha\min\left(1,\frac{\pi(A_t|X_t)}{\mu(A_t|X_t)}\right)\right)\middle|X_0=x,A_0=a\right] \tag{8}$$

is monotonic decreasing in α, meaning that the contraction rate is monotonic increasing in α. To this end, observe that the integrand of the expectation above almost-surely takes the form $\prod_{s=1}^{t}(1-z_s\alpha)$ for some coefficients $z_s\in[0,1]$. The derivative with respect to α of this expression is $\sum_{s=1}^{t}-z_s\prod_{s'\neq s}^{t}(1-z_{s'}\alpha)$, which is non-positive for $\alpha\in[0,1]$. It is again straightforward to apply the dominated convergence theorem to this derivative to obtain that the derivative of Expression (8) is non-positive for all $\alpha\in[0,1]$, and we thus obtain monotonicity as required. Finally, for strict monotonicity, note that if π and μ are *not* distinguishable under μ, then all truncated importance weights in the expressions above are equal to 1 almost-surely under the distribution over states visited when following μ. Hence, the contraction rate is in fact constant as a function of α, and we therefore do not have strict monotonicity. On the other hand, if π and μ are (x,a)-distinguishable under μ, then there exists a $t\in\mathbb{N}$ such that in the integrand of the expectation in Expression (8), in one of the terms constituting the product, the coefficient of α is less than 0 with positive probability. Thus, the integrand is strictly monotonic with positive probability, and hence Expression (8) itself is strictly monotonic, proving strict monotonicity of the contraction rate, as required. □

Proposition 3.3. Consider a target policy π, let μ be the behavioural policy, and assume that there is a unique greedy action $a^*(x) \in \mathcal{A}$ with respect to Q^π at each state x for each $x \in \mathcal{X}$. Then there exists a value of $\alpha \in (0, 1)$ such that the greedy policy with respect to the fixed point of α-Retrace coincides with the greedy policy with respect to Q^π, and the contraction rate for this α-Retrace is no greater than that for 1-Retrace. Further, if π and μ are (x, a)-distinguishable under μ for all $(x, a) \in \mathcal{X} \times \mathcal{A}$, then the contraction rate of α-Retrace is strictly lower than that of 1-Retrace.

Proof. That the greedy policies coincide follows as a consequence of the continuity of Q^ν with respect to the policy ν and the positivity of the minimum action gap $\Delta = \inf_{x \in \mathcal{X}, a \neq a^*(x)} (Q^\pi(x, a^*(x)) - Q^\pi(x, a))$; α may be selected so that e.g. $\|Q^{\pi^\alpha} - Q^\pi\|_\infty \leq \Delta/2$. The contraction result follows from the monotonicity property derived in Proposition 3.2. □

Proposition 3.4. Let $(x_t^{(k)}, a_t^{(k)}, r_t^{(k)})_{t=0}^\infty$ be an i.i.d. sequence of trajectories following μ, with initial state-action distribution given by ν. Let Γ be a target contraction rate such that $C_\nu(1) \geq \Gamma$. Let the stepsizes $(\varepsilon_k)_{k=0}^\infty$ satisfy the usual Robbins-Monro conditions $\sum_{k=0}^\infty \varepsilon_k = \infty$, $\sum_{k=0}^\infty \varepsilon_k^2 < \infty$. Then for any initial value ϕ_0 following the updates in (6), we have $\phi_k \to \phi^*$ in probability, where $\phi^* \in \mathbb{R}$ is the unique value such that $C_\nu(\alpha(\phi^*)) = \Gamma$.

Proof. The proof follows from an application of standard stochastic approximation theory to the solution of the root-finding problem $C_\nu(\alpha(\phi)) = \Gamma$. Firstly, by Proposition 3.2, the function $\phi \mapsto C_\nu(\alpha(\phi))$ is continuous and monotonic on $\mathbb{R}$. By the assumption that $C_\nu(1) \geq \Gamma$, it follows that $\phi \mapsto C_\nu(\alpha(\phi))$ is strictly monotonic, and moreover by inspecting the proof of Proposition 3.2, has positive derivative everywhere. By the intermediate value theorem, there exists a unique value $\phi^* \in \mathbb{R}$ such that $C_\nu(\alpha(\phi^*)) = \Gamma$.

Now note that for each $\phi \in \mathbb{R}$, the random variables $\hat{C}^{(k)}(\alpha(\phi))$ are i.i.d. unbiased, bounded estimators of $C_\nu(\alpha(\phi))$. Thus, the scheme (6) is a standard stochastic approximation scheme for the the root of a monotonic function, and the conditions of Theorem 2 of [Robbins and Monro, 1951] are satisfied, enabling us to conclude that $\phi_t \to \phi^*$ in probability, as required. □

Theorem 3.5. Assume the same conditions as Proposition 3.4, and additionally that: (i) trajectory lengths have finite second moment; (ii) immediate rewards are bounded. Let $(\phi_k)_{k=0}^\infty$ be defined as in Equation (6) and $(Q_k)_{k=0}^\infty$ be a sequence of Q-functions, with Q_{k+1} obtained from applying Retrace updates targeting $\alpha(\phi_k)\pi + (1 - \alpha(\phi_k))\mu$ to Q_k with trajectory $k + 1$, using stepsize ε_k. Then we have $\alpha(\phi_k) \to \alpha(\phi^*) =: \alpha^*$ and $Q_k \to Q^{\alpha^*\pi + (1-\alpha^*)\mu}$ almost surely.

Proof. Convergence in probability of $\alpha_k := \alpha(\phi_k)$ to α^* has been shown in Proposition 3.4; it is straightforward to upgrade this to almost-sure convergence using standard stochastic approximation theory. The intuition for the remainder of the proof is that when α_k is close to α^*, the C-trace updates are close to those of standard Retrace targeting the policy $\alpha^*\pi + (1 - \alpha^*)\mu$, which are known to converge under the conditions of the theorem. This is made rigorous by decomposing the update on the Q-function from the $(k + 1)^{\text{th}}$ trajectory as

$$Q_{k+1} = \overbrace{(\mathbf{1} - \widetilde{\varepsilon}_k) \odot Q_k + \widetilde{\varepsilon}_k \odot \mathcal{R}^{\alpha^*} Q_k}^{\text{Desired update}} + \overbrace{(Q_{k+1} - (\mathbf{1} - \widetilde{\varepsilon}_k) \odot Q_k - \widetilde{\varepsilon}_k \odot \mathcal{R}^{\alpha_k} Q_k)}^{\text{Martingale noise}} + \widetilde{\varepsilon}_k \odot \overbrace{(\mathcal{R}^{\alpha_k} Q_k - \mathcal{R}^{\alpha^*} Q_k)}^{\text{Perturbation}},$$

where $\mathcal{R}^\alpha$ denotes the Retrace operator targeting $\alpha\pi + (1-\alpha)\mu$, and with $\widetilde{\varepsilon}_k(x, a) = \varepsilon_k \mathbb{E}[\sum_t \mathbb{1}_{(x_t, a_t) = (x, a)} | (x_0, a_0) = (x, a)]$, and $\odot$ the Hadamard product and $\mathbf{1}$ the vector of 1's. It is then possible to appeal to Proposition 4.5 of Bertsekas and Tsitsiklis [1996] that $Q_k \to Q^{\alpha^*\pi + (1-\alpha^*)\mu}$ almost surely, using the assumptions of theorem. □

B Additional results

B.1 Operators for time-inhomogeneous policies

In this section, we provide a result which rigorously proves the connection between the n-step uncorrected target and the time-inhomogeneous policy mentioned in Section 2.

Proposition B.1. The n-step uncorrected update corresponding to the target

$$\sum_{s=0}^{n-1} \gamma^s r_s + \gamma^n \mathbb{E}_{A\sim\pi(\cdot|x_n)}\left[\hat{Q}(x_n, A)\right],$$

with the trajectory generated under μ, is a stochastic approximation to the operator $(T^\mu)^{n-1}T^\pi$, with fixed point given by the Q-function for the time-inhomogeneous policy which follows π at timesteps t satisfying $t \equiv n-1 \bmod n$, and μ otherwise.

Proof. We begin by taking the expectation of the update target conditional on the initial state-action pair, and showing that it is equal to $((T^\mu)^{n-1}T^\pi\hat{Q})(x_0, a_0)$. We proceed by induction. In the case $n = 1$, the expectation of the update is given by

$$\begin{aligned}
&\mathbb{E}_\mu\left[R(X_0, A_0) + \gamma\mathbb{E}_{A\sim\pi(\cdot|X_1)}\left[\hat{Q}(X_1, A)\right]\middle|X_0 = x_0, A_0 = a_0\right]\\
=&r(x_0, a_0) + \sum_{x'\in\mathcal{X}} P(x'|x, a)\gamma \sum_{a'\in\mathcal{A}} \pi(a'|x')\hat{Q}(x', a')\\
=&(T^\pi\hat{Q})(x_0, a_0),
\end{aligned}$$

as required. For the inductive step, we assume the result holds for some $n \geq 1$. Now observe that by conditioning on (X_1, A_1), we have

$$\begin{aligned}
&\mathbb{E}_\mu\left[\sum_{s=0}^{n}\gamma^s R(X_s, A_s) + \gamma^{n+1}\mathbb{E}_{A\sim\pi(\cdot|Xn)}\left[\hat{Q}(X_{n+1}, A)\right]\middle|X_0 = x_0, A_0 = a_0\right]\\
=&r(x_0, a_0) + \gamma\sum_{x_1\in\mathcal{X}} P(x_1|x_0, a_0)\sum_{a_1\in\mathcal{A}}\mu(a_1|x_1)\times\\
&\qquad\mathbb{E}\left[\sum_{s=1}^{n}\gamma^{s-1}R(X_s, A_s) + \gamma^{n+1}\mathbb{E}_{A\sim\pi(\cdot|Xn)}\left[\hat{Q}(X_{n+1}, A)\right]\middle|X_1 = x_1, A_1 = a_1\right]\\
\overset{(a)}{=}&r(x_0, a_0) + \gamma\sum_{x_1\in\mathcal{X}} P(x_1|x_0, a_0)\sum_{a_1\in\mathcal{A}}\mu(a_1|x_1)((T^\mu)^{n-1}T^\pi\hat{Q})(x_1, a_1)\\
=&(T^\mu(T^\mu)^{n-1}T^\pi\hat{Q})(x_0, a_0)\\
=&((T^\mu)^nT^\pi\hat{Q})(x_0, a_0),
\end{aligned}$$

as required, with (a) following from the induction hypothesis. Finally, for the interpretation of the fixed point of $(T^\mu)^{n-1}T^\pi$, observe that the time-inhomogeneous policy described in the statement of the proposition, which we denote $\pi\mu^{n-1}$ follows a stream of Markovian policies with period n, so it is possible to write down an n-step Bellman equation for its Q-function $Q^{\pi\mu^{n-1}}$. Doing so yields

$$\begin{aligned}
Q^{\pi\mu^{n-1}}(x, a) &= \mathbb{E}_{\substack{A_{1:n-1}\sim\mu(\cdot|X_{1:n-1})\\A_n\sim\pi(\cdot|X_n)}}\left[\sum_{s=0}^{n-1}\gamma^s R(X_s, A_s) + \gamma^n Q^{\pi\mu^{n-1}}(X_n, A_n)\middle|X_0 = x, A_0 = a\right]\\
&= \mathbb{E}_\mu\left[\sum_{s=0}^{n-1}\gamma^s R(X_s, A_s) + \gamma^n\mathbb{E}_{A_n\sim\pi(\cdot|X_n)}\left[Q^{\pi\mu^{n-1}}(X_n, A_n)\right]\middle|X_0 = x, A_0 = a\right].
\end{aligned}$$

We recognise the right-hand side as the operator $(T^\mu)^{n-1}T^\pi$, and thus $Q^{\pi\mu^{n-1}}$ is the fixed point of this operator. □

B.2 Further decompositions of evaluation error

In addition to the decomposition given in Proposition 2.1, there are decompositions of evaluation error based on other norms that may be of interest. We state one such decomposition below, and also note that there is also scope to use different norms to define the fundamental traded-off quantities, such as using the L^∞ norm to define an alternative notion of fixed-point bias, that lead to further decompositions.

Proposition B.2. Consider the task of evaluation of a policy π under behaviour μ, and consider an update rule $\hat{T}$ which stochastically approximates the application of an operator $\widetilde{T}$, with contraction rate Γ and fixed point $\widetilde{Q}$, to an initial estimate Q. Then we have the following decomposition:

$$\mathbb{E}\left[\|\hat{T}Q - Q^\pi\|_2^2\right] \leq 3\left[\underbrace{\mathbb{E}\left[\|\hat{T}Q - TQ\|_2^2\right]}_{\text{Variance}} + \underbrace{\Gamma^2|\mathcal{X}||\mathcal{A}|\|Q - \widetilde{Q}\|_\infty^2}_{\text{(Squared) contraction}} + \underbrace{\|\widetilde{Q} - Q^\pi\|_2^2}_{\text{(Squared) fixed-point bias}}\right].$$

Proof. The inequality is obtained in a manner analogous to that of Proposition 2.1. First, a Cauchy-Schwarz-style argument yields

$$\mathbb{E}\left[\|\hat{T}Q - Q^\pi\|_2^2\right] \leq 3\left[\mathbb{E}\left[\|\hat{T}Q - \widetilde{T}Q\|_2^2\right] + \|\widetilde{T}Q - \widetilde{Q}\|_2^2 + \|\widetilde{Q} - Q^\pi\|_2^2\right].$$

Then, the inequality $\|\cdot\|_2 \leq |\mathcal{X}||\mathcal{A}|\|\cdot\|_\infty$ is applied, together with the definition of T as a contraction mapping under $\|\cdot\|_\infty$ with contraction modulus Γ, to yield the statement. □

C Experimental details

C.1 Environments

Dirichlet-Uniform random MDPs. These random MDPs are specified by two parameteters: the number of states, n_s, and the number of actions, n_a. Transition distributions $P(\cdot|x,a)$ are sampled i.i.d. from a Dirchlet$(1,\ldots,1)$ distribution for each $\mathcal{X}\times\mathcal{A}$. Each immediate reward distribution is given by a Dirac delta, with locations drawn i.i.d. from the Uniform$([-1,1])$ distribution.

Garnet MDPs. Garnet MDPs [Archibald et al., 1995, Piot et al., 2014, Bhatnagar et al., 2009, Geist and Scherrer, 2014] are drawn from a distribution specified by three numbers: the number of states, n_s, the number of actions, n_a, and the *branching factor*, n_b. Each transition distribution $P(\cdot|x,a)$ is given by $n_b^{-1}\sum_{i=1}^{n_b}\delta_{z_i(x,a)}$, where $z_{1:n_b}(x,a)$ are drawn uniformly without replacement from the set of states of the MDP, independently for each state-action pair $(x,a)\in\mathcal{X}\times\mathcal{A}$. $\lfloor n_s/10\rfloor$ states are selected uniformly without replacement, such that any transition out of these states yields a reward of 1, whilst all other transitions in the MDP yield a reward of 0.

Chain MDP. Our chain MDP is specified by a number of states n_s, identified with the set $\{1,\ldots,n_s\}$. State n_s is terminal. Two actions, `left` and `right`, are available at each state, which deterministically move the agent into the corresponding state (taking action `left` in state 1 causes the agent to remain in state 1). Every transition caused by the action `right` incurs a reward of -1, unless the transition is into state n_s, in which case a reward of 50 is received.

C.2 Additional details for plots appearing in the main paper

Figure 1. We use a Dirichlet-Uniform random MDP (see Section C.1) with 5 states and 3 actions. The target π and behaviour μ policies were sampled independently, so that each distribution $\pi(\cdot|x)$ and $\mu(\cdot|x)$ are independent draws from the Dirichlet$(1,\ldots,1)$ distribution. We use a discount rate of 0.9, and a uniform initial state distribution. The variance variable is estimated by simulating 5000 trajectories of length 100, from an initial Q-function estimate set to 0.

Figure 3. In both tasks, the environment is the chain described in Section C.1 with $n_s = 20$. In both tasks, all learning algorithms use a learning rate of 0.1, and the discount factor is set to 0.9 throughout. In the control task, policy improvement is interleaved with 100 steps of environment experience, which are used by the relevant evaluation algorithm. All Retrace-derived methods use $\lambda = 1$. In both evaluation and control tasks, the experiments were repeated 200 times to estimate the standard error by bootstrapping, which is indicated in the plots by the shaded regions.

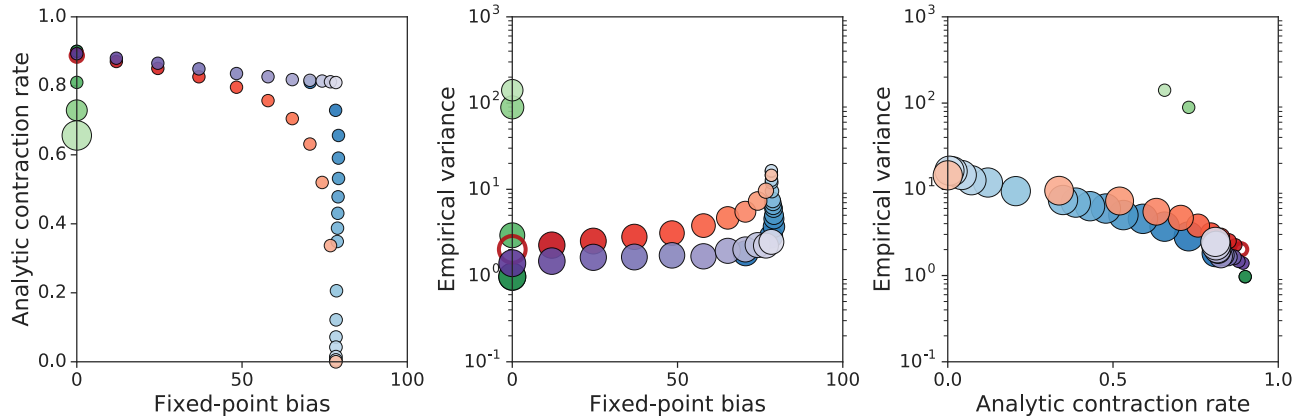


Figure 5: Trade-offs made by n-step uncorrected methods ($n = 1$ (light blue) through to $n = 50$ (dark blue), n-step importance-weighted methods ($n = 1$ (dark green) through to $n = 4$ (dark green), α-Retrace ($\alpha = 1$ (dark red) through to $\alpha = 0$ (light red)), and α-TreeBackup ($\alpha = 1$ (dark purple) through to $\alpha = 0$ (light purple)). Results are shown for the chain environment, and evaluation of a Dirichlet$(1, \ldots, 1)$ policy under behaviour generated by an independently sampled Dirichlet$(1, \ldots, 1)$ policy.

D Further experimental results

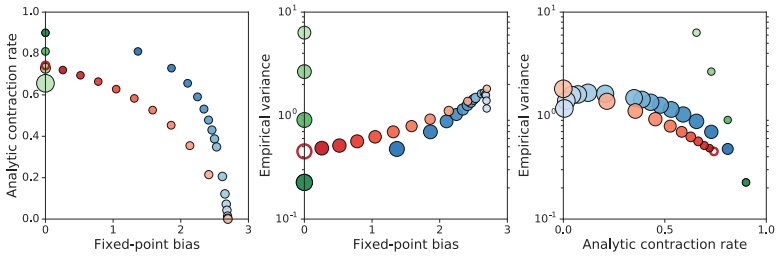
D.1 Further trade-off plots

In this section, we give several further examples of trade-offs made by off-policy algorithms. We begin by examining the trade-offs made by TreeBackup, for which the update target (for a target policy π given a trajectory generated according to a behaviour policy μ) is stated below for completeness.

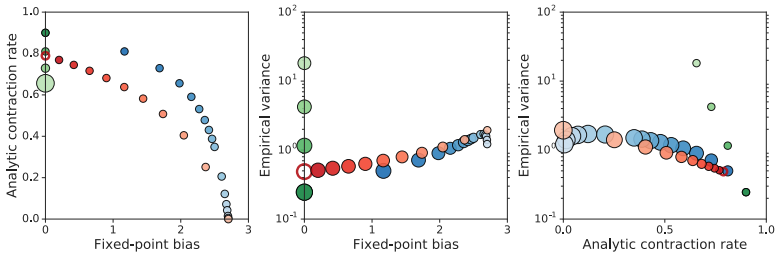
$$\hat{Q}(x_0, a_0) + \sum_{s \geq 0} \gamma^s \prod_{u=1}^{s} \pi(a_u | x_u) \left(r_s + \gamma \mathbb{E}_{A \sim \pi(\cdot | x_{s+1})} \left[\hat{Q}(x_{s+1}, A) \right] - \hat{Q}(x_s, a_s) \right) .$$

We show that mixing in a proportion $1 - \alpha$ of the behaviour policy into the target in TreeBackup (which we dub α-TreeBackup) leads to fundamentally different trade-off behaviour than in α-Retrace; see Figure 5. As can be seen in the plot, mixing in the behaviour policy leads to limited improvements in contraction rate relative to the trade-off achieved by α-Retrace, whilst incurring significant fixed-point bias.

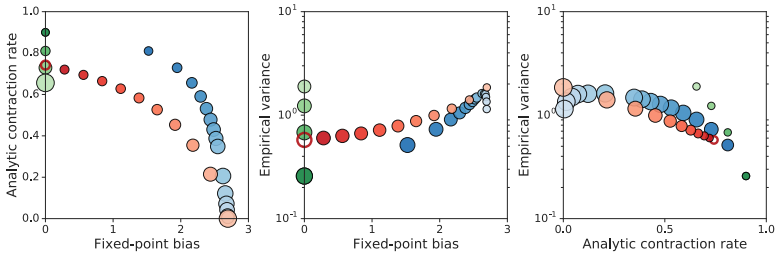
We next demonstrate the robustness of the behaviour exhibited in Figure 1 in a variety of environments, and with a variety of target/behaviour policy pairings. As in Figure 1, α-Retrace is illustrated in red, with dark red corresponding to $\alpha = 1$ through to $\alpha = 0$ in light red. n-step uncorrected methods are illustrated in blue, ranging from $n = 1$ (dark blue) through to $n = 50$ (light blue). n-step importance-weighted methods are illustrated in green, ranging from $n = 1$ (dark green) through to $n = 4$ (light green). Results are given for a Dirichlet-Uniform random MDP (Figure 6), a random garnet MDP (Figure 7), and the chain MDP described in Section C.1 (Figure 8). In all cases, we use a discount rate $\gamma = 0.9$, a learning rate for each algorithm of 0.1, and the variance variable is estimated from 5000 i.i.d. trajectories of length 100. All Retrace methods use $\lambda = 1$ (as presented in the main paper).



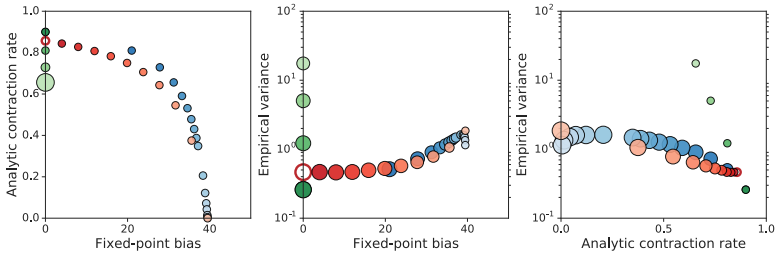
(a) Target policy: uniform. Behaviour policy: Dirichlet$(1, \ldots, 1)$ random.



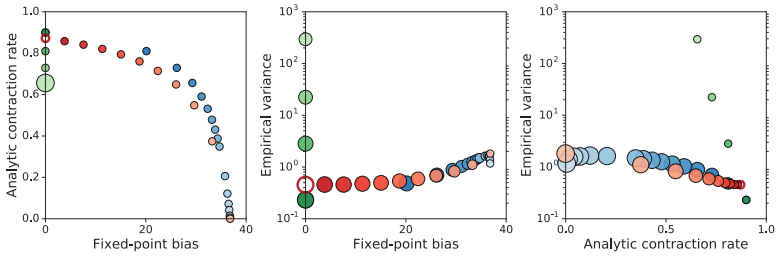
(b) Target policy: Dirichlet$(1, \ldots, 1)$ random. Behaviour policy: Independent Dirichlet$(1, \ldots, 1)$ random.



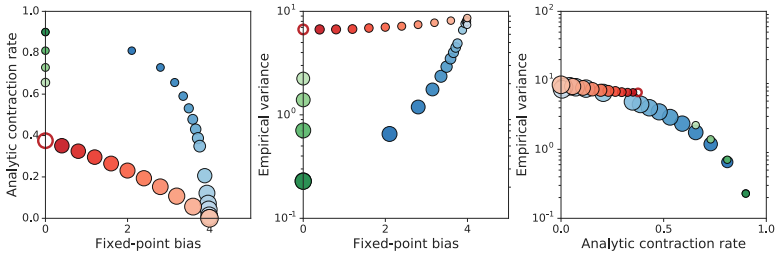
(c) Target policy: Dirichlet$(1, \ldots, 1)$ random. Behaviour policy: uniform.



(d) Target policy: optimal. Behaviour policy: uniform.

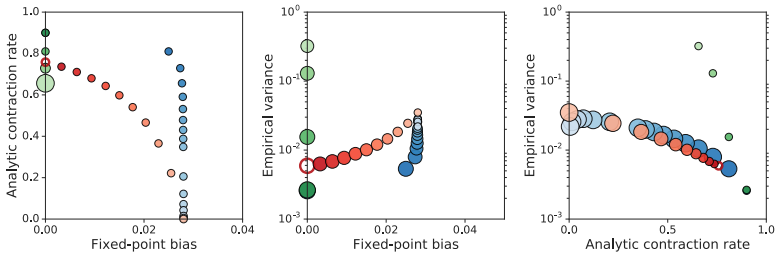


(e) Target policy: optimal. Behaviour policy: Dirichlet$(1, \ldots, 1)$ random.

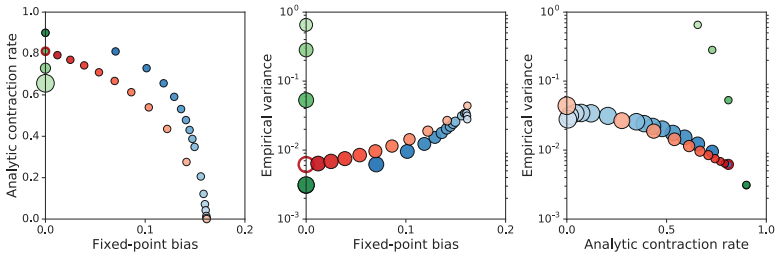


(f) Target policy: optimal. Behaviour policy: optimal, with uniform exploration at probability 0.1.

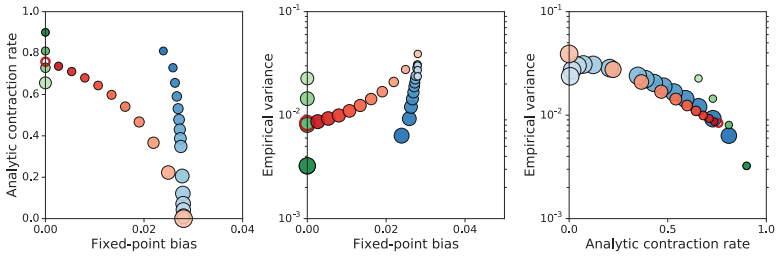
Figure 6: Trade-off plots for a Dirichlet-Uniform random MDP with 20 states and 3 actions, with a variety of target policy/behaviour policy pairings.



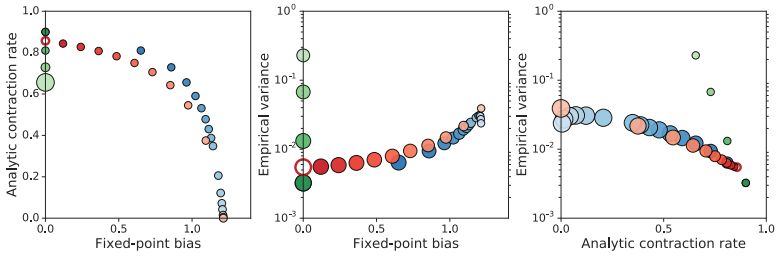
(a) Target policy: uniform. Behaviour policy: Dirichlet$(1, \ldots, 1)$ random.



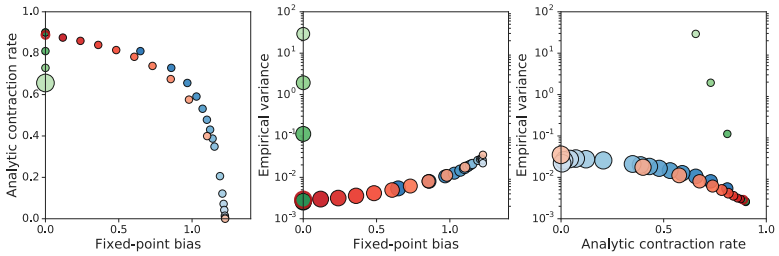
(b) Target policy: Dirichlet$(1, \ldots, 1)$ random. Behaviour policy: Independent Dirichlet$(1, \ldots, 1)$ random.



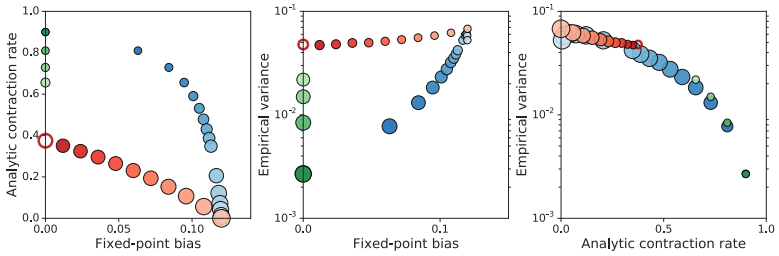
(c) Target policy: Dirichlet$(1, \ldots, 1)$ random. Behaviour policy: uniform.



(d) Target policy: optimal. Behaviour policy: uniform.

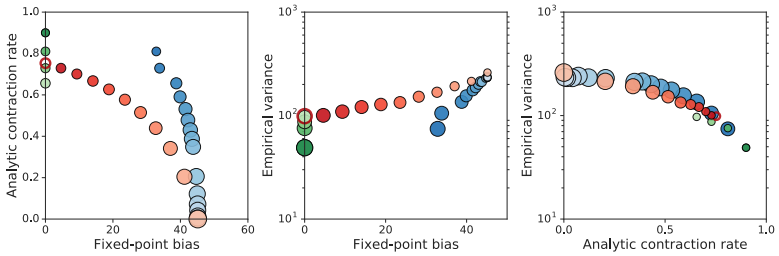


(e) Target policy: optimal. Behaviour policy: Dirichlet$(1, \ldots, 1)$ random.

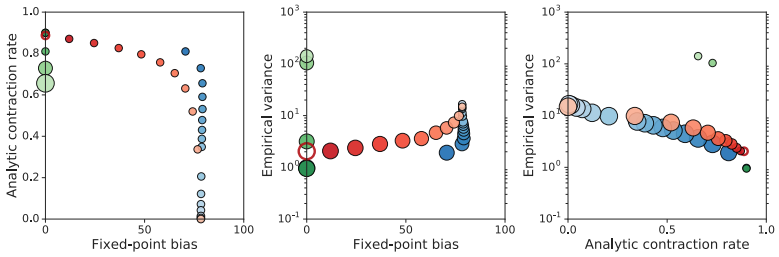


(f) Target policy: optimal. Behaviour policy: optimal, with uniform exploration at probability 0.1.

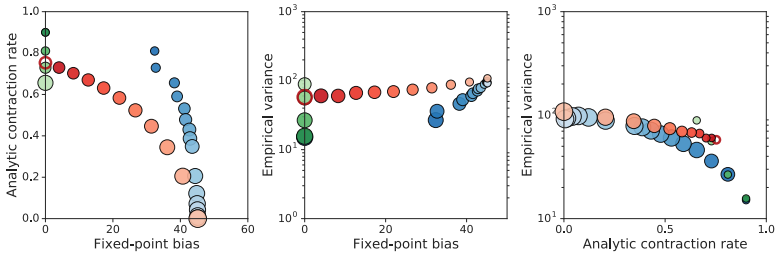
Figure 7: Trade-off plots for a garnet random MDP with 20 states and 3 actions, with a variety of target policy/behaviour policy pairings.



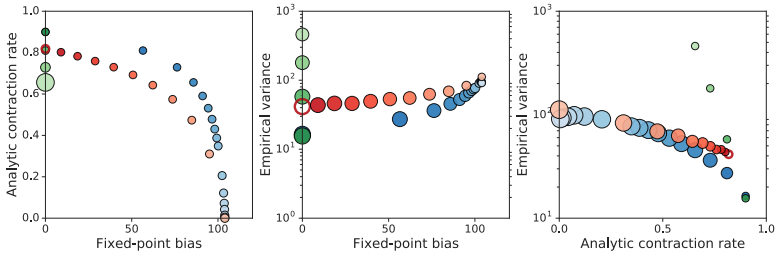
(a) Target policy: uniform. Behaviour policy: Dirichlet$(1, \ldots, 1)$ random.



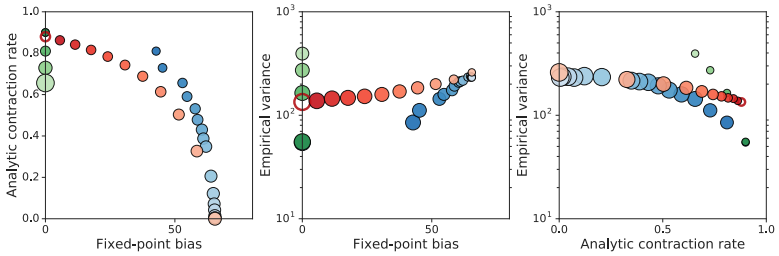
(b) Target policy: Dirichlet$(1, \ldots, 1)$ random. Behaviour policy: Independent Dirichlet$(1, \ldots, 1)$ random.



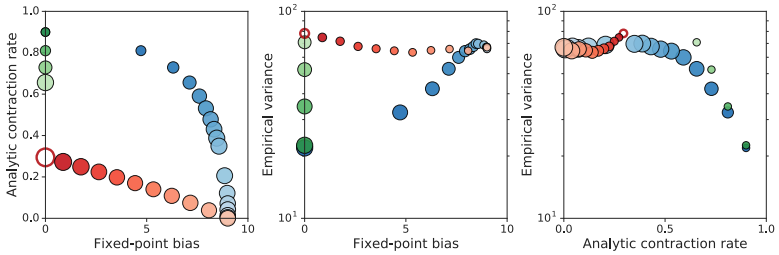
(c) Target policy: Dirichlet$(1, \ldots, 1)$ random. Behaviour policy: uniform.



(d) Target policy: optimal. Behaviour policy: uniform.



(e) Target policy: optimal. Behaviour policy: Dirichlet$(1, \ldots, 1)$ random.



(f) Target policy: optimal. Behaviour policy: optimal, with uniform exploration at probability 0.1.

Figure 8: Trade-off plots for the chain MDP described in Section C.1, with 20 states, with a variety of target policy/behaviour policy pairings.

E Large-scale experiment details

Episodes are limited to 30 minutes (108,000 environment frames). When reporting numeric scores, as opposed to learning curves, we give final agent performance as undiscounted episodic returns. The computing architectures used to run the two agents correspond precisely to the descriptions given in Mnih et al. [2015], Kapturowski et al. [2019].

Mini-batches are drawn from an experience replay buffer as described in the baseline agent papers [Mnih et al., 2015, Kapturowski et al., 2019]. For Retrace and C-trace, the n-step loss function is modified to use the Retrace update for the, possibly modified, target policy. GPU training was performed on an NVIDIA Tesla V100.

Only for R2D2 experiments, all agents (including Retrace-based algorithms) use the invertible value function rescaling of R2D2. Finally, for C-trace, the target policy is given by

$$\hat{\pi} := (1-\alpha)\pi + \alpha\mu,$$

where π is the greedy policy on the current action-values and μ is the ϵ-greedy policy followed by the actor generating the current trajectory. The value of α is adapted with each mini-batch using Robbins-Monro updates with truncated trajectory targets, as described in Section (3.2). The average observed contraction rate of Retrace over the mini-batch is calculated from the Retrace weights (see Equation (5)),

$$\hat{C}(\alpha) = 1 - (1-\gamma)\sum_{t=0}^{N}\gamma^t \prod_{s=1}^{t}\left((1-\alpha) + \alpha\min\left(1, \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)}\right)\right).$$

For simplicity we restate the Robbins-Monro update as a loss, scale up by $1000/(1-\gamma)$ (to counter-act the small learning rate from Adam) and add it to the primary loss. We use a value of $\lambda = 1.0$ for all Retrace and C-trace large-scale experiments. We considered $\lambda = 0.97$, in keeping with published work, but found larger values to perform better overall.

E.1 R2D2 experiments

Network architecture. R2D2, and our Retrace variants, use the 3-layer convolutional network from DQN [Mnih et al., 2015], followed by an LSTM with 512 hidden units, which then feeds into a dueling architecture of size 512 [Wang et al., 2016]. Like the original R2D2, the LSTM receives the reward and one-hot action vector from the previous time step as inputs.

Hyperparameters. The hyperparameters used for the R2D2 agents follow those of Kapturowski et al. [2019], and are reproduced in Table 1 for completeness.

Number of actors	256
Actor parameter update interval	400 environment steps
Sequence length m	80 (+ prefix of $l = 40$ for burn-in)
Replay buffer size	4×10^6 observations (10^5 part-overlapping sequences)
Priority exponent	0.9
Importance sampling exponent	0.6
Discount γ	0.997
Minibatch size	64
Optimiser	Adam [Kingma and Ba, 2015]
Optimiser settings	learning rate $= 10^{-4}$, $\varepsilon = 10^{-3}$
Target network update interval	2500 updates
Value function rescaling	$h(x) = \text{sign}(x)(\sqrt{\lvert x\rvert + 1} - 1) + \epsilon x,\ \epsilon = 10^{-3}$

Table 1: Hyperparameters values used in R2D2 experiments.

E.2 DQN experiments

Network architecture. The DQN, DoubleDQN, n-step and Retrace-based agents use the 3-layer convolutional network from DQN [Mnih et al., 2015], but unlike the R2D2 agents do not use an LSTM or dueling architecture.

Notice that the Retrace and C-trace agents are effectively using DoubleDQN-style updates due to the target probabilities not coming from the target network.

Hyperparameters. For sequential DQN-agents (n-step and Retrace) we performed a preliminary hyperparameter sweep to determine appropriate learning rates for n-step and Retrace updates. We swept over learning rates $(0.00025, 0.0001, 0.00005, 0.00001)$ for both algorithms, and for n-step we jointly swept over two values for n (3 and 5). These were run on four Atari 2600 games (Alien, Amidar, Assault, Asterix), with the best performing hyperparameters for each method used for the Atari-57 experiments.

Interestingly, we found a small learning rate of 0.00001 worked best for both algorithms and that a larger $n = 5$ performed best for n-step.

Both algorithms used a maximum sequence length of 16. Due to shortness of the sequence length we use truncated trajectory corrections as described in the main text. Note that the truncation $\max(\Gamma, \gamma^N)$ is applied to each element of the sequence independently, therefore the value of N will begin at $N = 16$ for the first element and reduce to $N = 1$ for the final transition in the replay sequence.

F Further large-scale results

F.1 Detailed R2D2 results

We give further experimental results to complement the summary presented in the main paper; per-game training curves are given in Figure 9.

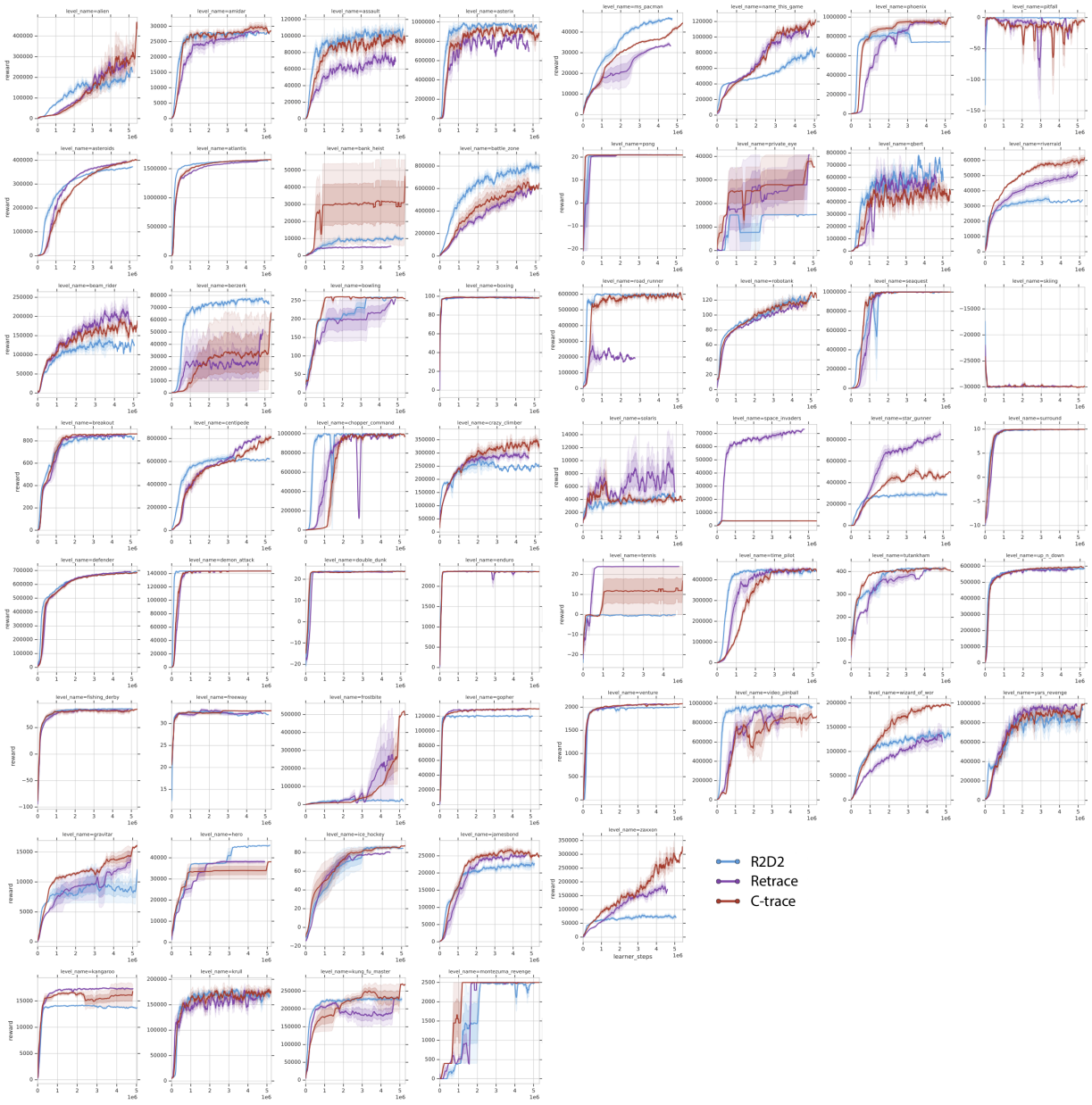


Figure 9: Training curves for 57 Atari games for R2D2 with n-step uncorrected returns (light blue), Retrace-R2D2 (black), and C-trace-R2D2 (red).

F.2 Detailed DQN results

We give further experimental results to complement the summary presented in the main paper. Results for varying the contraction hyperparameter are given in Figure 10, and per-game training curves for the main paper results are given in Figure 11.

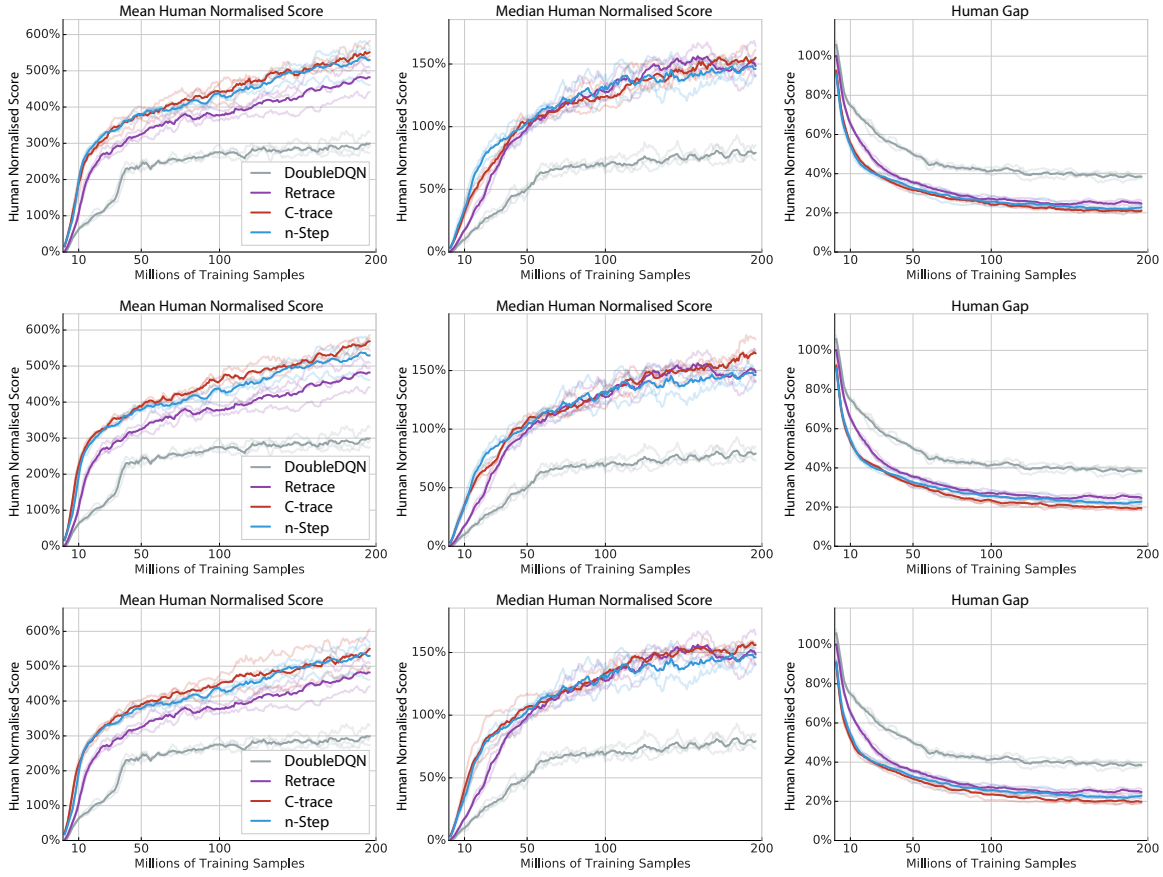


Figure 10: Atari-57 results for single-actor agent, as presented in the main text, but varying the C-trace contraction parameter: (**top**) γ^5, (**centre**) γ^7, and (**bottom**) γ^{10}. Notice that due to its adaptation of α, C-trace is highly robust to the choice of contraction target.

F.3 Empirical verification of C-trace contraction rate

In this section, we provide additional analysis of the empirical R2D2 results described in the main paper, to investigate whether C-trace is able to target the desired contraction rate in practice when used in combination with a deep reinforcement learning architecture. We compute averaged contraction rates using Equation (5), and in Figure 12 we provide kernel density estimation (KDE) plots of these contraction rates achieved by C-trace and Retrace at the end of training across two classes of games. Firstly, those games for which Retrace achieves a contraction rate of more than γ^{10}, the chosen target rate for C-trace in this instance, and secondly, the complement of these games. Splitting the games into these two classes illustrates the behaviour of C-trace clearly. In the first class of games, it is possible for C-trace to attain the contraction rate γ^{10}, by lowering the mixture parameter, whilst in the second class, Retrace already has a contraction rate below the target of γ^{10}. This description is reflected in the plots; in the left plot of Figure 12, the Retrace distribution lies to the right of the target contraction rate, whilst the C-trace distribution is centred precisely on this rate, whilst on the right, the Retrace distribution lies to the left of the target contraction rate, and the C-trace distribution closely matches it, as there is no possibility of increasing the target contraction rate. We also provide per-game KDE plots of contraction rates attained throughout training in Figure 13.

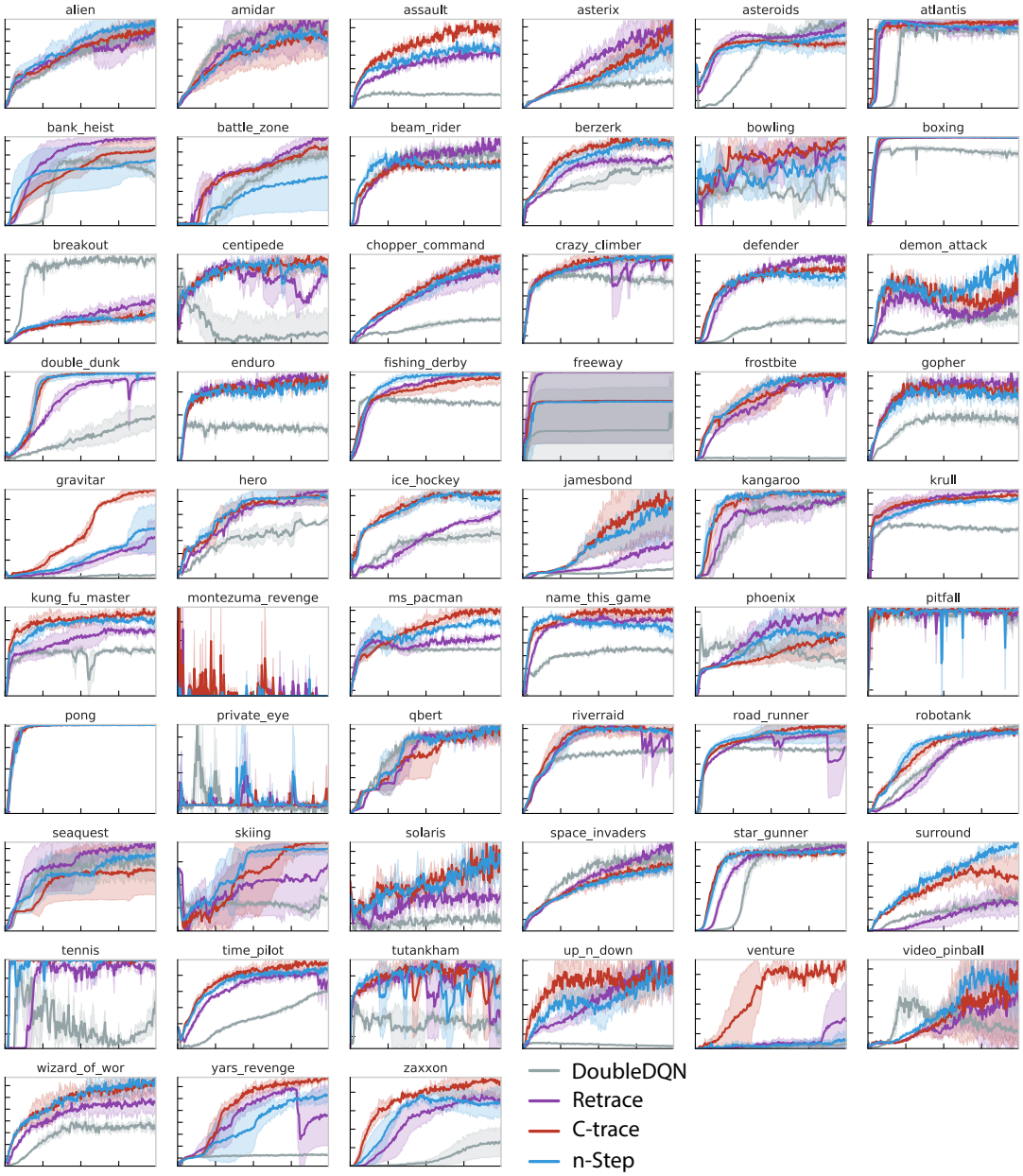


Figure 11: Training curves for 57 Atari games for Double DQN (grey), Double DQN with n-step uncorrected returns (light blue), Retrace-DQN (black), and C-trace-DQN (red).

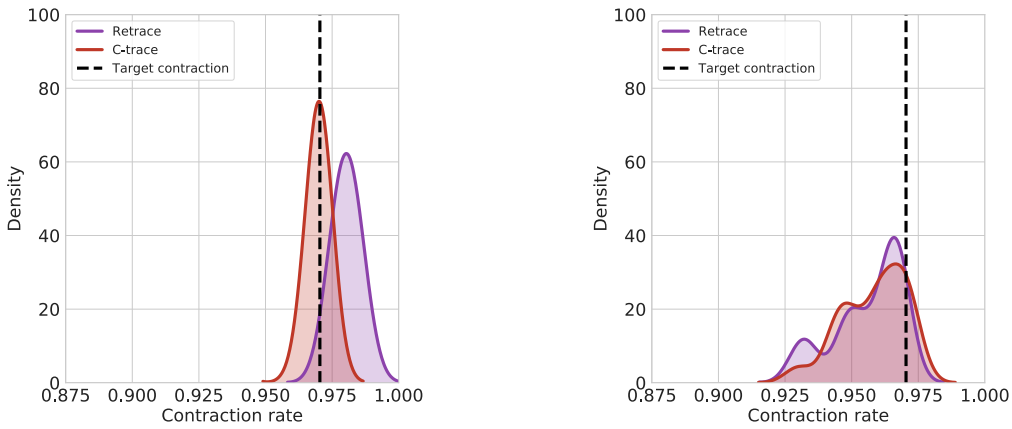


Figure 12: Contraction rates achieved by Retrace and C-trace at the end of training, across two classes of games. Left: games in which the contraction rate of Retrace is greater than the C-trace target of γ^{10}. Right: games for which the contraction rate of Retrace is less than the C-trace target of γ^{10}.

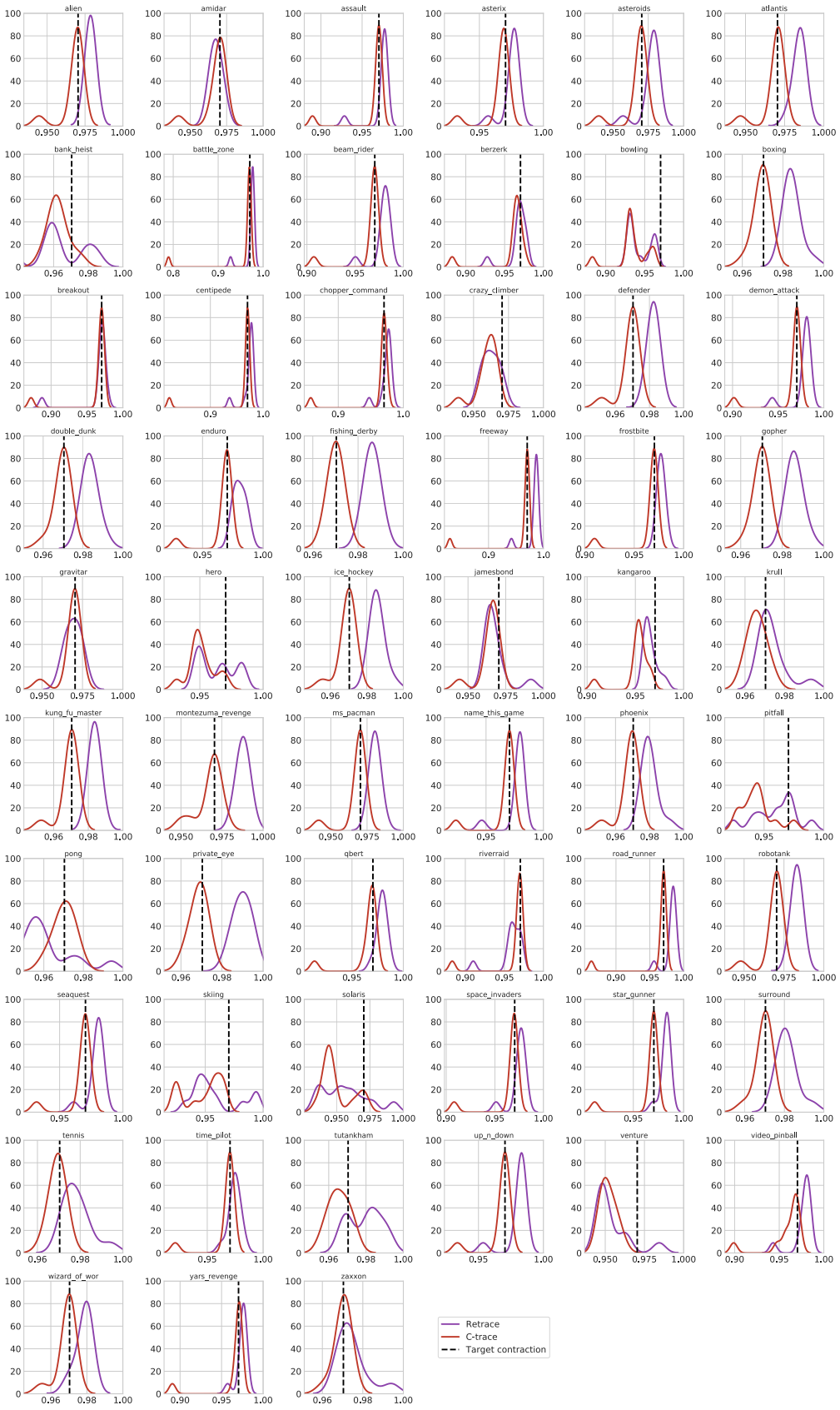


Figure 13: Per-game contraction rates for Retrace and C-trace throughout training.