CONTROLLING THE OVERESTIMATION BIAS OF Q-LEARNING FROM AN ACTION-SELECTION AND ACTION-EVALUATION OVERLAP PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

The well-known estimation bias causes Q-value estimates to be overestimated or underestimated, potentially impairing learning and policy performance. This bias is closely related to the sharing of Q-functions between action selection and value evaluation. However, existing methods lack a systematic analysis of how the sharing degree affects estimation bias, providing limited theoretical guidance for bias minimization across different settings. We formalize the sharing degree as the overlap between the Q-functions used for action selection and value evaluation, and analyze how this overlap affects estimation bias and should be chosen to minimize bias. Under the stated assumptions, we prove that the signed estimation bias is a monotonic function of this overlap, and that the bias-minimizing overlap is governed by the signal-to-noise ratio (SNR). Motivated by this finding, we propose Selective-Overlap Q-learning (SOQ). SOQ uses all Q-functions for value estimation and selects a Q-function subset for action selection according to a state-dependent proxy for SNR, aiming to approximate the bias-minimizing overlap and reduce estimation bias. By adapting the overlap between action selection and value evaluation, SOQ improves the reliability of Q-value estimation across different settings.

1 INTRODUCTION

Q-learning Watkins & Dayan (1992) and its variants Chen et al. (2021b); Wu et al. (2022); Abliz (2024) have achieved remarkable success in Markov Decision Processes (MDPs), such as recommendation systems Li et al. (2024), UAV control Farid et al. (2025), robot control Gao (2024) and so on Chu et al. (2022); Zhao et al. (2023); Zhang et al. (2023). At each update, Q-learning estimates the current state-action pair using the immediate reward and the evaluated Q-value of the best action in the next state. This update can be viewed as two coupled operations: action selection and action evaluation. In standard Q-learning, the same Q-function is used for both operations. Because the same noisy estimates determine and evaluate the maximizing action, this coupling can induce positive selection bias. Such bias may degrade learning, particularly when estimation errors are amplified by function approximation Melo (2001). Double Q-learning Thrun & Schwartz (2014); Hasselt (2010) mitigates this issue by decoupling action selection from action evaluation with two independent Q-functions. However, this decoupling often leads to underestimation bias. Although underestimation avoids overly optimistic value estimates, excessive underestimation can still slow down convergence and reduce sample efficiency.

Numerous approaches have been proposed to mitigate estimation bias. Methods such as Weighted Double Q-learning (Zhang et al., 2017) and ADDQ (Döring et al., 2025) balance overestimation and underestimation through weighted combinations of value estimates. Ensemble-based methods, including Maxmin Q-learning (Lan et al., 2020), AdaEQ (Wang et al., 2021), REDQ (Chen et al., 2021a), and Order Q-learning (Tan et al., 2024), regulate estimation bias through ensemble aggregation, ensemble size, or the choice of order statistic. A practical challenge is that action-value gaps and estimation noise vary across states and throughout learning, changing the balance between overestimation and underestimation. Effective bias control therefore requires adapting to the current estimation conditions. This motivates identifying a control variable that directly governs estimation bias and an observable signal that guides its adjustment.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Example 1. *Consider a single-state, 10-action bandit with self-loop transitions, where every action returns a reward $r \sim \mathcal{N}(0, 1)$. With $\gamma = 0.95$, the optimal expected Q-value is 0. We compare fully coupled Q-learning and Ensemble Q-learning Anschel et al. (2017), decoupled Double Q-learning and EBQL Peer et al. (2021), and overlap-based OQ and SOQ. OQ(M, N) uses M Q-tables for action evaluation and N for action selection, while SOQ adapts N; the ensemble configurations use four Q-tables. Figure 1 shows that the bias shifts from underestimation toward overestimation as N increases. Among the evaluated configurations, SOQ yields the mean closest to zero after 10,000 environment steps.*

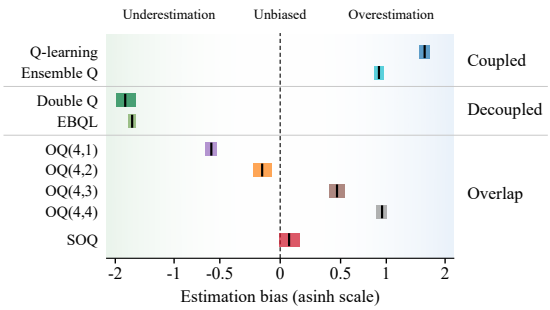


Figure 1: Estimation bias under different overlap conditions. Black marks and colored intervals denote means and 95% confidence intervals, respectively, over 100 independent runs after 10,000 environment steps.

In this paper, we provide a unified view of estimation bias through the degree of sharing between the Q-functions used for action selection and action evaluation. When the shared Q-function is used for both operations, as in standard Q-learning, the estimator tends to suffer from overestimation. When independent Q-functions are used, as in Double Q-learning, the estimator tends to suffer from underestimation. This observation suggests that overestimation and underestimation are not separate phenomena, but two outcomes induced by different degrees of sharing between action selection and action evaluation. To formalize this idea, we measure the sharing degree by the overlap between the Q-functions used for action selection and action evaluation. Based on this formulation, we analyze how estimation bias varies with the overlap. Under a stylized two-action Gaussian error model, the signed bias increases monotonically with overlap. Increasing the overlap shifts the estimator toward overestimation, whereas decreasing it shifts the estimator toward underestimation. Example 1 illustrates this bias transition in a simple self-loop bandit. The corresponding zero-bias overlap varies with the signal-to-noise ratio (SNR), providing a direction for adapting the overlap across estimation settings.

Motivated by these findings, we propose Selective-Overlap Q-learning (SOQ). SOQ retains all Q-functions for action evaluation and adapts the action-selection subset using an empirical signal coverage ratio (SCR). This observable proxy for action-ranking uncertainty guides the overlap toward the bias-minimizing regime as estimation conditions vary across states and throughout learning. Experiments on tabular tasks and Atari games demonstrate reduced estimation bias and improved learning performance, supporting adaptive overlap as a practical approach to controlling Q-value estimation bias.

2 Related Works

Methods that use the same noisy Q-function for maximizing action selection and evaluation are susceptible to overestimation bias. Variants such as Averaged Q-learning Anschel et al. (2017), ScQ Zhu & Rigotti (2021) and softmax-based operators Asadi & Littman (2017); Song et al. (2019); Gan et al. (2021) attempt to mitigate this effect by smoothing or aggregating value estimates. However, these methods retain coupling between action selection and evaluation and therefore do not fully remove the mechanism that can produce overestimation. In contrast, Double Q-learning Hasselt (2010) reduces overestimation by evaluating the selected action with a separate estimator. This decoupling removes the direct reuse of the selected estimation error, but can lead to underestimation when noise causes the selector to choose a suboptimal action Abliz & Ying (2022). Ensemble extensions such as EBQL Peer et al. (2021) preserve this decoupled selection–evaluation structure and can inherit the same suboptimal-selection mechanism. Methods such as Doubly Bounded Q-learning Ren et al. (2021) introduce auxiliary estimators to alleviate the resulting bias.

To overcome these inherent biases, subsequent work focuses on designing mechanisms to control estimation bias. Weighted DQ Zhang et al. (2017) performs linear weighting between Q and DQ to correct for overestimation and underestimation. Maxmin Q-learning Lan et al. (2020) instead estimates the maximum expected Q-value using the minimum over multiple Q-functions. REDQ Chen et al. (2021a) select a subset from multiple Q-functions to serve as Q-estimators in order to control bias. AdaEQ Wang et al. (2021) further extends this framework by adaptively selecting the number of Q-functions in an attempt to mitigate estimation bias. Order Q-learning Tan et al. (2024) generalizes ensemble-based methods by constructing an order-statistic Q-estimator over multiple Q-functions, enabling more flexible and fine-grained bias control. AC-CDQ Jiang et al. (2021) attempts to balance overestimation and underestimation through a clipping-based action candidate selection strategy. ADDQ Döring et al. (2025) uses distributed Q-learning to estimate state uncertainty in order to balance overestimation and underestimation. However, these methods control bias using some fixed global settings or heuristic adjustments, and their performance is often sensitive to hyperparameter choices across different environments.

Different from these methods, we characterize estimation bias through the overlap between the Q-functions used for action selection and action evaluation. Under the stylized Gaussian error model analyzed in Section 3, the signed bias is monotonic in the overlap size and its zero-bias point depends on the signal-to-noise ratio (SNR). The resulting chain connects overlap as a structural control variable, SNR as theoretical guidance for its adjustment, and SCR as an observable signal for statewise adaptation.

3 Theoretical Analysis of Selective Overlap

3.1 Overlap Estimator and Bias Characterization

We formalize the overlap in a statistical estimation setting and, under the assumptions stated below, show that the signed bias changes monotonically with overlap size. Consider the problem of estimating the maximal expected value $\mu^* \triangleq \max_a \mathbb{E}[X_a]$, where satisfies $X_a \sim \mathcal{N}(\mu_a, \sigma^2)$. For each action a, let $S_a = \{S_a(i)\}_{i=1}^M$ denote M i.i.d. samples of X_a, and let $\hat{\mu}_a(S_a)$ be the empirical mean computed from S_a. We first recall two extreme estimators. The maximum estimator (ME) uses the same samples for action selection and action evaluation:

$$\hat{a}^* = \arg\max_a \hat{\mu}_a(S_a), \quad \hat{\mu}^*_{ME} = \hat{\mu}_{\hat{a}^*}(S_{\hat{a}^*}).$$

Because the selected action is also evaluated by the same samples, ME positively selects estimation errors and therefore tends to overestimate.

In contrast, the cross-validation estimator (CE) selects the action with one subset and evaluates it with an independent subset:

$$\hat{a}^* = \arg\max_a \hat{\mu}_a(S_a^{(1)}), \quad \hat{\mu}^*_{CE} = \hat{\mu}_{\hat{a}^*}(S_{\hat{a}^*}^{(2)}).$$

This decoupling removes positive selection bias, but can underestimate when a suboptimal action is selected due to sampling noise. ME and CE thus represent the two extremes of sample overlap between action selection and action evaluation, with complete overlap in ME and none in CE.

To formalize this idea, we introduce the *Overlap Estimator* (OE). Instead of using all samples for action selection, OE randomly selects a subset $S_a^N \subset S_a$, where N is the overlap size (i.e., $|S_a^N| = N$). The selected subset is used for action selection, while all samples are used for action evaluation:

$$\hat{a}^* = \arg\max_a \hat{\mu}_a(S_a^N), \qquad \hat{\mu}^*_{OE} = \hat{\mu}_{\hat{a}^*}(S_{\hat{a}^*}).$$

OE controls the overlap between action selection and action evaluation by using a subset of the evaluation samples for selection. The selection sample size N determines the overlap: when $N = M$, the two sample sets coincide and OE reduces to the maximum estimator; decreasing N reduces the overlap. Under the two-action Gaussian model in Lemma 1, the signed bias is strictly increasing in N, so adjusting the overlap shifts the bias toward overestimation or underestimation. Because the selection samples remain part of the evaluation samples, lower overlap does not imply independence between the two sample sets. We next characterize the estimation bias of OE in Lemma 1.

Lemma 1. *(Proof in Appendix A.1.) Assume that there are two actions, i.e., $m = 2$. Let $X = (X_1, X_2)$, where $X_1 \sim \mathcal{N}(\mu_1, \sigma^2)$, $X_2 \sim \mathcal{N}(\mu_2, \sigma^2)$ with $\mu_1 > \mu_2$. Define $\Delta = \mu_1 - \mu_2$. Given M i.i.d. samples of each X_a, i.e., $S_a = \{S_a(i)\}_{i=1}^{M}$ and randomly sample a subset S_a^N with size N, $S_a^N \subset S_a$. Let the SNR (signal-to-noise ratio) $\varepsilon = \frac{\Delta}{\sigma}$ and $z = \frac{\sqrt{N}\Delta}{\sqrt{2}\,\sigma} = \frac{\sqrt{N}}{\sqrt{2}}\,\varepsilon$. For estimating the maximal expected value of X, the estimation bias of OE is:*

$$\mathrm{Bias}^{\mathrm{OE}} = \frac{\sqrt{2N}}{M} \cdot \sigma\phi(z) - \Delta(1 - \Phi(z)),$$

where $\phi(z)$ and $\Phi(z)$ denote the probability density function and cumulative distribution function of the standard normal distribution, respectively.

Lemma 1 explicitly characterizes estimation bias as a function of the overlap size N, identifying N as a control variable for estimation bias. The following proposition establishes the direction of this control.

Proposition 1. *(Proof in Appendix A.2.) For fixed M, Δ, and σ, viewing the bias in Lemma 1 as a continuous function of $N \in [0, M]$, it is monotonically increasing with respect to the overlap size N. Moreover, the following properties hold: (1) as $N \to 0^+$, the bias is negative, $\mathrm{Bias}^{\mathrm{OE}} \to -\Delta/2 < 0$; (2) when $N = M$, OE has positive bias: $\mathrm{Bias}^{\mathrm{OE}} > 0$; and (3) there exists an intermediate overlap size $N^* \in (0, M)$ such that $\mathrm{Bias}^{\mathrm{OE}} = 0$.*

Proposition 1 establishes the overlap size N as a control variable for estimation bias. Under the assumptions of Lemma 1, increasing N shifts the signed bias from underestimation toward overestimation, with a unique zero-bias point in the continuous extension. This characterizes the direction of bias control and shows how adjusting the overlap can reduce the magnitude of estimation bias. This result provides a unified explanation of overestimation and underestimation through the selection–evaluation structure. ME and CE represent fully coupled selection and evaluation and independent evaluation, respectively, corresponding to the structures used by Q-learning and Double Q-learning. OE recovers ME at $N = M$ and approaches the bias behavior of CE as the evaluation-to-selection sample ratio M/N grows, connecting the two estimation regimes through adjustable overlap.

3.2 SNR-Driven Characterization of the Zero-Bias Overlap

Proposition 1 establishes the existence of a zero-bias overlap in the continuous extension of the Gaussian error model. We next examine how this overlap depends on the signal-to-noise ratio (SNR), $\varepsilon = \Delta/\sigma$, which compares the action-value gap with the standard deviation of estimation noise. A larger SNR makes the optimal action easier to distinguish, whereas a smaller SNR makes action selection more susceptible to noise.

Proposition 2. *(Proof in Appendix A.3.) For fixed M, as $\varepsilon \to 0$, the continuous zero-bias overlap satisfies*

$$N^* = \frac{\pi M^2}{4}\varepsilon^2\big(1 + O(\varepsilon^2)\big). \tag{1}$$

Proposition 2 motivates a smaller overlap at low SNR, while the high-SNR endpoint motivates a larger overlap as action misranking becomes rare. Since SNR is unobservable during learning, we introduce the signal coverage ratio (SCR). SCR measures the average probability that a suboptimal action's estimated value exceeds that of the optimal action. We use this quantity to guide overlap adaptation.

Lemma 2. *(Proof in Appendix A.4.) Let $K = |\mathcal{A}| \geq 2$ and let $\Delta = \mu_{a^*} - \min_{a \in \mathcal{A}} \mu_a > 0$ denote the total action-value gap. Under the independent, equal-variance Gaussian error model of Lemma 1, suppose the action values are equally spaced as $\mu_{a_{(j)}} = \mu_{a^*} - j\Delta/(K-1)$ for $j = 1, \ldots, K-1$. Define the normalized signal coverage ratio as*

$$c \triangleq \frac{1}{K-1} \sum_{a \in \mathcal{A} \setminus \{a^*\}} \Pr\big(Q(s', a) > Q(s', a^*)\big). \tag{2}$$

Then, with the SNR $\varepsilon = \Delta/\sigma$,

$$c = \frac{1}{K-1} \sum_{j=1}^{K-1} \Phi\Big(- \frac{j\varepsilon}{(K-1)\sqrt{2}} \Big). \tag{3}$$

The SCR is strictly decreasing in ε and reduces to $\Phi(-\varepsilon/\sqrt{2})$ when $K = 2$.

Lemma 2 extends the SCR–SNR relation to multiple actions before the quantity is used by the algorithm. Its two limiting values provide the endpoints for adjusting N without requiring the SNR to be recovered explicitly.

Lemma 3. *(Proof in Appendix A.5.) Under the assumptions of Lemma 2, $c \to 1/2$ as $\varepsilon \to 0$.*

Lemma 4. *(Proof in Appendix A.6.) Under the assumptions of Lemma 2, $c \to 0$ as $\varepsilon \to \infty$.*

The endpoint results suggest a monotonic overlap schedule: when the SCR is close to $1/2$ (low SNR), N should be small, whereas an SCR close to 0 (high SNR) permits a larger N. Intermediate SCR values can therefore be mapped to intermediate overlap sizes.

3.3 Selective-Overlap Q-learning

Motivated by Propositions 1 and 2, Selective-Overlap Q-learning (SOQ) retains all Q-functions for action evaluation and uses a subset for action selection. The subset size determines the overlap N and adapts through an empirical multi-action SCR, without estimating SNR explicitly. Figure 2 contrasts fully overlapping, non-overlapping, and selective-overlap estimators. SOQ adjusts the selection subset between full and minimal overlap while retaining the all-M evaluator, controlling the balance between overestimation and underestimation.

Formally, let $Q_t = \{Q_t^i\}_{i=1}^M$ be the M maintained Q-tables and choose $N \in \{1, \dots, M\}$. At the current update, sample an index set $\mathcal{I}_t \subseteq [M]$ uniformly without replacement, with $|\mathcal{I}_t| = N$. The corresponding selector ensemble is

$$Q_{\mathcal{A},t} \triangleq \{Q_t^i : i \in \mathcal{I}_t\} \subseteq Q_t. \tag{4}$$

The agent interacts with the environment using an ε-greedy policy derived from the current Q-values. At each time step, it observes a transition (s, a, r, s', τ), where $s \in \mathcal{S}$ denotes the current state, $a \in \mathcal{A}$ is the selected action, $r \in \mathbb{R}$ is the reward, $s' \in \mathcal{S}$ is the next state, and $\tau \in \{0, 1\}$ indicates whether the transition is terminal. After that, Overlap Q-learning uses the sampled selector ensemble to select the target action:

$$\hat{a}_N \in \arg\max_{a'} \frac{1}{N} \sum_{i \in \mathcal{I}_t} Q_t^i(s', a'). \tag{5}$$

Finally, we update one Q-function $Q^j(s, a)$ uniformly at random:

$$\begin{cases} Y_t^{\mathrm{SOQ}} = r_t + \gamma(1 - \tau_t) \dfrac{1}{M} \displaystyle\sum_{i=1}^{M} Q_t^i(s', \hat{a}_N), \\ Q_{t+1}^j(s, a) = Q_t^j(s, a) + \alpha_t(s, a)[Y_t^{\mathrm{SOQ}} - Q_t^j(s, a)], \end{cases} \tag{6}$$

This design directly implements the selective-overlap estimator in reinforcement learning. The subset $Q_{\mathcal{A},t}$ determines the overlap between action selection and action evaluation. The fixed-N analysis suggests increasing the selector size when action ranking is reliable and decreasing it when ranking is noise-dominated. SOQ operationalizes this direction through the empirical SCR, without observing the latent statewise SNR. The population SCR in Equation (2) is not directly observable because it depends on the unknown optimal action and comparison probabilities. For each target state s', we first define an empirical reference action using the full ensemble average,

$$\bar{a} \triangleq \arg\max_{a \in \mathcal{A}} \frac{1}{M} \sum_{i=1}^{M} Q^i(s', a). \tag{7}$$

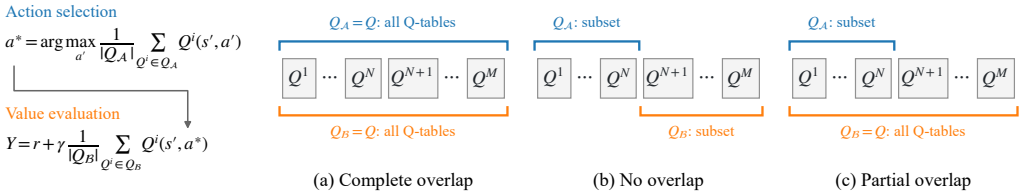


Figure 2: Selection–evaluation overlap. Blue and orange brackets mark the selection and evaluation sets, respectively. In (c), the N selection Q-functions form a subset of the M evaluation Q-functions.

We then form a finite-ensemble proxy for the multi-action SCR using ordered comparisons among the maintained Q-tables:

$$\widehat{c}(s') \triangleq \frac{1}{K-1} \sum_{a' \in \mathcal{A} \setminus \{\bar{a}\}} \left[\frac{1}{M^2} \sum_{i=1}^{M} \sum_{k=1}^{M} \mathbf{1}\big(Q^k(s', a') > Q^i(s', \bar{a})\big) \right]. \tag{8}$$

For each alternative action, the bracketed term is its empirical misranking frequency over all ordered pairs of Q-tables. Thus, Equation (8) directly mirrors the action-wise average in Equation (2). The statistic $\widehat{c}$ replaces both the population probabilities and the unknown action a^* with finite-ensemble counterparts. It therefore provides an observable empirical approximation of c.

Let $\rho = 1/2$ denote the low-SNR endpoint established in Lemma 3, and define $U(s') = \max\{\rho, 1 - \widehat{c}(s')\}$. We use the endpoint-linear controller

$$N = \mathrm{clip}_{[1,M]}\left(\mathrm{round}\left[M \frac{U(s') - \rho}{1 - \rho} \right] \right). \tag{9}$$

Here, $\mathrm{clip}_{[1,M]}$ enforces the legal integer range after rounding. Equation (9) yields $N = M$ at $\widehat{c} = 0$ and $N = 1$ when $\widehat{c} \geq 1 - \rho$; for $\rho = 1/2$, these correspond to the high- and low-SNR endpoints. This endpoint interpolation translates the SNR-dependent direction established above into a state-dependent rule for adjusting the overlap. Combining the adaptive overlap rule with the tabular update yields SOQ. Complete pseudocode is provided in Algorithm 1 in Appendix C.1, alongside the fixed-overlap specialization and deep-network implementations.

Deep instantiation. SOQ-DQN uses an online ensemble and a periodically synchronized target ensemble. For each replay sample, a target-head subset selects the bootstrap action and the all-head target mean evaluates it. Behavior and evaluation policies use the all-head online mean. Complete procedures are given in Appendix C.

3.4 Bellman-Target Bias and Tabular Convergence

Algorithm 1 inserts the statewise selective-overlap estimator into a bootstrapped Q-learning update. Equation (6) makes the two roles explicit: N_t determines the action-selection subset, while all M estimators participate equally in action evaluation. Relative to the full-ensemble greedy target, selective overlap introduces the selection-loss term characterized in Appendix B. Thus, overlap changes a transient component of the bootstrap target, while Theorem 1 separately characterizes the asymptotic tabular limit. The fixed-overlap analysis determines the direction of adaptation. Because N_t is computed from the same ensemble realization used for action selection and evaluation, however, the fixed-N formula is not an exact finite-sample bias characterization of the adaptive controller.

Proposition 3 (Propagation of Bellman-target perturbations)**.** *Let $q_{k+1} = \mathcal{T}^\star q_k + \delta_k$ be an associated deterministic or mean-operator recursion, where $\mathcal{T}^\star$ is the γ-contractive Bellman optimality operator and δ_k is the one-step target discrepancy induced by the estimator. If $\|\delta_k\|_\infty \leq \eta$ for all k, then*

$$\limsup_{k \to \infty} \|q_k - Q^\star\|_\infty \leq \frac{\eta}{1 - \gamma}. \tag{10}$$

If $\|\delta_k\|_\infty \to 0$, then $q_k \to Q^\star$.

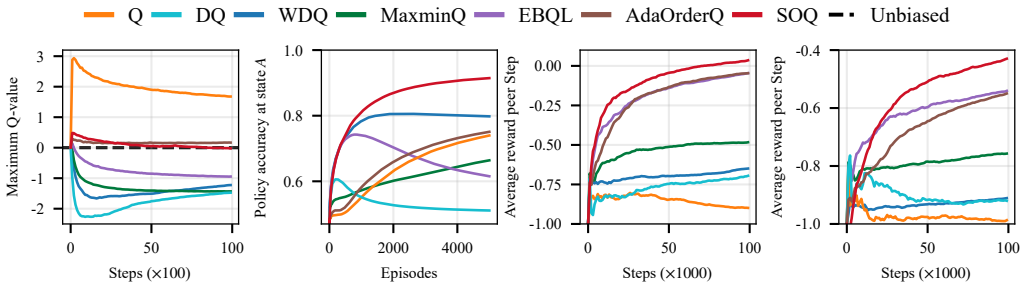
Theorem 1 (Tabular convergence of adaptive SOQ)**.** *Consider a finite MDP with bounded rewards and $\gamma < 1$. Suppose every state–action pair x is visited infinitely often. Along the local visit clock of x, let the gains $\alpha_n(x) \in [0,1]$ satisfy $\sum_n \alpha_n(x) = \infty$ and $\sum_n \alpha_n^2(x) < \infty$. After observing each transition, the controller may choose any legal $N_t \in \{1,\dots,M\}$ and selector set based on the current ensemble, that transition, and its own randomization. After the common target is constructed, the update index is sampled conditionally uniformly from $[M]$ and independently of that target. Then Algorithm 1 satisfies*

$$\max_{i\in[M]} \|Q_t^i - Q^\star\|_\infty \longrightarrow 0 \qquad \textit{almost surely.} \tag{11}$$

Theorem 1 identifies a common, N-independent tabular limit. Its proof first establishes consensus among the maintained estimators and then reduces their mean to asynchronous Q-learning with a vanishing target perturbation. Thus, N_t need not converge. The result establishes invariance of the tabular limit to adaptive overlap, not finite-time optimality of the controller or convergence with nonlinear function approximation. The target decomposition and proof are given in Appendix B.

4 Tabular Experiments

We evaluate SOQ through two sets of tabular experiments: comparisons with existing methods on estimation bias and control performance, and an assessment of bias across signal and noise levels. Detailed settings are reported in Appendix D.



(a) Multi-Armed bandit (b) MDP (c) Gridworld 3×3 (d) Gridworld 4×4

Figure 3: Tabular performance over 100 independent runs per setting: (a) mean maximum Q-value in the bandit, with zero indicating unbiased estimation; (b) cumulative optimal-action selection rate in the Meta Chain MDP, averaged over six settings; (c,d) mean reward per step in the two Gridworld environments.

Comparison of different Methods. We compare SOQ with Q-learning (Q), Double Q-learning (DQ), Weighted Double Q-learning (WDQ) (Zhang et al., 2017), Maxmin Q-learning (MaxminQ) (Lan et al., 2020), Ensemble Bootstrapping Q-learning (EBQL) (Peer et al., 2021), and Adaptive Order Q-learning (AdaOrderQ) (Tan et al., 2024). We follow the hyperparameter recommendations in the original papers, while fixing the ensemble size to $M = 8$ for ensemble-based methods. Detailed settings are provided in Appendix D.

We use a self-loop bandit, a Meta Chain MDP, and Gridworld to assess estimation bias, action selection, and control performance. The bandit has one state and ten actions with zero-mean rewards, giving a zero optimal Q-value as the bias reference. The Meta Chain MDP Peer et al. (2021) offers a choice between terminating with zero reward and entering a state with multiple noisy-reward actions. Positive and negative reward means test how overestimation and underestimation affect this choice. We measure the cumulative optimal-action selection rate at the start state, averaged equally across six reward-mean settings. Gridworld evaluates control performance on 3×3 and 4×4 domains Hasselt (2010); Zhang et al. (2017); Zhu & Rigotti (2021).

Figure 3(a) contrasts the positive bias of Q-learning with the negative bias of Double Q-learning. SOQ approaches the zero-bias reference and achieves the smallest final absolute bias among the compared methods. In the Meta Chain MDP, SOQ achieves the highest cumulative optimal-action

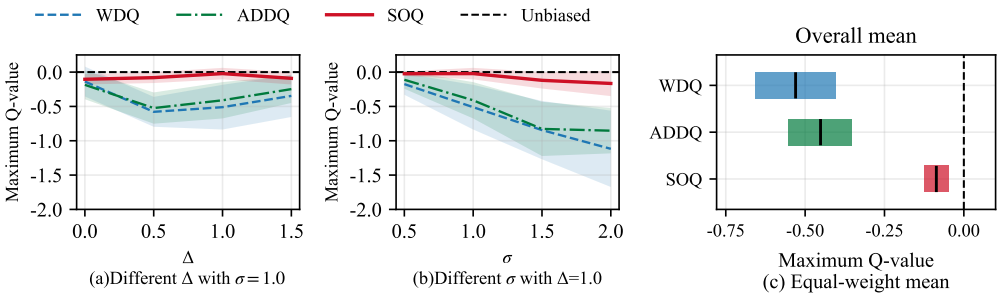


Figure 4: Comparison with adaptive methods across signal and noise settings. (a) and (b) report mean maximum Q-values over 100 independent runs per setting, with shaded 95% confidence intervals. (c) reports the equally weighted mean across seven distinct settings; black ticks mark means and colored rectangles show 95% confidence intervals.

selection rate by the end of training (Figure 3(b)). It also achieves the highest final average reward per step in both Gridworld environments (Figures 3(c) and 3(d)). Together, these results show that SOQ controls estimation bias while improving decision quality and tabular control performance.

Evaluation under different Environments. To examine robustness across signal and noise levels, we vary the reward distributions in the same ten-action self-loop bandit. Rewards are Gaussian, with means equally spaced over $[-\Delta, 0]$ and a shared standard deviation σ. We compare WDQ, ADDQ (Döring et al., 2025), and SOQ under these settings. In Figure 4(a), Δ varies with $\sigma = 1.0$, whereas in Figure 4(b), σ varies with $\Delta = 1.0$. Figure 4(c) summarizes the seven distinct settings by their equally weighted mean. Each point in Figures 4(a) and 4(b) is the mean final maximum Q-value over 100 independent runs.

As can be seen from the figure, although WDQ and ADDQ can converge to be nearly unbiased under some environmental settings, their bias grows significantly larger when facing more varied configurations. In contrast, the proposed SOQ can stably converge to be nearly unbiased under all kinds of settings, indicating that our algorithm has a stronger capability for bias minimization.

5 Deep Reinforcement Learning Experiments

We evaluate on Atari-10, the ten-game subset of the Arcade Learning Environment identified by Aitchison et al. (2023). We compare SOQ-DQN (Algorithm 2) with DQN (Mnih et al., 2015), EBQL (Peer et al., 2021), and MINTO (Hendawy et al., 2026). The experimental protocol specifies five random seeds per method and game and a training budget of 20 million action-decision steps (80 million frames). EBQL, MINTO, and SOQ use eight Q-heads each.

We measure the area under the evaluation learning curve (AUC), capturing performance throughout training. Seed AUCs are averaged within each game, normalized relative to DQN, and then averaged equally across games (DQN = 100). Appendix E details the evaluation protocol, normalization, and bootstrap procedure; Figure 7 provides per-game learning curves.

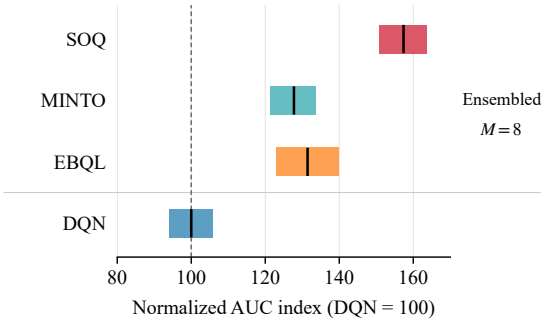


Figure 5: Normalized AUC on Atari-10. Black ticks indicate equal-game arithmetic means; colored blocks indicate 95% confidence intervals.

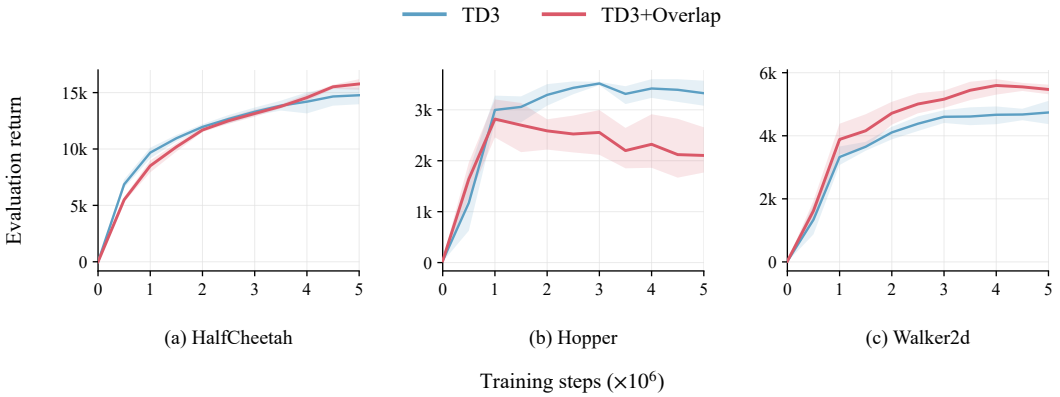


Figure 6: Continuous-control learning curves. Lines show mean evaluation returns over five training seeds after 0.5M-step interval averaging; shaded regions indicate 95% confidence intervals.

Figure 5 shows that SOQ achieves the highest mean normalized AUC, improving over DQN by 57.4%. Its index of 157.4 exceeds EBQL (131.4) and MINTO (127.8). Thus, SOQ delivers higher aggregate performance over the training horizon than DQN and the compared ensemble baselines, demonstrating the practical value of adaptive selection–evaluation overlap in deep reinforcement learning.

Continuous control. We extend selection–evaluation overlap to a TD3-based actor–critic implementation, denoted TD3+Overlap, and compare it with TD3 (Fujimoto et al., 2018) on HalfCheetah-v4, Hopper-v4, and Walker2d-v4. Each method is trained for five million steps with five random seeds per task. To handle continuous actions, TD3+Overlap learns a selector actor with a state-dependent subset of eight critics, whose size is controlled by a local SCR. Its target actor supplies bootstrap actions, evaluated by the mean of all target critics. A separate behavior actor learns from the full ensemble. This carries the subset-selection/full-ensemble-evaluation principle into the actor–critic setting while retaining TD3's target-policy smoothing and delayed policy updates. Details are provided in Appendix F. Figure 6 shows higher late-training returns on HalfCheetah and Walker2d, but lower returns on Hopper. These results demonstrate that the overlap approach can improve continuous-control performance, with benefits that vary across tasks.

6 Limitations

Our closed-form bias analysis assumes two actions with independent, equal-variance Gaussian errors; the multi-action SCR analysis additionally assumes equally spaced action values. Extending this characterization to correlated estimates and function approximation, and establishing optimality guarantees for the empirical SCR controller under learning dynamics, remain directions for future work.

7 Conclusion

We introduced selection–evaluation overlap as a control variable for Q-value estimation bias. Our analysis links the zero-bias overlap to SNR and motivates SOQ, which uses a state-dependent SCR to adapt the selection subset while retaining the full evaluation ensemble. Tabular experiments demonstrate reduced estimation bias, and Atari experiments show improved aggregate learning performance, supporting adaptive overlap as an effective approach to value-based reinforcement learning.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing, code development, figure preparation, and checking mathematical derivations. The authors take full responsibility for the accuracy and integrity of the final manuscript, code, and results.

REPRODUCIBILITY STATEMENT

The tabular and deep implementations are specified in Appendix C (Algorithms 1 and 2, respectively). Complete proofs of all theoretical results are provided in the appendix. Experimental protocols, environment definitions, and hyperparameter settings are reported in the appendix, and all experimental results were averaged over repeated runs with reported confidence intervals. The anonymized source code is available at https://anonymous.4open.science/r/Overlap-Q-learning-C168.

REFERENCES

Patigül Abliz. A controlling estimation bias method: Max_mix_min estimator for q-learning: P. abliz. *The Journal of Supercomputing*, 80(13):19248–19273, 2024.

Patigül Abliz and Shi Ying. Underestimation estimators to q-learning. *Information Sciences*, 607: 173–185, 2022.

Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the arcade learning environment down to five games. In *International Conference on Machine Learning*, pp. 421–438. PMLR, 2023.

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

Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pp. 243–252. PMLR, 2017.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W Ross. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021a.

Yifei Chen, Lambert Schomaker, and Marco A Wiering. An investigation into the effect of the learning rate on overestimation bias of connectionist q-learning. In *ICAART (2)*, pp. 107–118, 2021b.

Chen Chu, Yong Li, Jinzhuo Liu, Shuyue Hu, Xuelong Li, and Zhen Wang. A formal model for multiagent q-learning dynamics on regular graphs. In *IJCAI*, pp. 194–200, 2022.

Leif Döring, Benedikt Wille, Maximilian Birr, Mihail Bîrsan, and Martin Slowik. Addq: Adaptive distributional double q-learning. In *International Conference on Machine Learning*, pp. 14344–14390. PMLR, 2025.

Ghulam Farid, Muhammad Bilal, Lanyong Zhang, Ayman Alharbi, Ishaq Ahmed, and Muhammad Azhar. An improved deep q-learning approach for navigation of an autonomous uav agent in 3d obstacle-cluttered environment. *Drones*, 9(8):518, 2025.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2018.

Yaozhong Gan, Zhe Zhang, and Xiaoyang Tan. Stabilizing q learning via soft mellowmax operator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7501–7509, 2021.

Tianci Gao. Optimizing robotic arm control using deep q-learning and artificial neural networks through demonstration-based methodologies: A case study of dynamic and static conditions. *Robotics and Autonomous systems*, 181:104771, 2024.

Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23, 2010.

Ahmed Hendawy, Henrik Metternich, Théo Vincent, Mahdi Kallel, Jan Peters, and Carlo D'Eramo. Use the online network if you can: Towards fast and stable reinforcement learning. In *International Conference on Learning Representations*, 2026.

Haobo Jiang, Jin Xie, and Jian Yang. Action candidate based clipped double q-learning for discrete and continuous action tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7979–7986, 2021.

Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. In *International Conference on Learning Representations*, 2020.

Fan Li, Hong Qu, Liyan Zhang, Mingsheng Fu, Wenyu Chen, and Zhang Yi. Q-ader: An effective q-learning for recommendation with diminishing action space. *IEEE Transactions on Neural Networks and Learning Systems*, 36(5):8510–8524, 2024.

Shie Mannor, Duncan Simester, Peng Sun, and John N Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

Francisco S Melo. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Oren Peer, Chen Tessler, Nadav Merlis, and Ron Meir. Ensemble bootstrapping for q-learning. In *International conference on machine learning*, pp. 8454–8463. PMLR, 2021.

Zhizhou Ren, Guangxiang Zhu, Hao Hu, Beining Han, Jianglun Chen, and Chongjie Zhang. On the estimation bias in double q-learning. *Advances in Neural Information Processing Systems*, 34: 10246–10259, 2021.

Zhao Song, Ron Parr, and Lawrence Carin. Revisiting the softmax bellman operator: New benefits and new perspective. In *International conference on machine learning*, pp. 5916–5925. PMLR, 2019.

Tao Tan, Hong Xie, and Defu Lian. Adaptive order q-learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4946–4954, 2024.

Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 connectionist models summer school*, pp. 255–263. Psychology Press, 2014.

John N. Tsitsiklis. Asynchronous stochastic approximation and Q-Learning. *Machine Learning*, 16 (3):185–202, 1994.

Hang Wang, Sen Lin, and Junshan Zhang. Adaptive ensemble q-learning: Minimizing estimation bias via error feedback. *Advances in neural information processing systems*, 34:24778–24790, 2021.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Haolin Wu, Jianwei Zhang, Zhuang Wang, Yi Lin, and Hui Li. Sub-avg: Overestimation reduction for cooperative multi-agent reinforcement learning. *Neurocomputing*, 474:94–106, 2022.

Zhe Zhang, Yukun Zou, Junjie Lai, and Qing Xu. M2dqn: A robust method for accelerating deep q-learning network. In *Proceedings of the 2023 15th International Conference on Machine Learning and Computing*, pp. 116–120, 2023.

Zongzhang Zhang, Zhiyuan Pan, and Mykel J Kochenderfer. Weighted double q-learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3455–3461, 2017.

Fuqing Zhao, Qiaoyun Wang, and Ling Wang. An inverse reinforcement learning framework with the q-learning mechanism for the metaheuristic algorithm. *Knowledge-Based Systems*, 265: 110368, 2023.

Rong Zhu and Mattia Rigotti. Self-correcting q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11185–11192, 2021.

A Proofs for the Selective-Overlap Analysis

A.1 Proof of Lemma 1

Let $\hat{a}^* = \arg\max_a \hat{\mu}_a(S_a^N)$ be the action selected by the N-sample means. We split the M evaluation samples into the N overlapping samples that were also used for selection and the remaining $M - N$ samples, which are independent of the selection event. Conditioning on the selection, the expected evaluation decomposes as

$$\mathbb{E}\big[\hat{\mu}_{\hat{a}^*}(S_{\hat{a}^*})\big] = \frac{N}{M}\,\mathbb{E}\Big[\max_a \hat{\mu}_a(S_a^N)\Big] + \frac{M-N}{M}\,\mathbb{E}\big[\mu_{\hat{a}^*}\big], \tag{12}$$

where the first term uses the conditional (positively selected) mean of the overlapping samples and the second term uses the unconditional mean of the independent samples. Subtracting μ_1 and defining

$$G_N \triangleq \mathbb{E}\Big[\max_a \hat{\mu}_a(S_a^N)\Big] - \mu_1, \qquad R_N \triangleq \mu_1 - \mathbb{E}\big[\mu_{\hat{a}^*}\big],$$

we obtain the standard decomposition

$$\mathrm{Bias}^{\mathrm{OE}} = \frac{N}{M}\,G_N - \left(1 - \frac{N}{M}\right) R_N.$$

For two actions, let $D = \hat{\mu}_1(S_1^N) - \hat{\mu}_2(S_2^N) \sim \mathcal{N}(\Delta,\, 2\sigma^2/N)$ and $z = \Delta/\sqrt{2\sigma^2/N} = \sqrt{N}\Delta/(\sqrt{2}\sigma)$. A suboptimal action is selected with probability $\Phi(-z)$, hence

$$R_N = \Delta\,\Phi(-z) = \Delta\,(1 - \Phi(z)).$$

Using the expectation of the maximum of two independent Gaussians,

$$\mathbb{E}\Big[\max_a \hat{\mu}_a(S_a^N)\Big] = \mu_1 + \sqrt{\tfrac{2\sigma^2}{N}}\,\phi(z) - \Delta\,(1 - \Phi(z)),$$

so $G_N = \sqrt{\frac{2\sigma^2}{N}}\,\phi(z) - \Delta\,(1 - \Phi(z))$. Substituting G_N and R_N into the decomposition,

$$\begin{aligned}
\mathrm{Bias}^{\mathrm{OE}} &= \frac{N}{M}\Big[\sqrt{\tfrac{2\sigma^2}{N}}\,\phi(z) - \Delta(1 - \Phi(z))\Big] - \left(1 - \frac{N}{M}\right)\Delta(1 - \Phi(z)) \\
&= \frac{\sqrt{2N}}{M}\,\sigma\,\phi(z) - \Delta\,(1 - \Phi(z)),
\end{aligned}$$

which proves the lemma.

Monotonicity in N and M. Differentiating the bias with respect to the overlap size N gives

$$\frac{\partial \mathrm{Bias}^{\mathrm{OE}}}{\partial N} = \frac{\phi(z)}{2\sqrt{N}}\left[\frac{\sqrt{2}\,\sigma}{M} + \frac{\Delta^2}{\sqrt{2}\,\sigma}\left(1 - \frac{N}{M}\right)\right] > 0,$$

which is strictly positive for $0 < N \le M$ and $\sigma > 0$, so the signed bias is monotonically increasing in the overlap size N. For fixed N,

$$\frac{\partial \mathrm{Bias}^{\mathrm{OE}}}{\partial M} = -\frac{\sqrt{2N}\,\sigma\,\phi(z)}{M^2} < 0,$$

so increasing the number of evaluation estimators M reduces the signed bias.

Bias gap to an independent-evaluation estimator. Let CE_N denote the comparison estimator that uses independent samples for action evaluation while keeping the same selection procedure. Under the two-action Gaussian model,

$$\mathrm{Bias}^{\mathrm{OE}}(N, M) - \mathrm{Bias}^{\mathrm{CE}}(N) = \frac{\sqrt{2N}\,\sigma}{M}\,\phi(z_N), \qquad z_N = \frac{\sqrt{N}\Delta}{\sqrt{2}\,\sigma}.$$

Using $\phi(z) \le 1/\sqrt{2\pi}$, we obtain

$$0 \le \mathrm{Bias}^{\mathrm{OE}} - \mathrm{Bias}^{\mathrm{CE}} \le \frac{\sigma\sqrt{N}}{M\sqrt{\pi}} \longrightarrow 0 \qquad (M/N \to \infty).$$

Thus the bias gap vanishes as M/N grows, so OE approaches the bias of an independent-evaluation estimator in the low-overlap regime.

A.2 Proof of Proposition 1

By Lemma 1, $\mathrm{Bias}^{\mathrm{OE}}(N) = \frac{\sqrt{2N}}{M}\,\sigma\,\phi(z) - \Delta\,[1-\Phi(z)]$, where $z = \frac{\sqrt{N}\Delta}{\sqrt{2}\,\sigma}$.

Negative bias near $N \to 0^+$. As $N \to 0^+$, we have $z \to 0$, $\phi(z) \to \phi(0) = 1/\sqrt{2\pi}$, and $\Phi(z) \to 1/2$. The positive term $\frac{\sqrt{2N}}{M}\sigma\phi(z) \to 0$, hence

$$\lim_{N\to 0^+} \mathrm{Bias}^{\mathrm{OE}}(N) = -\Delta\left(1-\frac{1}{2}\right) = -\frac{\Delta}{2} < 0.$$

Positive bias at $N = M$. Let $z_M = \frac{\sqrt{M}\Delta}{\sqrt{2}\,\sigma}$. Since $\Delta = \frac{\sqrt{2}\,\sigma}{\sqrt{M}}\,z_M$,

$$\begin{aligned}\mathrm{Bias}^{\mathrm{OE}}(M) &= \frac{\sqrt{2}}{\sqrt{M}}\,\sigma\,\phi(z_M) - \Delta\,(1-\Phi(z_M)) \\ &= \frac{\sqrt{2}\,\sigma}{\sqrt{M}}\,[\phi(z_M) - z_M\,(1-\Phi(z_M))]\,.\end{aligned}$$

By the Mills-ratio inequality, for $z_M > 0$,

$$1-\Phi(z_M) < \frac{\phi(z_M)}{z_M} \quad\Longleftrightarrow\quad \phi(z_M) - z_M\,(1-\Phi(z_M)) > 0.$$

Therefore $\mathrm{Bias}^{\mathrm{OE}}(M) > 0$.

Existence of N^*. The function $\mathrm{Bias}^{\mathrm{OE}}(N)$ is continuous in $N \in (0, M]$, is negative as $N \to 0^+$, and is positive at $N = M$. The intermediate value theorem therefore gives a root $N^* \in (0, M)$. Together with the monotonicity established in the proof of Lemma 1, this root is unique, and the signed bias changes from underestimation to overestimation as N increases.

A.3 Proof of Proposition 2

Setting $\mathrm{Bias}^{\mathrm{OE}}(N) = 0$ in Lemma 1 and using $z = \sqrt{N}\Delta/(\sqrt{2}\sigma)$ gives

$$\frac{z\phi(z)}{1-\Phi(z)} = \frac{M\Delta^2}{2\sigma^2} = \frac{M\varepsilon^2}{2} \triangleq \lambda. \tag{13}$$

For fixed M, the low-SNR limit $\varepsilon \to 0$ implies $\lambda \to 0$ and $z \to 0$. Around $z = 0$,

$$\frac{z\phi(z)}{1-\Phi(z)} = \sqrt{\frac{2}{\pi}}\,z + \frac{2}{\pi}z^2 + O(z^3).$$

Inverting this expansion yields

$$z^* = \sqrt{\frac{\pi}{2}}\,\lambda + O(\lambda^2).$$

Substituting $N = 2\sigma^2 z^2/\Delta^2 = 2z^2/\varepsilon^2$ and $\lambda = M\varepsilon^2/2$ gives

$$N^* = \frac{\pi M^2}{4}\varepsilon^2\big(1+O(\varepsilon^2)\big),$$

which proves Proposition 2.

A.4 Proof of Lemma 2

The total action-value gap is Δ, so the adjacent-action gap is $\Delta/(K-1)$. For the jth suboptimal action, the equal-spacing assumption gives

$$Q(s', a_{(j)}) - Q(s', a^*) \sim \mathcal{N}\left(-\frac{j\Delta}{K-1}, 2\sigma^2\right).$$

Therefore,

$$\Pr\big(Q(s', a_{(j)}) > Q(s', a^*)\big) = \Phi\left(-\frac{j\Delta}{(K-1)\sqrt{2}\sigma}\right) = \Phi\left(-\frac{j\varepsilon}{(K-1)\sqrt{2}}\right),$$

where $\varepsilon = \Delta/\sigma$. Averaging these pairwise probabilities over the $K-1$ suboptimal actions yields Equation (3). For fixed K, its derivative is

$$\frac{\mathrm{d}c}{\mathrm{d}\varepsilon} = -\frac{1}{(K-1)^2\sqrt{2}} \sum_{j=1}^{K-1} j\,\phi\left(\frac{j\varepsilon}{(K-1)\sqrt{2}}\right) < 0,$$

so c is strictly decreasing in ε. When $K = 2$, Equation (3) reduces to $c = \Phi(-\varepsilon/\sqrt{2})$.

A.5 Proof of Lemma 3

By Equation (3), each summand converges to $\Phi(0) = 1/2$ as $\varepsilon \to 0$. Hence,

$$\lim_{\varepsilon\to 0} c = \frac{1}{K-1} \sum_{j=1}^{K-1} \frac{1}{2} = \frac{1}{2}.$$

A.6 Proof of Lemma 4

By Equation (3), each summand converges to zero as $\varepsilon \to \infty$. Since the sum contains finitely many terms,

$$\lim_{\varepsilon\to\infty} c = \frac{1}{K-1} \sum_{j=1}^{K-1} 0 = 0.$$

B Bellman-Target Decomposition and Tabular Convergence

At update t, let N_t denote the current value of N in Algorithm 1, let $[M] = \{1, \ldots, M\}$, and let $\mathcal{I}_t \subseteq [M]$ be the selector set sampled in Equation (4), with $|\mathcal{I}_t| = N_t$. Define

$$\bar{Q}_t(s,a) = \frac{1}{M}\sum_{i=1}^{M} Q_t^i(s,a), \qquad \bar{Q}_{\mathcal{I}_t,t}(s,a) = \frac{1}{N_t}\sum_{i\in\mathcal{I}_t} Q_t^i(s,a).$$

The selector chooses $\widehat{A}_t(s) \in \arg\max_a \bar{Q}_{\mathcal{I}_t,t}(s,a)$, whereas the SOQ target evaluates that action with $\bar{Q}_t$. Define the full-ensemble greedy target and the corresponding selection loss as

$$Y_t^{\mathrm{Full}} = R_t + \gamma(1-\tau_t)\max_{a\in\mathcal{A}} \bar{Q}_t(S_{t+1},a), \tag{14}$$

$$L_t(s) = \max_{a\in\mathcal{A}} \bar{Q}_t(s,a) - \bar{Q}_t(s,\widehat{A}_t(s)). \tag{15}$$

Proposition 4 (Target decomposition)**.** *For every transition and every legal N_t,*

$$Y_t^{\mathrm{SOQ}} = Y_t^{\mathrm{Full}} - \gamma(1-\tau_t)L_t(S_{t+1}), \tag{16}$$

and

$$0 \le L_t(s) \le 2\|\bar{Q}_{\mathcal{I}_t,t}(s,\cdot) - \bar{Q}_t(s,\cdot)\|_\infty \le 2\max_{i\in[M]} \|Q_t^i - \bar{Q}_t\|_\infty. \tag{17}$$

Proof. Equation (16) follows by subtracting the value of the selector action from the maximum full-ensemble value. Let $a_M \in \arg\max_a \bar{Q}_t(s,a)$ and $a_{\mathcal{I}} = \widehat{A}_t(s)$. Since $a_{\mathcal{I}}$ maximizes $\bar{Q}_{\mathcal{I}_t,t}$,

$$\begin{aligned} L_t(s) &= \bar{Q}_t(s,a_M) - \bar{Q}_t(s,a_{\mathcal{I}}) \\ &\le |\bar{Q}_t(s,a_M) - \bar{Q}_{\mathcal{I}_t,t}(s,a_M)| + |\bar{Q}_{\mathcal{I}_t,t}(s,a_{\mathcal{I}}) - \bar{Q}_t(s,a_{\mathcal{I}})| \\ &\le 2\|\bar{Q}_{\mathcal{I}_t,t}(s,\cdot) - \bar{Q}_t(s,\cdot)\|_\infty. \end{aligned}$$

The final inequality in Equation (17) follows because the selector mean is an average of deviations $Q_t^i - \bar{Q}_t$. This argument is distribution-free and does not assert samplewise monotonicity in N_t.

Proof of Proposition 3. Contractivity and $\mathcal{T}^\star Q^\star = Q^\star$ give

$$\|q_{k+1} - Q^\star\|_\infty \le \gamma \|q_k - Q^\star\|_\infty + \|\delta_k\|_\infty.$$

Iterating this inequality yields

$$\|q_K - Q^\star\|_\infty \le \gamma^K \|q_0 - Q^\star\|_\infty + \sum_{k=0}^{K-1} \gamma^{K-1-k} \|\delta_k\|_\infty. \tag{18}$$

The uniform bound in Equation (10) follows from the geometric series. If $\|\delta_k\|_\infty \to 0$, splitting the sum into a finite prefix and a vanishing tail gives $q_k \to Q^\star$.

Lemma 5 (Asynchronous contraction updates with vanishing perturbations)**.** *Let $F : \mathbb{R}^d \to \mathbb{R}^d$ be monotone and a γ-contraction in $\|\cdot\|_\infty$, with fixed point $x^\star$. Consider a bounded asynchronous recursion that updates coordinate I_t as*

$$\begin{aligned} x_{t+1}(I_t) = x_t(I_t) + \eta_t(I_t)\big[&(Fx_t)(I_t) - x_t(I_t) \\ &+ \xi_{t+1}(I_t) + c_{t+1}(I_t)\big], \end{aligned} \tag{19}$$

with all other coordinates unchanged. Suppose every coordinate is updated infinitely often; its local gains lie in $[0, 1]$, have divergent sum, and have square-summable squares. Suppose also that ξ_{t+1} is a martingale-difference noise with uniformly bounded conditional second moment, and $\|c_{t+1}\|_\infty \to 0$ almost surely. Then $x_t \to x^\star$ almost surely.

Proof. Fix $\varepsilon > 0$. For every deterministic time K, define lower and upper comparison recursions initialized at x_K, driven by the same future coordinates, gains, and martingale increments, with mean operators $F(x) - \varepsilon\mathbf{1}$ and $F(x) + \varepsilon\mathbf{1}$. Take the countable intersection of the probability-one events on which all such recursions converge and $c_t \to 0$. On this event, some K satisfies $\sup_{t \ge K} \|c_t\|_\infty \le \varepsilon$, so the comparison processes defined for that K bound the perturbed recursion. Standard asynchronous contraction convergence applies to both processes (Tsitsiklis, 1994), and their fixed points are within $\varepsilon/(1-\gamma)$ of $x^\star$. Repeating the argument for a countable sequence $\varepsilon \downarrow 0$ proves the claim.

Proof of Theorem 1. We first specify the sampling order used in the argument. Let $\mathcal{H}_t$ contain the history through the current tables, state S_t, behavior action A_t, and local gain, but not the transition at time t. After observing (R_t, S_{t+1}, τ_t), the controller chooses N_t and the selector set, and constructs the common target Y_t^{SOQ}. Let $\mathcal{G}_t$ augment $\mathcal{H}_t$ with these variables, but not the update index J_t. Algorithm 1 samples

$$\Pr(J_t = j \mid \mathcal{G}_t) = \frac{1}{M}, \qquad j \in [M]. \tag{20}$$

Thus, the controller and selector may depend on the observed transition and the current ensemble, while J_t remains conditionally independent of the constructed target.

Boundedness. Choose

$$V_{\max} \ge \max\left\{ \max_{i \in [M]} \|Q_0^i\|_\infty, \frac{R_{\max}}{1-\gamma} \right\}.$$

If all tables lie in $[-V_{\max}, V_{\max}]$, then their full mean and the SOQ evaluation lie in the same interval. Hence $|Y_t^{\mathrm{SOQ}}| \le R_{\max} + \gamma V_{\max} \le V_{\max}$. Because $\alpha_t(x_t) \in [0, 1]$, every update is a convex combination of two values in this interval. Induction therefore establishes boundedness of all iterates and targets.

Consensus. Fix a state–action coordinate x and follow its local visits. Stack its table values as $q_t = (Q_t^1(x), \ldots, Q_t^M(x))^\top$, and define

$$m_t = \frac{1}{M}\mathbf{1}^\top q_t, \qquad d_t = \left(I - \frac{1}{M}\mathbf{1}\mathbf{1}^\top\right) q_t.$$

At a visit to x, conditional on $\mathcal{G}_t$, the common target y_t is fixed and

$$q_{t+1} = q_t + \alpha_t(x) e_{J_t}(y_t - Q_t^{J_t}(x)).$$

Using $\sum_j d_t(j) = 0$ and $\sum_j d_t(j) Q_t^j(x) = \|d_t\|_2^2$ gives

$$\mathbb{E}\big[\|d_{t+1}\|_2^2 \mid \mathcal{G}_t\big] \le \left(1 - \frac{2\alpha_t(x)}{M}\right) \|d_t\|_2^2 + C\alpha_t^2(x), \tag{21}$$

where $C < \infty$ follows from boundedness. Robbins–Siegmund applied on the local visit clock shows that $\|d_t\|_2^2$ converges and that $\sum_t \alpha_t(x)\|d_t\|_2^2 < \infty$. Since $\sum_t \alpha_t(x) = \infty$, the limit is zero. The state and action spaces are finite and every coordinate is visited infinitely often, so

$$\mathcal{D}_t \triangleq \max_{i \in [M]} \|Q_t^i - \bar{Q}_t\|_\infty \longrightarrow 0 \qquad \text{almost surely.} \tag{22}$$

Mean process. Let $X_t = (S_t, A_t)$, and let $\nu_t(x)$ be the number of visits to coordinate x through time t. At the visited coordinate, write $\alpha_t(X_t) = \alpha_{\nu_t(X_t)}(X_t)$. The ensemble mean obeys the exact recursion

$$\bar{Q}_{t+1}(X_t) = \bar{Q}_t(X_t) + \frac{\alpha_t(X_t)}{M}\left[Y_t^{\mathrm{SOQ}} - Q_t^{J_t}(X_t)\right]. \tag{23}$$

Let

$$Y_t^{\mathrm{Full}} = R_t + \gamma(1 - \tau_t) \max_{a \in \mathcal{A}} \bar{Q}_t(S_{t+1}, a).$$

Using Proposition 4, the bracket in Equation (23) decomposes as

$$\begin{aligned} Y_t^{\mathrm{SOQ}} - Q_t^{J_t}(X_t) = (\mathcal{T}^\star \bar{Q}_t)(X_t) - \bar{Q}_t(X_t) \\ + \xi_{t+1}(X_t) + c_{t+1}(X_t), \end{aligned} \tag{24}$$

where

$$\xi_{t+1}(X_t) = Y_t^{\mathrm{Full}} - (\mathcal{T}^\star \bar{Q}_t)(X_t) + \bar{Q}_t(X_t) - Q_t^{J_t}(X_t), \tag{25}$$

$$c_{t+1}(X_t) = -\gamma(1 - \tau_t) L_t(S_{t+1}). \tag{26}$$

Set both quantities to zero away from X_t. The transition model makes the first difference in Equation (25) conditionally mean zero given $\mathcal{H}_t$. Equation (20) makes the second difference conditionally mean zero given $\mathcal{G}_t$. The tower property therefore makes ξ_{t+1} a martingale-difference noise relative to the pre-transition history, and boundedness supplies a uniform conditional second-moment bound.

The effective local gain $\alpha_t(X_t)/M$ retains the Robbins–Monro properties. Moreover, Equations (17) and (22) give

$$\|c_{t+1}\|_\infty \le 2\gamma \mathcal{D}_t \longrightarrow 0 \qquad \text{almost surely.}$$

Applying Lemma 5 with $F = \mathcal{T}^\star$ yields $\bar{Q}_t \to Q^\star$ almost surely. Consensus then gives $Q_t^i \to Q^\star$ for every $i \in [M]$, which proves Equation (11). The argument does not require N_t to converge.

C Algorithmic Specifications

C.1 Tabular Selective-Overlap Q-learning

Algorithm 1 combines the empirical SCR and adaptive overlap rule in Equations (8)–(9) with the tabular update in Equation (6).

Algorithm 1: Tabular Selective-Overlap Q-learning

Parameter: number of Q-tables M, $\rho = 0.5$
Init: Q-tables $Q = \{Q^1, Q^2, \cdots, Q^M\}$; start state s_0

for $t = 0, 1, 2, \ldots$ **do**
- Choose action a at state s by ϵ-greedy policy
- $r, s', \tau \leftarrow$ Environment(s, a)
- ▷ Action selection
  - $N \leftarrow$ Equation 8, 9
  - Sample $\mathcal{I}_t \subseteq [M]$, $|\mathcal{I}_t| = N$, uniformly without replacement
  - Select target action $\hat{a}_N$ using Equation 5
- ▷ Action evaluation
  - Independently and uniformly sample the update index $i \in [M]$
  - Update $Q_t^i(s, a)$ as Equation 6
- $s \leftarrow s'$

C.2 Fixed-Overlap OQ

Fixed-overlap OQ isolates the effect of the selector size from the adaptive controller. For integers $1 \le N_0 \le M$, OQ(M, N_0) sets $N_t = N_0$ throughout learning and samples a fresh selector set $\mathcal{I}_t \subseteq [M]$, with $|\mathcal{I}_t| = N_0$, uniformly without replacement for each target construction. Its target action and tabular target are

$$\widehat{a}_{N_0} \in \arg\max_{a'} \frac{1}{N_0} \sum_{i \in \mathcal{I}_t} Q_t^i(s', a'), \tag{27}$$

$$Y_t^{\mathrm{OQ}(M,N_0)} = r_t + \gamma(1-\tau_t)\frac{1}{M}\sum_{i=1}^{M} Q_t^i(s', \widehat{a}_{N_0}). \tag{28}$$

One uniformly sampled Q-table is updated using this common target, as in Equation (6). Thus, OQ(M, N_0) differs from tabular SOQ only by omitting the SCR computation and holding the selector size fixed. The endpoints OQ(M, M) and OQ$(M, 1)$ retain the same all-M evaluator but use fully overlapping and minimum-size selectors, respectively.

C.3 Overlap DQN and Deep SOQ

The deep implementation maintains a shared online feature extractor with M online Q-heads $\{Q_{\theta_i}\}_{i=1}^M$ and a periodically synchronized target copy $\{Q^-_{\bar{\theta}_i}\}_{i=1}^M$. Every learner step updates all M online heads and the shared extractor. Each stored transition carries persistent, independently sampled Bernoulli bootstrap memberships, which define the replay view for each head. Head j independently samples a minibatch $\mathcal{B}_j$ with replacement from its eligible transitions. A fresh selector subset is drawn for each sampled transition, without requiring it to contain head j.

For a replay transition $b = (s_b, a_b, r_b, s'_b, \tau_b)$ and a selector set $\mathcal{I}_b$, the deep target is

$$\widehat{a}_b \in \arg\max_{a'} \frac{1}{N_b} \sum_{i \in \mathcal{I}_b} Q^-_{\bar{\theta}_i}(s'_b, a'), \tag{29}$$

$$y_b = r_b + \gamma(1-\tau_b)\frac{1}{M}\sum_{i=1}^{M} Q^-_{\bar{\theta}_i}(s'_b, \widehat{a}_b). \tag{30}$$

Overlap DQN, denoted OQ-DQN(M, N_0), fixes $N_b = N_0$. SOQ-DQN instead computes N_b from Equations (8) and (9) using the target-head ensemble at s'_b. In both cases, the subset is sampled independently for each replay sample, and all M target heads evaluate the selected action with equal weight. Behavior and evaluation actions are greedy with respect to the all-head online mean, with ϵ-greedy exploration used during training.

Joint update and training settings. Let ψ denote the shared extractor parameters and θ_j the parameters specific to head j. With targets held fixed, each head uses the Huber loss ℓ_1 with unit threshold:

$$\mathcal{L}_j = \frac{1}{B}\sum_{b \in \mathcal{B}_j} \ell_1\big(Q_{\theta_j}(s_b, a_b) - y_b\big), \qquad \mathcal{L} = \frac{1}{M}\sum_{j=1}^{M} \mathcal{L}_j. \tag{31}$$

The shared extractor receives $\nabla_\psi \mathcal{L}$. Each head receives $M\nabla_{\theta_j}\mathcal{L} = \nabla_{\theta_j}\mathcal{L}_j$, preserving its per-head gradient scale. The extractor gradient and each head gradient are clipped separately to norm 10 before a single joint optimizer step. The Atari implementation uses $M = 8$, $B = 32$ per head, and bootstrap probability $p = 0.5$. Thus, each learner step draws 8×32 replay samples across heads. Learning starts after 100,000 environment steps and runs every four steps; the target extractor and all target heads are synchronized every 10,000 steps.

Algorithm 2: Overlap DQN and Deep SOQ

Parameter: heads M; per-head batch size B; mode $g \in \{\text{Fixed}, \text{Adaptive}\}$; fixed size N_0; bootstrap probability p; target-update period C

Init: shared online extractor ψ and heads $\{\theta_i\}_{i=1}^{M}$; matching target copies; replay buffer $\mathcal{D}$

for $t = 0, 1, 2, \ldots$ **do**
- Select a_t by an ϵ-greedy policy based on $M^{-1} \sum_i Q_{\theta_i}(s_t, \cdot)$
- Observe (r_t, s_{t+1}, τ_t) and sample $m_t^i \sim \text{Bernoulli}(p)$ independently for $i \in [M]$
- Store $(s_t, a_t, r_t, s_{t+1}, \tau_t, m_t^1, \ldots, m_t^M)$ in $\mathcal{D}$
- **if** *a learner update is scheduled* **then**
  - **for** $j = 1, \ldots, M$ **do**
    - Independently sample B transitions with replacement from $\{b \in \mathcal{D} : m_b^j = 1\}$ to form $\mathcal{B}_j$
    - **foreach** *sampled transition* $b \in \mathcal{B}_j$ **do**
      - **if** $g = \text{Fixed}$ **then**
        - $N_b \leftarrow N_0$
      - **else**
        - Compute $\widehat{c}_b$ and N_b using the target ensemble and Equations (8)–(9)
      - Sample $\mathcal{I}_b \subseteq [M]$, $|\mathcal{I}_b| = N_b$, uniformly without replacement
      - Select $\widehat{a}_b$ and construct the fixed target y_b using Equations (29)–(30)
    - Compute the head loss $\mathcal{L}_j$ using Equation (31)
  - Backpropagate $\mathcal{L} = M^{-1} \sum_j \mathcal{L}_j$; multiply each head's gradient by M
  - Clip the extractor and each head gradient separately; take one joint optimizer step
- **if** t *is a multiple of* C **then**
  - Synchronize the target extractor and all target heads

Algorithm 2 changes only the action-selection subset between the fixed and adaptive modes. Replay membership, per-head minibatch sampling, joint all-head updates, all-head target evaluation, behavior policy, and evaluation policy remain identical across overlap conditions.

D Details of Tabular Experiments

Experiments were conducted on servers equipped with four NVIDIA A100 GPUs and 512 GB of RAM. Each deep reinforcement learning run took 36 hours on average.

D.1 Environments

We evaluate SOQ on three tabular environments: a multi-armed bandit, a Meta Chain MDP, and Gridworld. These environments are designed to examine different aspects of estimation bias, including direct bias measurement, policy accuracy, and tabular control performance.

Multi-Armed Bandit. Following prior work Mannor et al. (2007), we adopt a multi-armed bandit environment to analyze how different parameters affect the estimation bias of SOQ. The environment has one state and ten actions. Every action returns to the same state, with no terminal transition, and rewards are independently sampled from $\mathcal{N}(0, 1)$. With $\gamma = 0.95$, all true action values are zero. Figure 3(a) reports the mean maximum estimated Q-value over 100 independent runs of 10,000 environment steps.

For Figure 4, we retain this transition structure and set

$$r \mid a_i \sim \mathcal{N}\left(-\frac{(i-1)\Delta}{9}, \sigma^2\right), \qquad i = 1, \ldots, 10.$$

Here Δ is the total reward-mean gap and σ is the reward standard deviation. The optimal Q-value remains zero, including when $\Delta = 0$ and all actions have equal means. We use $\Delta \in \{0, 0.5, 1, 1.5\}$ with $\sigma = 1$, and $\sigma \in \{0.5, 1, 1.5, 2\}$ with $\Delta = 1$. The shared setting is counted once, giving seven distinct environments. Each method uses 100 independent runs of 30,000 steps per setting, with fixed learning rate $\alpha = 0.1$ and $\gamma = 0.95$.

Meta Chain MDP. To evaluate policy accuracy, we follow prior work Peer et al. (2021) and use a Meta Chain MDP. Each episode starts at state A. Choosing left gives zero reward and leads to state B; choosing right gives zero reward and terminates at state C. State B has 50 actions, each producing an independent reward $r \sim \mathcal{N}(\mu, 2^2)$ and terminating at state D. With $\gamma = 1$, the true values at A are $Q^*(A, \text{left}) = \mu$ and $Q^*(A, \text{right}) = 0$. Left is therefore optimal for $\mu > 0$, and right for $\mu < 0$. We use $\mu \in \{-0.6, -0.4, -0.2, 0.2, 0.4, 0.6\}$, with 100 independent runs of 5,000 episodes per setting.

The reported policy accuracy is the cumulative optimal-action selection rate at A. For a run in setting μ, after t episodes it is

$$\text{Acc}_\mu(t) = \frac{1}{t} \sum_{e=1}^{t} \mathbf{1}\{a_e(A) = a^*_\mu(A)\},$$

where $a_e(A)$ is the action actually taken in episode e, including ϵ-greedy exploration with $\epsilon = 0.1$. At each episode, we first average this rate over the 100 runs within each setting, then average equally over the six settings.

Gridworld. We further evaluate SOQ in tabular control domains using Gridworld environments Hasselt (2010); Zhang et al. (2017); Zhu & Rigotti (2021). The Gridworld consists of a $w \times w$ state space corresponding to physical locations, with four available actions: up, down, left, and right. We consider two settings:$w \in \{3, 4\}$, corresponding to 3×3 and 4×4 Gridworlds. The goal state is located at the northeast corner, and the start state is located at the southwest corner. Executing any action in the goal state returns the agent to the start state and gives a random reward sampled uniformly from $\{r_1 = 40, r_2 = -30\}$. In any non-goal state, executing any action gives a random reward sampled uniformly from $\{r_3 = 10, r_4 = -12\}$.

D.2 Hyperparameter Settings

Following prior work, we use count-based learning rates and exploration schedules in the tabular experiments. For the bandit experiments in Figure 3 and the overlap ablations, we set

$$\gamma = 0.95, \qquad \alpha = \frac{1}{n(s,a)^{0.8}}, \qquad \epsilon = \frac{1}{\sqrt{n(s)}}.$$

Here $n(s)$ counts state visits. The same exploration schedule is used in Figure 4, with the fixed learning rate specified above.

For the Meta Chain MDP, we use

$$\gamma = 1.0, \qquad \alpha = \frac{1}{n(s,a)}, \qquad \epsilon = 0.1.$$

For the Gridworld environments, we use

$$\gamma = 0.95, \qquad \alpha = \frac{1}{n(s,a)}, \qquad \epsilon = \frac{1}{\sqrt{n(s,a)}}.$$

Here, γ denotes the discount factor, α denotes the learning rate, ϵ denotes the probability of random exploration, and $n(s,a)$ denotes the number of visits to action a in state s.

D.3 Confidence Intervals for Figure 4

Results in Figure 4 are taken from the last scheduled logging checkpoint of each 30,000-step run under the fixed-learning-rate protocol.

For Figures 4(a) and 4(b), each environment setting has $n = 100$ independent run-level maximum Q-values. Let $\bar{x}$ and s denote their sample mean and sample standard deviation. The shaded intervals are pointwise 95% Student-t confidence intervals for the mean,

$$\bar{x} \pm t_{0.975,n-1}\frac{s}{\sqrt{n}},$$

where $t_{0.975,n-1}$ is the 97.5th percentile of the Student-t distribution with $n - 1 = 99$ degrees of freedom. Each interval is computed separately at its environment setting.

Figure 4(c) averages the seven settings with equal weights, counting the setting shared by (a) and (b) once. Its 95% confidence intervals use 20,000 stratified bootstrap resamples: within each setting, we resample the 100 independent runs with replacement and recompute the equally weighted mean. The interval endpoints are the 2.5th and 97.5th percentiles.

E Atari Evaluation Details

Figure 5 uses the Atari-10 subset defined by Aitchison et al. (2023): Amidar, BattleZone, Bowling, DoubleDunk, Frostbite, KungFuMaster, NameThisGame, Phoenix, Qbert, and Riverraid. The experiment uses five random seeds per method and game.

E.1 Training and Evaluation Protocol

We use ALE-v5 with the minimal action set and sticky-action probability 0.25. Each reset applies 1–30 random no-op actions and a FIRE action when required. Actions repeat for four frames, with pixelwise maximum pooling over the final two frames. Observations are converted to 84×84 grayscale images, stacked over four frames, and scaled to $[0, 1]$. Training treats life loss as an episode boundary and clips rewards to their signs; evaluation uses full-game episodes and reports unclipped returns.

The encoder has three ReLU convolutional layers with 32, 64, and 64 channels, kernels of sizes 8×8, 4×4, and 3×3, and strides 4, 2, and 1, followed by a 512-unit ReLU layer. SOQ shares this

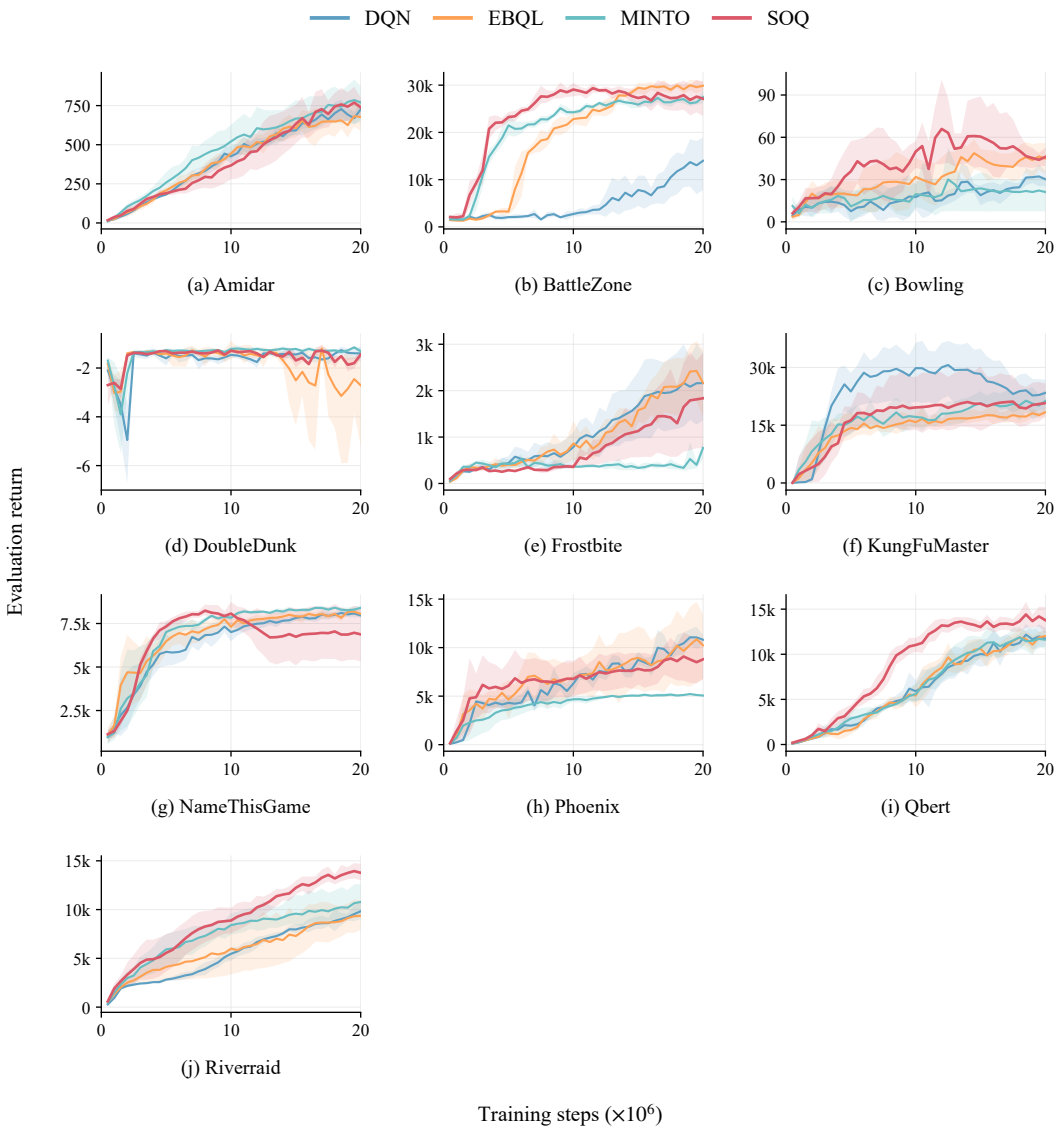


Figure 7: Evaluation learning curves on Atari-10. Lines show mean evaluation returns across seeds after 0.5M-step interval averaging; shaded regions indicate pointwise 95% confidence intervals.

encoder across eight Q-heads. We use Adam with learning rate 6.25×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.999)$, numerical stabilizer 1.5×10^{-4}, Huber loss, gradient-norm clipping at 10, discount 0.99, and replay capacity 10^6. Minibatches contain 32 transitions per head. Updates begin after 100,000 decision steps and occur every four steps; target networks synchronize every 10,000 steps. SOQ uses $\rho = 0.5$ and bootstrap membership probability 0.5, with the update specified in Appendix C.3.

Exploration is random for the first 100,000 steps. The ϵ-greedy schedule decreases linearly from 1 to 0.01 over the first million global training steps, then remains at 0.01. Every 50,000 training steps, the greedy policy is evaluated for ten episodes with fixed environment seeds and a limit of 5,000 decision steps per episode. Evaluation interactions are excluded from the training budget.

E.2 Aggregate AUC

Let $\overline{A}_{k,g}$ be the mean AUC across runs for algorithm k on game g, and let D_g be the corresponding original DQN mean AUC. The displayed index is

$$I_k = 100 + \frac{1}{10}\sum_{g=1}^{10} 100\frac{\overline{A}_{k,g} - D_g}{|D_g|}. \tag{32}$$

The 95% percentile confidence intervals use 50,000 bootstrap resamples of seeds within each algorithm–game group, preserving group sizes and holding the ten games and DQN references D_g fixed. We resample the numerator for every algorithm, including DQN.

E.3 Per-game Learning Curves

Figure 7 reports evaluation returns on Atari-10, with ten episodes per evaluation. Each seed trajectory is averaged within non-overlapping 0.5M-step intervals, then averaged across seeds and plotted at the interval endpoints. AUC is computed from the unbinned trajectories. Shaded regions show pointwise 95% percentile confidence intervals from 50,000 bootstrap resamples of whole seed trajectories within each algorithm–game group.

F Continuous-control Evaluation Details

F.1 Common Settings

We evaluate TD3 and TD3+Overlap on Gymnasium HalfCheetah-v4, Hopper-v4, and Walker2d-v4 with five seeds (0–4) per method and task and five million environment steps per run. Networks receive the low-dimensional state vector and use two 256-unit ReLU hidden layers. Actor outputs pass through $\tanh$ and are scaled to the action bounds; critics have independent parameters. Both methods use Adam with actor and critic learning rates 3×10^{-4}, batch size 256, replay capacity 10^6, discount 0.99, and target-update coefficient $\tau = 0.005$. Learning begins after 25,000 steps. Rewards are unmodified, and bootstrap targets use true environmental termination rather than time-limit truncation.

F.2 TD3+Overlap

TD3+Overlap retains TD3's delayed actor updates, target-policy smoothing, and Polyak target updates. It maintains $M = 8$ critics and two actors with independent parameters and optimizers: the behavior actor μ_b controls interaction and evaluation, while the selector actor μ_s learns bootstrap actions. The actors start from identical parameters and optimize

$$\mathcal{L}_b = -\mathbb{E}_s\left[\frac{1}{M}\sum_{i=1}^{M} Q_i(s, \mu_b(s))\right], \qquad \mathcal{L}_s = -\mathbb{E}_s\left[\frac{1}{N(s)}\sum_{i\in S_{N(s)}} Q_i(s, \mu_s(s))\right]. \tag{33}$$

Here S_N contains the first N critics in a random permutation fixed at initialization. The state-dependent size $N(s)$ is treated as a discrete control input to the selector loss. The behavior actor thus learns from all critics, while the selector learns from an adaptive subset.

Local SCR. For each state in an actor minibatch, form $K = 16$ candidate actions: the target behavior action $a_0 = \mu_b^-(s)$ and 15 independent uniform samples from the box $[a_0 - 0.3h, a_0 + 0.3h] \cap [l, u]$, where l, u are the action bounds and $h = (u - l)/2$. Define the reference candidate and empirical SCR by

$$k^* = \arg\max_{0\le k<K} \frac{1}{M}\sum_{i=1}^{M} Q_i^-(s, a_k), \tag{34}$$

$$\widehat{c}_{\mathrm{raw}}(s) = \frac{1}{(K-1)M^2}\sum_{k\neq k^*}\sum_{i=1}^{M}\sum_{j=1}^{M} H_\eta\big(Q_i^-(s, a_k) - Q_j^-(s, a_{k^*})\big), \tag{35}$$

where $H_\eta(x)$ equals 1 for $x > \eta$, $1/2$ for $|x| \le \eta$, and 0 otherwise, with $\eta = 10^{-6}$. Clip $\widehat{c}_{\text{raw}}$ to $[0, 1/2]$ to obtain $\widehat{c}$. The linear rule in Equation (9), with $\rho = 1/2$, becomes

$$N(s) = \text{clip}_{[1,M]}(\text{round}[M(1 - 2\widehat{c}(s))]) \,. \tag{36}$$

Lower SCR increases the selector subset, while higher SCR decreases it.

Bootstrap target and updates. The target selector actor generates the bootstrap action. With TD3's smoothing noise, the target is

$$\widetilde{a}' = \text{clip}_{[l,u]}\left(\mu_s^-(s') + h \odot \text{clip}_{[-0.5,0.5]}(\xi)\right), \quad \xi \sim \mathcal{N}(0, 0.2^2 I), \tag{37}$$

$$y = r + \gamma(1-d)\frac{1}{M}\sum_{j=1}^{M} Q_j^-(s', \widetilde{a}'), \tag{38}$$

where d indicates true termination. This replaces TD3's minimum of two target critics with the full-ensemble mean. The local SCR controls selector-actor training; the Bellman target directly uses its target policy. Each transition receives persistent, independent Bernoulli replay memberships with probability 0.5 for each critic. Each environment step updates every critic once, minimizing its mean squared Bellman error on an independent eligible minibatch. Every two critic rounds, both actors update using a batch from the full replay buffer, followed by Polyak updates of all target networks. Behavior actions use Gaussian exploration noise with standard deviation 0.1 in normalized action coordinates, scaled by h and clipped to the action bounds; evaluation uses μ_b without exploration noise.

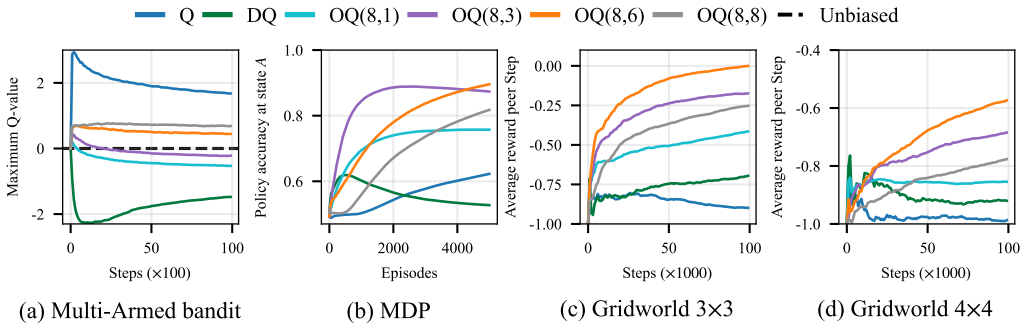
F.3 Evaluation

For Figure 6, evaluation occurs every 50,000 steps using ten episodes. Curves show arithmetic means across five seeds after non-overlapping 0.5M-step interval averaging, plotted at interval endpoints. Shaded regions indicate pointwise 95% percentile confidence intervals from 50,000 bootstrap resamples of whole seed trajectories within each method–task group.

G Additional Experiments

Throughout this section, OQ(M, N) denotes the fixed-overlap tabular algorithm defined in Appendix C.2, with M evaluators and a selector size of N. SOQ denotes Algorithm 1, which replaces the fixed selector size with the empirical SCR controller.

G.1 Effect of Selection Subset Size



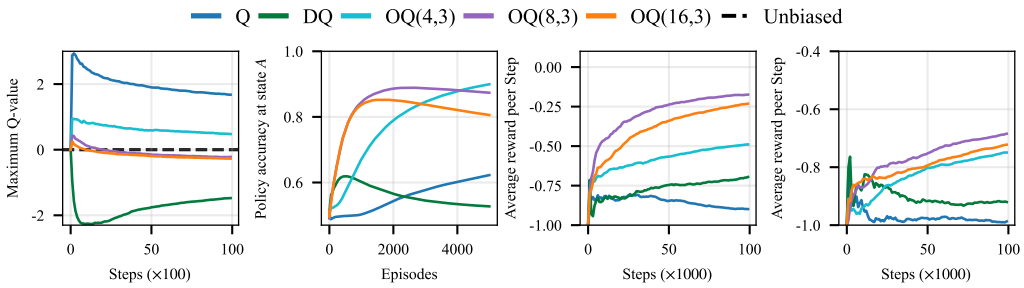
(a) Multi-Armed bandit (b) MDP (c) Gridworld 3×3 (d) Gridworld 4×4

Figure 8: OQ(M, N) with different N in Multi-Armed bandit, MDP, Gridworld 3×3 and Gridworld 4×4. All experimental results are averaged over 100 repeated runs.

With the evaluation ensemble fixed at $M = 8$, varying $N \in \{1, 3, 6, 8\}$ changes how many Q-functions participate in action selection. In the bandit, increasing N shifts OQ from underestimation

toward overestimation (Figure 8(a)). The intermediate settings also improve decisions: OQ$(8, 3)$ and OQ$(8, 6)$ achieve higher policy accuracy in the Meta Chain MDP and higher mean reward per step in both Gridworlds than OQ$(8, 1)$ and OQ$(8, 8)$ (Figures 8(b–d)). These comparisons show that the selection subset can improve bias and control performance without changing the evaluation ensemble size.

G.2 Effect of Evaluation Ensemble Size

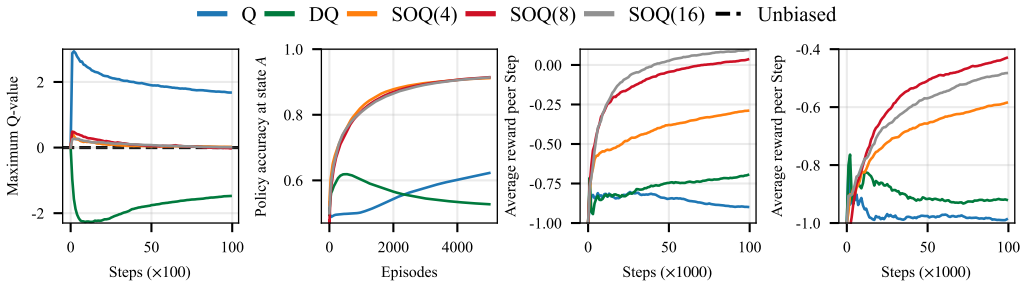


(a) Multi-Armed bandit (b) MDP (c) Gridworld 3×3 (d) Gridworld 4×4

Figure 9: OQ(M, N) with different M in Multi-Armed bandit, MDP, Gridworld 3×3 and Gridworld 4×4. All experimental results are averaged over 100 repeated runs.

Increasing the evaluation ensemble has a different effect when the selection subset remains fixed at $N = 3$. OQ$(4, 3)$ overestimates in the bandit, whereas OQ$(8, 3)$ and OQ$(16, 3)$ underestimate; the negative bias is larger for $M = 16$ than for $M = 8$ (Figure 9(a)). In the Meta Chain MDP, policy accuracy decreases as M grows, while OQ$(8, 3)$ achieves the highest mean reward per step in both Gridworlds (Figures 9(b–d)). Thus, a larger evaluation ensemble does not consistently improve performance at fixed N. Together with the preceding experiment, this motivates adapting the selection subset to the estimation conditions.

G.3 Adaptive Overlap across Ensemble Sizes



(a) Multi-Armed bandit (b) MDP (c) Gridworld 3×3 (d) Gridworld 4×4

Figure 10: SOQ(M) with different M in Multi-Armed bandit, MDP, Gridworld 3×3 and Gridworld 4×4. All experimental results are averaged over 100 repeated runs.

SOQ adapts N for each ensemble size, reducing the need to tune the selection subset manually. For $M \in \{4, 8, 16\}$, its final bandit estimates remain close to zero, with absolute biases between 0.012 and 0.021 (Figure 10(a) and Table 1). All three configurations achieve higher final policy accuracy in the Meta Chain MDP than the fixed-overlap configurations evaluated here. On the 4×4 Gridworld, SOQ(4) achieves a mean reward per step of -0.585, close to the best tested fixed-overlap result of -0.574 from OQ$(8, 6)$. SOQ(8) and SOQ(16) exceed all tested fixed-overlap configurations on

both Gridworlds (Figures 10(c,d)). These results support adapting overlap across different ensemble sizes to improve control performance.

Table 1: Final performance in tabular environments, averaged over 100 runs. Columns report the mean maximum Q-value in the bandit, optimal-action selection rate in the MDP, and mean reward per step in each Gridworld.

Algorithm	**Performance in different environments**			
	Multi-Armed	**MDP**	**Gridworld 3×3**	**Gridworld 4×4**
Q-learning	1.677	0.623	-0.898	-0.987
Double Q	-1.475	0.527	-0.695	-0.920
OQ(4,3)	0.477	0.899	-0.489	-0.749
OQ(8,1)	-0.530	0.757	-0.415	-0.855
OQ(8,3)	-0.220	0.874	-0.174	-0.685
OQ(8,6)	0.443	0.896	-0.001	-0.574
OQ(8,8)	0.686	0.818	-0.253	-0.776
OQ(16,3)	-0.272	0.806	-0.232	-0.722
SOQ(4)	0.021	0.912	-0.289	-0.585
SOQ(8)	-0.021	0.915	0.035	-0.430
SOQ(16)	0.012	0.914	0.095	-0.483