Appendix

A. Action-stability

Lemma 1 (Value-based stability)**.** *Value-based objectives are action stable since we can let $y^* = r$ and this minimizes the square loss at every action.*

Proof. Consider a datapoint $z = (x, r)$ which becomes $z(a) = (x, a, r(a))$ when we sample action a from the behavior. At this datapoint, the value-based objective for an estimated Q function $\hat{Q}$ is

$$\ell(z(a), \hat{Q}(x, a)) = (r(a) - \hat{Q}(x, a))^2 \tag{8}$$

This is minimized at all a by $\hat{Q}(x, a) = r(a)$. So setting $y^* = \hat{Q}(x, \cdot) = r$, we can exactly minimize ℓ at z. Since such a y^* exists, the objective is by definition action-stable. □

Lemma 2 (Policy-based instability)**.** *All policy-based objectives which take the form $\ell(z(a), \pi(a|x)) = f(z(a))\pi(a|x)$ are not action-stable at z unless $f(z(a)) > 0$ for exactly one action a.*

Proof. Consider a datapoint $z = (x, r)$ which becomes $z(a) = (x, a, r(a), p(a))$ when we sample action a from the behavior with probability $p(a)$. At this datapoint, a generic policy-based objective evaluated on a policy $\hat{\pi}$ takes the form

$$\ell(z(a), \hat{\pi}(a|x)) = f(z(a))\hat{\pi}(a|x) \tag{9}$$

As special examples of the function f we have the generic policy-based objective from Equation (4) when $f(z(a)) = \frac{r(a)}{p(a)}$. Moreover we can incorporate any baseline function $b(x)$ so that $f(z(a)) = \frac{r(a)-b(x)}{p(a)}$. This algorithm covers the one presented by Joachims et al. (2018).

Now to prove the claim, we have three cases: (1) $f(z(a)) < 0$ for all a, (2) $f(z(a)) > 0$ for at least two actions a_1, a_2, and (3) $f(z(a)) > 0$ at exactly one action a_1. We will show that in cases 1 and 2 the objective is action-unstable, but in case 3 it is action-stable.

Case 1. Assume that $f(z(a)) < 0$ for all a. Now for any given a to maximize the objective $f(z(a))\hat{\pi}(a|x)$ while ensuring that $\hat{\pi}(a|x)$ is a valid probability we must set $\hat{\pi}(a|x) = 0$. But, if we set $\hat{\pi}(a|x) = 0$ for all a, we no longer have a valid probability distribution, since $0 \notin \Delta^K$. Thus, we cannot find $y^* \in \Delta^K$ that optimizes the loss at z across all actions, so the objective is action-unstable.

Case 2. Assume that $f(z(a)) > 0$ for at least two actions a_1, a_2. Now at a_1, a_2 the objective $f(z(a))\hat{\pi}(a|x)$ is maximized by setting $\pi(a|x) = 1$. However, there is no valid element y of Δ^K such that $y(a_1) = 1$ and $y(a_2) = 1$. Thus, we cannot find $y^* \in \Delta^K$ that optimizes the loss at z across all actions, so the objective is action-unstable.

Case 3. Assume $f(z(a)) > 0$ at exactly one action a_1. Then at action a_1 we can maximize $f(z(a_1))\hat{\pi}(a_1|x)$ by setting $\hat{\pi}(a_1|x) = 1$. And since $f(z(a)) \leq 0$ for all other actions $a \neq a_1$, we can maximize $f(z(a))\hat{\pi}(a|x)$ by setting $\hat{\pi}(a|x) = 0$. Now since $\mathbb{1}[a = a_1] \in \Delta^K$, there does exist a vector $y^* \in \mathcal{Y}$ which exactly optimizes ℓ regardless of which action is sampled. So, the objective is action-stable if and only if we are in this case. □

B. Value-based learning

Theorem 1 (Reduction to regression)**.** *By Assumption 1 we have $\beta(a|x) \geq \tau$ for all x, a. Then with $\hat{Q}_{S_B}$ as defined in (5) we have*

$$V(\pi^*) - V(\pi_{\hat{Q}_{S_B}}) \leq \frac{2}{\sqrt{\tau}} \sqrt{\mathop{\mathbb{E}}_{x,a\sim\beta}[(Q(x, a) - \hat{Q}_{S_B}(x, a))^2]}.$$

Proof. The proof follows directly from linking the subsequent lemmas with $\hat{\pi} = \pi_{\hat{Q}_{S_B}}$ and Π be the set of all policies in Lemma 3. □

Lemma 3 (Mismatch: from MSE to Regret)**.** *Assume strict positivity. Let $\hat{\pi}$ be the greedy policy with respect to some $\hat{Q}$ and let Π be any class of policies to compete against, which contains $\hat{\pi}$. Then*

$$\sup_{\pi \in \Pi} V(\pi) - V(\hat{\pi}) \leq 2\sqrt{\sup_{\pi \in \Pi} \mathbb{E}_{x,a \sim \mathcal{D}, \pi}[(Q(x,a) - \hat{Q}(x,a))^2]} \tag{10}$$

Proof. We can expand the definition of regret and then add and subtract and apply a few inequalities. Let $\bar{\pi}$ be the policy in Π which maximizes V. Then

$$\begin{aligned}
\sup_{\pi \in \Pi} V(\pi) - V(\hat{\pi}) &= \mathbb{E}_x \Big[\mathbb{E}_{a \sim \bar{\pi}|x}[Q(x,a)] - \mathbb{E}_{a \sim \hat{\pi}|x}[Q(x,a)] \Big] && (11)\\
&= \mathbb{E}_x \Big[\mathbb{E}_{a \sim \bar{\pi}|x}[Q(x,a)] - \mathbb{E}_{a \sim \hat{\pi}|x}[\hat{Q}(x,a)] + \mathbb{E}_{a \sim \hat{\pi}|x}[\hat{Q}(x,a)] - \mathbb{E}_{a \sim \hat{\pi}|x}[Q(x,a)] \Big] && (12)\\
&\leq \mathbb{E}_x \Big[\mathbb{E}_{a \sim \bar{\pi}|x}[|Q(x,a) - \hat{Q}(x,a)|] + \mathbb{E}_{a \sim \hat{\pi}|x}[|Q(x,a) - \hat{Q}(x,a)|] \Big] && (13)\\
&\leq \sqrt{\mathbb{E}_x \mathbb{E}_{a \sim \bar{\pi}|x}[(Q(x,a) - \hat{Q}(x,a))^2]} + \sqrt{\mathbb{E}_x \mathbb{E}_{a \sim \hat{\pi}|x}[(Q(x,a) - \hat{Q}(x,a))^2]} && (14)\\
&\leq 2\sqrt{\sup_{\pi \in \Pi} \mathbb{E}_x \big[\mathbb{E}_{a \sim \pi|x}[(Q(x,a) - \hat{Q}(x,a))^2] \big]} && (15)
\end{aligned}$$

The first inequality holds since $\hat{\pi}$ maximizes $\hat{Q}$ and by using the definition of absolute value, the second by Jensen, and the third by introducing the supremum. □

Lemma 4 (Transfer: from β to π)**.** *Assume strict positivity and take any Q-function $\hat{Q}$ and any policy π, then*

$$\mathbb{E}_{x,a \sim \mathcal{D}, \pi}[Q(x,a) - \hat{Q}(x,a))^2] < \frac{1}{\tau} \Big(\mathbb{E}_{x,a \sim \mathcal{D}, \beta}[(Q(x,a) - \hat{Q}(x,a))^2] \Big). \tag{16}$$

Proof. Let π be any policy. Then

$$\begin{aligned}
\mathbb{E}_x \mathbb{E}_{a \sim \pi|x}[(Q(x,a) - \hat{Q}(x,a))^2] &= \int_x p(x) \sum_a \pi(a|x)(Q(x,a) - \hat{Q}(x,a))^2 dx && (17)\\
&= \int_x \sum_a \pi(a|x) \frac{\beta(a|x)}{\beta(a|x)} p(x)(Q(x,a) - \hat{Q}(x,a))^2 dx && (18)\\
&< \frac{1}{\tau} \int_x \sum_a \beta(a|x) p(x)(Q(x,a) - \hat{Q}(x,a))^2 dx && (19)\\
&= \frac{1}{\tau} \mathbb{E}_{x,a \sim \mathcal{D}, \beta}[(Q(x,a) - \hat{Q}(x,a))^2] && (20)
\end{aligned}$$

where we use a multiply and divide trick and apply the definition of strict positivity to ensure that $\frac{\pi(a|x)}{\beta(a|x)} < \frac{1}{\tau}$. □

C. Policy-based learning

C.1. In-sample regret

Lemma 5. *Let Π be an interpolating class and $K = 2$. Then there exists a π_B as defined in Equation (4) such that*

1. *the behavior of π_B at each datapoint $x_i \in S$ only depends on $a_i, r_i(a_i)$, and p_i*
2. *$\pi_B(\cdot|x_i) \in \{(0,1),(1,0)\}$.*

Proof. We will begin by proving part 2. Note that the objective that π_B optimizes takes the form $\frac{r_i(a_i)}{p_i}\pi(a_i|x_i)$ at each datapoint. Since probabilities are constrained to $[0,1]$ this is optimized by $\pi(a_i|x_i)=0$ if $\frac{r_i(a_i)}{p_i}<0$ and $\pi(a_i|x_i)=1$ if $\frac{r_i(a_i)}{p_i}>0$. Since we have an overparameterized model class, we know that Π contains a π_B that can exactly choose the optimizer at each datapoint. Since $K=2$, once we know $\pi(a_i|x_i)$ we immediately have $\pi(\bar{a}_i|x_i)=1-\pi(a_i|x_i)$ (where $\hat{a}_i$ is the action that is not equal to a_i). Thus $\pi_B(\cdot|x_i)\in\{(0,1),(1,0)\}$.

Now part 1 follows directly since the above reasoning showed that $\pi_B(\cdot|x_i)$ is defined precisely by the sign of $\frac{r_i(a_i)}{p_i}$ and the identity of a_i. □

Theorem 2 (In-sample regret lower bound)**.** *Let $K=2$ and the policy class be overparameterized. Define $\Delta_r(x)=\big|\mathbb{E}_{r|x}[r(1)-r(2)]\big|$ as the absolute expected gap in rewards at x. Define $p_u(x)$ to be the probability that the policy-based objective is action-unstable at x. Recall that $\beta(a|x)\geq\tau$ by Assumption 1. Then*

$$\mathbb{E}_S[V(\pi^*;S)-V(\pi_B;S)]\geq\tau\mathbb{E}_x\big[p_u(x)\Delta_r(x)\big].$$

Proof. By part 1 of Lemma 5 and linearity of expectation we can decompose the expected in-sample value as

$$\mathbb{E}_S[V(\pi^*;S)-V(\pi_B;S)]=\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{x_i,r_i,a_i}\Big[\mathbb{E}_{a\sim\pi^*}\mathbb{E}_{r|x_i}[r(a)]-\mathbb{E}_{a\sim\pi_B}\mathbb{E}_{r|x_i}[r(a)]\Big].$$

Since the data are iid we further have that

$$\mathbb{E}_S[V(\pi^*;S)-V(\pi_B;S)]=\mathbb{E}_{x_i,r_i,a_i}\Big[\mathbb{E}_{a\sim\pi^*}\mathbb{E}_{r|x_i}[r(a)]-\mathbb{E}_{a\sim\pi_B}\mathbb{E}_{r|x_i}[r(a)]\Big].$$

Define the event $U_{x,r}$ to be the event that the policy-based objective is action-unstable at x,r. So $p_u(x)=\mathbb{E}_{r|x}[\mathbb{1}[U_{x,r}]]$. We can split this expectation up into stable and unstable parts by conditioning on either $\overline{U}_{x_i,r_i}$ or U_{x_i,r_i}, and lower bound the regret on the stable datapoints by 0:

$$\begin{aligned}\mathbb{E}_S[V(\pi^*;S)-V(\pi_B;S)]&=\mathbb{E}_{x_i,r_i|\overline{U}_{x_i,r_i}}\mathbb{E}_{a_i|x_i}\Big[\mathbb{E}_{a\sim\pi^*}\mathbb{E}_{r|x_i}[r(a)]-\mathbb{E}_{a\sim\pi_B}\mathbb{E}_{r|x_i}[r(a)]\Big]\\&\quad+\mathbb{E}_{x_i,r_i|U_{x_i,r_i}}\mathbb{E}_{a_i|x_i}\Big[\mathbb{E}_{a\sim\pi^*}\mathbb{E}_{r|x_i}[r(a)]-\mathbb{E}_{a\sim\pi_B}\mathbb{E}_{r|x_i}[r(a)]\Big]\\&\geq\mathbb{E}_{x_i,r_i|U_{x_i,r_i}}\mathbb{E}_{a_i|x_i}\Big[\mathbb{E}_{a\sim\pi^*}\mathbb{E}_{r|x_i}[r(a)]-\mathbb{E}_{a\sim\pi_B}\mathbb{E}_{r|x_i}[r(a)]\Big].\end{aligned}$$

By part 2 of Lemma 5 we know that $\pi_B(\cdot|x_i)$ is either $(1,0)$ or $(0,1)$. Conditioned on the objective being unstable at x_i and using the fact that there are only two actions, we know that $\pi_B(x_i)$ must be different depending on whether $a_i=1$ or $a_i=2$. Define $a^1_{i,B}$ to be the action that π_B selects at x_i when $a_i=1$ and $a^2_{i,B}$ the action when $a_i=2$. Let a^*_i be the action chosen by the deterministic optimal policy π^* at x_i. Thus we can split the expectation over a_i in the above expression and then plug in definitions to get:

$$\mathbb{E}_S[V(\pi^*;S)-V(\pi_B;S)]\geq\mathbb{E}_{x_i,r_i|U_{x_i,r_i}}\Big[\beta(a_i=1|x_i)\mathbb{E}_{r|x_i}[r(a^*_i)-r(a^1_{i,B})]+\beta(a_i=2|x_i)\mathbb{E}_{r|x_i}[r(a^*_i)-r(a^2_{i,B})]\Big].$$

Since we assumed that $\beta(a|x_i)\geq\tau$ for all a we can lower bound the above by

$$\mathbb{E}_S[V(\pi^*;S)-V(\pi_B;S)]\geq\tau\mathbb{E}_{x_i,r_i}\Big[\mathbb{1}[U_{x_i,r_i}]\Big(\mathbb{E}_{r|x_i}[r(a^*_i)-r(a^1_{i,B})]+\mathbb{E}_{r|x_i}[r(a^*_i)-r(a^2_{i,B})]\Big)\Big].$$

Finally, we note that since $a^1_{i,B} \neq a^2_{i,B}$ and there are only 2 actions that the above is precisely

$$\begin{aligned}\mathbb{E}_S[V(\pi^*;S) - V(\pi_B;S)] &\geq \tau \mathbb{E}_{x_i,r_i}\left[\mathbb{1}[\overline{E}_{x_i,r_i}]\mathbb{E}_{r|x_i}[r(a_i^*) - r(a \neq a_i^*)]\right] \\ &= \tau \mathbb{E}_{x_i,r_i}[\mathbb{1}[\overline{E}_{x_i,r_i}]\Delta_r(x_i)] \\ &= \tau \mathbb{E}_{x_i}[\mathbb{E}_{r_i|x_i}[\mathbb{1}[U_{x_i,r_i}]]\Delta_r(x_i)] \\ &= \tau \mathbb{E}_{x_i}[p_u(x_i)\Delta_r(x_i)] \\ &= \tau \mathbb{E}_x[p_u(x)\Delta_r(x)].\end{aligned}$$

□

C.2. Connection to noisy classification

This section states and proves the Theorem referenced in the main text connecting action-unstable policy-based learning with noisy classification.

Theorem 4 (Noisy classification reduction)**.** *Take any noise level $\eta < 1/2$ and any binary classification problem $\mathcal{C}$ consisting of a distribution $\mathcal{D}_\mathcal{C}$ over $\mathcal{X}$ and a labeling function $y_\mathcal{C} : \mathcal{X} \to \{-1, 1\}$. There exists an offline contextual bandit problem $\mathcal{B}$ with noiseless rewards such that*

1. *Maximizing $\hat{V}_B$ in $\mathcal{B}$ is equivalent to minimizing the 0/1 loss on a training set drawn from $\mathcal{C}$ where labels are flipped with probability η.*
2. *Maximizing $\hat{V}_F$ in $\mathcal{B}$ is equivalent to minimizing the 0/1 loss on a training set drawn from $\mathcal{C}$ with noiseless training labels.*

Proof. First we will construct the bandit problem $\mathcal{B}$ with two actions corresponding to the classification problem $\mathcal{C}$. For any constant $c_r > 0$ we define $\mathcal{B}$ by

$$x \sim \mathcal{D}_\mathcal{C}, \qquad r|x = \begin{cases} c_r(1-\eta, \eta) & y_\mathcal{C}(x) = 1 \\ c_r(\eta, 1-\eta) & y_\mathcal{C}(x) = -1 \end{cases}, \qquad \beta(1|x) = \begin{cases} 1-\eta & y_\mathcal{C}(x) = 1 \\ \eta & y_\mathcal{C}(x) = -1 \end{cases} \tag{21}$$

Now we will show that in this problem, $\hat{V}_B$ is equivalent to the 0/1 loss for $\mathcal{C}$ with noisy labels. To do this first note that by construction, for x with $y_\mathcal{C}(x) = 1$ we have $\frac{r(1)|x}{\beta(1|x)} = \frac{c_r(1-\eta)}{1-\eta} = c_r$ and $\frac{r(2)|x}{\beta(2|x)} = \frac{c_r\eta}{\eta} = c_r$, and similarly for x with $y_\mathcal{C}(x) = -1$ we have $\frac{r(1)|x}{\beta(1|x)} = \frac{c_r\eta}{\eta} = c_r$ and $\frac{r(2)|x}{\beta(2|x)} = \frac{c_r(1-\eta)}{1-\eta} = c_r$.

$$\hat{V}_B(\pi) = \frac{1}{N}\sum_{i=1}^N r_i(a_i)\frac{\pi(a_i|x_i)}{\beta(a_i|x_i)} = \frac{1}{N}\sum_{i=1}^N \frac{r_i(a_i)}{\beta(a_i|x_i)}\pi(a_i|x_i) \tag{22}$$

$$= \frac{c_r}{N}\sum_{i=1}^N \pi(a_i|x_i) \tag{23}$$

This is equivalent to 0/1 loss with noisy labels since β generates a_i according to $y_\mathcal{C}$ where the label is flipped with probability η.

Now we will show that $\hat{V}_F$ is equivalent to the 0/1 loss for $\mathcal{C}$ with clean labels. Note that by construction $r(a)|x = c_r\eta + \pi^*(a|x)c_r(1-2\eta)$. So,

$$\hat{V}_F(\pi) = \frac{1}{N}\sum_{i=1}^N \langle r_i, \pi(\cdot|x_i)\rangle = \frac{c_r}{N}\sum_{i=1}^N \langle \eta\mathbf{1} + (1-2\eta)\pi^*(\cdot|x_i), \pi(\cdot|x_i)\rangle \tag{24}$$

$$= \frac{c_r\eta}{N} + \frac{c_r(1-2\eta)}{N}\sum_{i=1}^N \langle \pi^*(\cdot|x_i), \pi(\cdot|x_i)\rangle \tag{25}$$

This is equivalent to 0/1 loss with noisy labels since π^* exactly corresponds to $y_\mathcal{C}$. □

C.3. Nearest Neighbor

Theorem 3 (Regret lower bound for one nearest neighbor)**.** *Let $\Delta_r = r_{\max} - r_{\min}$. Then there exist problem instances with noiseless rewards where*

$$\limsup_{N\to\infty} \mathbb{E}_S[V(\pi_F) - V(\pi_B)] = \frac{\Delta_r}{2},$$

but

$$\limsup_{N\to\infty} \mathbb{E}_S[V(\pi^*) - V(\pi_F)] = 0.$$

Proof. First we need to formally define the nearest neighbor rules that interpolate the objectives $\hat{V}_B$ and $\hat{V}_F$. These are simple in the case of two actions. Let $i(x)$ be the index of the nearest neighbor to x in the dataset. Then

$$\pi_B(a|x) = \begin{cases} 1 & \big(a = a_{i(x)} \text{ AND } r_{i(x)}(a_{i(x)}) > 0\big) \text{OR} \big(a \neq a_{i(x)} \text{ AND } r_{i(x)}(a_{i(x)}) \leq 0\big) \\ 0 & otherwise. \end{cases} \tag{26}$$

This is saying that π_B chooses the same action as the observed nearest neighbor if that reward was positive, and the opposite action if that was negative. And for the full feedback we just choose the best action from the nearest datapoint.

$$\pi_F(a|x) = \begin{cases} 1 & a = \arg\max_{a'} r_{i(x)}(a') \\ 0 & otherwise. \end{cases} \tag{27}$$

Now we can construct the problem instances needed for the Theorem. To construct the example, take a bandit problem with two actions (called 1 and 2):

$$x \sim U([-1,1]), \quad r|x = (1, 1+\Delta_r), \quad \beta(1|x) = \beta(2|x) = 1/2 \,\forall\, x, a$$

The true optimal policy has $\pi^*(2|x) = 1$ for all x and $V(\pi^*) = 1 + \Delta_r$. The policy with full feedback π_F is to always choose action 2, since every observation will show that action 2 is better.

Now, we will show that in the limit of infinite data, π_F has no regret. Since the rewards are noiseless, the maximum observed reward at a context is exactly the optimal action at that context. Thus, we precisely have a classification problem with noiseless labels so that the Bayes risk is 0. Since we π^* is continuous, the class conditional densities (determined by the indicator of the argmax of Q) are piecewise continuous. This allows us to apply the classic result of (Cover & Hart, 1967) that a nearest neighbor rule has asymptotic risk less twice the Bayes risk, which in this case is zero. This means that asymptotically $P(\pi_F(a|x) \neq \pi^*(a|x)) = 0$ which immediately gives the second desired result of zero regret in the limit of infinite data under full feedback.

Now we note that since rewards are always positive, we can simplify the definition of π_B as

$$\pi_B(a|x) = \mathbb{1}[a = a_{i(x)}]. \tag{28}$$

Then we have that

$$\begin{aligned} V(\pi_F) - V(\pi_B) &= \mathbb{E}_x[\mathbb{E}_{a\sim\pi_F|x}[Q(x,a)] - \mathbb{E}_{a\sim\pi_B|x}[Q(x,a)]] && (29)\\ &= \mathbb{E}_x[\Delta_r + 1 - (\pi_B(1|x) + \pi_B(2|x)(\Delta_r + 1))]] && (30)\\ &= \Delta_r + 1 - \mathbb{E}_x[\mathbb{1}[a_{i(x)} = 1] + (\Delta_r + 1)\mathbb{1}[a_{i(x)} = 2]] && (31) \end{aligned}$$

Taking expectation over S we get

$$\begin{aligned} \mathbb{E}_S[V(\pi_F) - V(\pi_B)] &= \mathbb{E}_S[\Delta_r + 1 - \mathbb{E}_x[\mathbb{1}[a_{i(x)} = 1] + (\Delta_r + 1)\mathbb{1}[a_{i(x)} = 2]]] && (32)\\ &= \Delta_r + 1 - \mathbb{E}_x[P_S(a_{i(x)} = 1) + (\Delta_r + 1)P_S(a_{i(x)} = 2)]] && (33)\\ &= \Delta_r + 1 - \mathbb{E}_x[\frac{1}{2} + (\Delta_r + 1)\frac{1}{2}]] && (34)\\ &= \frac{\Delta_r}{2} && (35) \end{aligned}$$

This construction did not depend on the size of the dataset, so it is even true as the number of datapoints tends to infinity. □

D. Discussion of doubly robust algorithms

Before going into the comparison, we will define the doubly robust algorithm (Dudík et al., 2011) in our notation. Specifically,

$$\widehat{V}_{DR}(\pi) := \sum_{i=1}^{N}\left[\sum_{a}\pi(a|x_i)\hat{Q}(x_i,a) + \frac{\pi(a_i|x_i)}{\beta(a_i|x_i)}(r_i(a_i) - \hat{Q}(x_i,a_i))\right], \qquad \hat{\pi}_{DR} = \arg\max_{\pi\in\Pi}\widehat{V}_{DR}(\pi) \tag{36}$$

As stated in the main text, when we use overparameterized models and train $\hat{Q}$ on the same data that we use to optimize the policy, then doubly robust methods are equivalent to the vanilla value-based algorithm. This is formalized in Lemma 6 below.

This equivalence can be avoided by using crossfitting so that $\hat{Q}$ is not trained on the same data as π. However, then it is possible that the doubly robust policy objective becomes action-unstable. This is true *even with* access to the true Q function, but requires stochastic rewards. To construct such an example we leverage the stochastic rewards so that instability only occurs at datapoints where certain reward vectors are sampled. This is shown in Lemma 7 below.

One final point is to consider the motivation for doubly robust methods. Usually it is motivated by concerns about consistency of the value function estimation or estimation of behavior policy (Dudík et al., 2011). However, in our setting we have (1) an overparamterized model class which is large enough to contain the true value function, and (2) exact access to the behavior probabilities. So it is not clear why doubly robust methods would be motivated in our setting.

Lemma 6 (Equivalence of DR and vanilla VB)**.** *When we use overparameterized models and do not use crossfitting, doubly robust learning from Equation (36) is equivalent to vanilla value-based learning from Equation (5).*

Proof. When the model for $\hat{Q}$ is overparameterized and trained on the full dataset, we know that $\hat{Q}(x_i,a_i) = r_i(a_i)$. Thus we get that

$$\widehat{V}_{DR}(\pi) = \sum_{i=1}^{N}\left[\sum_{a}\pi(a|x_i)\hat{Q}(x_i,a) + \frac{\pi(a_i|x_i)}{\beta(a_i|x_i)}(r_i(a_i) - \hat{Q}(x_i,a_i))\right] \tag{37}$$

$$= \sum_{i=1}^{N}\left[\sum_{a}\pi(a|x_i)\hat{Q}(x_i,a) + \frac{\pi(a_i|x_i)}{\beta(a_i|x_i)}(0)\right] \tag{38}$$

$$= \sum_{i=1}^{N}\sum_{a}\pi(a|x_i)\hat{Q}(x_i,a) \tag{39}$$

With an overparameterized policy class, we can exactly recover the greedy policy relative to $\hat{Q}$ to optimize this objective. □

Lemma 7 (Instability of DR)**.** *There exist problems with stochastic rewards where even with access to the exact Q function, the doubly robust policy objective is action-unstable with probability 1/2.*

Proof. We need only consider one datapoint since the action-stability property is defined on a per datapoint basis. To make this construction we will consider only two actions.

$$r|x = \begin{cases}(0,1) & w.p.\ 1/2\\ (0,-2) & otherwise\end{cases}, \qquad \beta(\cdot|x) = (1/2, 1/2) \tag{40}$$

So, we know that

$$Q(\cdot|x) = (0, -0.5) \tag{41}$$

Now we claim that when the sampled datapoint has $r = (0,1)$ the doubly robust objective is action-unstable (and this happens with probability 1/2 by construction). We can explicitly expand the DR objective for the policy π at x when action a is sampled

$$\ell_{DR}(\pi, x, a, r) = \pi(1|x)\cdot 0 + \pi(2|x)\cdot(-0.5) + \frac{\pi(a|x)}{1/2}(r(a) - Q(x,a)) \tag{42}$$

So when $a = 1$ we have $r(a) = 0$ and $Q(x, a) = 0$ so that

$$\ell_{DR}(\pi, x, a, r) = \pi(2|x) \cdot (-0.5) + 2 \cdot \pi(1|x)(0 - 0) = \pi(2|x) \cdot (-0.5) \tag{43}$$

And when $a = 2$ we have $r(a) = 1$ (because that was the sampled reward) and $Q(x, a) = 0$ so that

$$\ell_{DR}(\pi, x, a, r) = \pi(2|x) \cdot (-0.5) + 2 \cdot \pi(2|x)(1 - (-0.5)) = \pi(2|x) \cdot (2.5) \tag{44}$$

Now, this is clearly action-unstable since the optimizer when $a = 1$ is sampled is $\pi(\cdot|x) = (1, 0)$ while when $a = 2$ is sampled we get $\pi(\cdot|x) = (0, 1)$. □

E. Experiments

E.1. Synthetic data

Data. As described in the main text we sample some hidden reward matrix θ and then sample contexts and rewards from isotropic Gaussians:

$$\theta \sim U([0, 1]^{K \times d}), \quad x \sim \mathcal{N}(0, I_d), \quad r \sim \mathcal{N}(\theta x, \epsilon I_d).$$

Actions are sampled according to a uniform behavior:

$$a \sim \beta(\cdot|x) = U(\{1, \ldots, K\}).$$

We set $K = 2, d = 10, \epsilon = 0.1$. For each random seed we take $N = 100$ training points and sample an independent test set of 500 points. For experiment 1 we sample θ and one dataset of x, r tuples, then we sample 20 independent sets of actions. For experiment 2 we sample all parameters separately to construct each of the 50 datasets.

Model. For policies and Q functions we use a multilayer perceptron with one hidden layer of width 512 and ReLU activations. The only difference between policy and Q architecture is that the policy has a softmax layer on the output so that the output is a probability distribution.

Learning. We train using SGD with momentum. Learning rate is 0.01, momentum is 0.9, batch size is 10, and weight decay is 0.0001. We train every model for 1000 epochs decreasing the learning rate by a factor of 10 after 200 epochs. This trains well past the point of convergence in our experience.

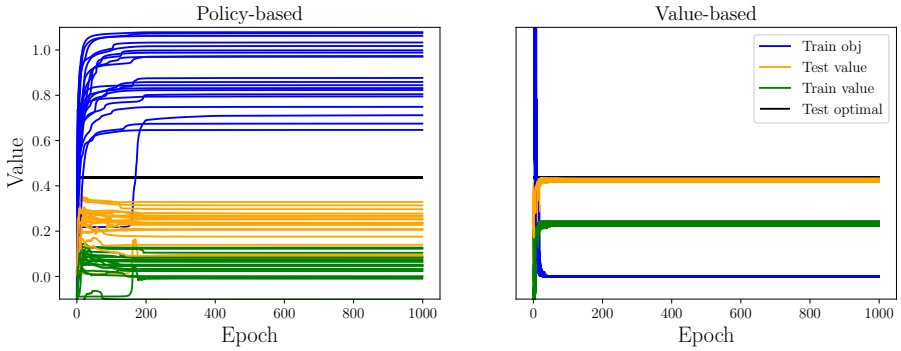

Figure 4. We show learning curves across each of the twenty different action resampled datasets.

Extended results. Figure 4 shows learning curves for each of the twenty different action datasets from experiment 1. We use "train obj" to refer to the training objective which is squared error for value-based learning and $\hat{V}_B$ for policy-based learning. We use "train value" and "test value" to refer to $V(\pi; S)$ for S being the train and test sets respectively. We can evaluate the true value at each datapoint since we know the full reward vector at each datapoint.

We see that the policy-based objective is dramatically higher than the highest achievable value due to overfitting of the noise in the actions. The gap between train and test value is mot likely explained by noise in the contexts sampled in those respective datasets (by chance the test set has higher value contexts).

E.2. CIFAR-10

Data. We use a bandit version of the CIFAR-10 dataset (Krizhevsky, 2009). We split the train set into a train set of the first 45000 examples and validation set of the last 5000. We normalize the images and use data augmentation of random flips and crops of size 32. Each of the 10 labels becomes an action. We define rewards to be 1 for a correct prediction and 0 for an incorrect prediction. We use two different behavior policies. One is a uniform behavior that selects each action with probability 0.1 and the other is the hand-crafted behavior policy from (Joachims et al., 2018).

Model. We use a ResNet-18 (He et al., 2016) from PyTorch (Paszke et al., 2019) for both the policy and the Q function. The only modification we make to accommodate for the smaller images in CIFAR is to remove the first max-pooling layer.

Learning. We train using SGD with momentum 0.9,a batch size 128, and weight decay of 0.0001 for 1000 epochs. Training takes about 20 hours for each run on an NVIDIA RTX 2080 Ti GPU. We use a learning rate of 0.1 for the first 200 epochs, 0.01 for the next 200, and 0.001 for the last 600. To improve stability we use gradient clipping and reduce the learning rate in the very first epoch to 0.01.

Extended results. Figures 5 and 6 show learning curves for each of the three algorithms we consider across each dataset. The labels refer to the same quantities as they did on the synthetic problem.

One interesting phenomena is that the unstable policy-based algorithm displays a clear overfitting phenomena as we would predict due to the noise in the actions being transferred into noise in the objective. Since we have strictly positive rewards here, this is also an instance of "propensity overfitting" (Swaminathan & Joachims, 2015b). As a result, limiting the capacity of the model class by early stopping could improve performance somewhat. But by limiting capacity in this way we are exiting the overparameterized/interpolating regime described by Zhang et al. (2016).

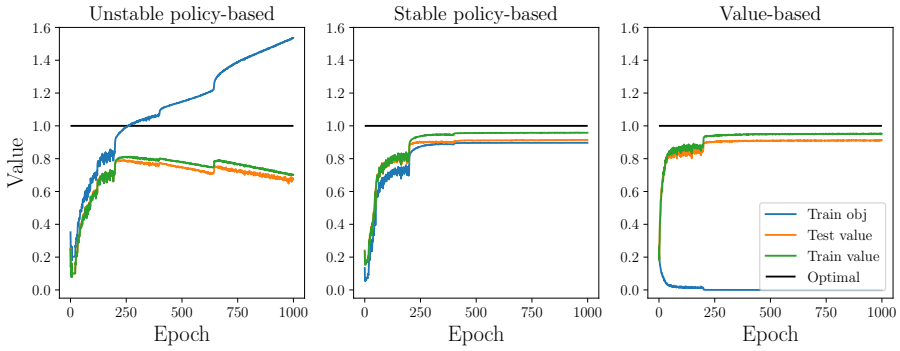

Figure 5. Learning curves on the hand-crafted action dataset.

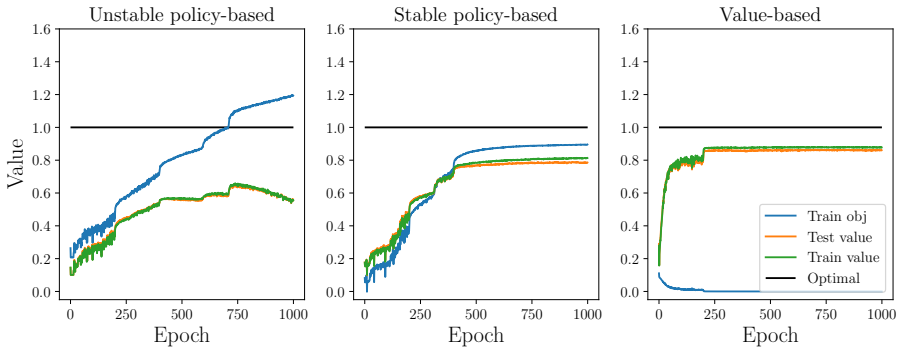

Figure 6. Learning curves on the uniform action dataset.

F. Small model classes

In this section we state and prove theorems that bound each term of our regret decomposition for each algorithm we consider when we use finite model classes. Similar results can be shown for other classical notions of model class complexity. We include these results for completeness, but the main focus of our paper is the overparameterized regime where such bounds are vacuous.

Theorem 5 (Policy-based learning with a small model class)**.** *Assume strict positivity and a finite policy class* Π*. Let* $\varepsilon_\Pi = V(\pi^*) - \sup_{\pi\in\Pi} V(\pi)$*. Denote* $\Delta_r = r_{max} - r_{min}$*. Then we have that for any* $\delta > 0$ *with probability* $1-\delta$ *each of the following holds:*

$$\textit{Approximation Error} = V(\pi^*) - \sup_{\pi\in\Pi} V(\pi) \leq \varepsilon_\Pi$$

$$\textit{Estimation Error} = \sup_{\pi\in\Pi} V(\pi) - V(\pi_F) \leq 2\Delta_r\sqrt{\frac{\log(2|\Pi|/\delta)}{2N}}$$

$$\textit{Bandit Error} = V(\pi_F) - V(\pi_B) \leq \frac{2\Delta_r}{\tau}\sqrt{\frac{\log(2|\Pi|/\delta)}{2N}}$$

Proof. The bound on approximation error follows directly from the definition of ε_Π. The bound on the estimation error follows from a standard application of a Hoeffding bound on the random variables $X_i = \langle r_i, \pi(\cdot|x_i)\rangle$ which are bounded by Δ_r and a union bound over the policy class.

The bound on bandit error essentially follows Theorem 3.2 of (Strehl et al., 2010), we include a proof for completeness:

$$\begin{aligned}
V(\pi_F) - V(\pi_B) &= V(\pi_F) - \hat{V}_B(\pi_B) + \hat{V}_B(\pi_B) - V(\pi_B) \\
&\leq V(\pi_F) - \hat{V}_B(\pi_F) + \hat{V}_B(\pi_B) - V(\pi_B) \\
&\leq 2\sup_{\pi\in\Pi} |V(\pi) - \hat{V}_B(\pi)| \\
&\leq \frac{2\Delta_r}{\tau}\sqrt{\frac{\log(2|\Pi|/\delta)}{2N}}
\end{aligned}$$

The first inequality comes from the definition of π_B. The second comes since both $\pi_F, \pi_B \in \Pi$. And the last inequality follows from an application of a Hoeffding bound on the random variables $X_i = r_i(a_i)\frac{\pi(a_i|x_i)}{p_i}$ which are bounded by $\frac{\Delta_r}{\tau}$ and a union bound over the policy class. □

Theorem 6 (Value-based learning with a small model class)**.** *Assume strict positivity and a finite function class* $\mathcal{Q}$ *which induces a finite class of greedy policies* $\Pi_\mathcal{Q}$*. Let* $\varepsilon_\mathcal{Q} = \inf_{\widehat{Q}\in\mathcal{Q}} \mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \widehat{Q}(x,a))^2]$*. Denote* $\Delta_r = r_{max} - r_{min}$*. Then we have that for any* $\delta > 0$ *with probability* $1-\delta$ *each of the following holds:*

$$\textit{Approximation Error} = V(\pi^*) - \sup_{\pi\in\Pi_\mathcal{Q}} V(\pi) \leq 2\sqrt{\varepsilon_\mathcal{Q}/\tau} \tag{45}$$

$$\textit{Estimation Error} = \sup_{\pi\in\Pi_\mathcal{Q}} V(\pi) - V(\pi_F) \leq 2\Delta_r\sqrt{\frac{\log(|\mathcal{Q}|/\delta)}{2N}} \tag{46}$$

$$\textit{Bandit Error} = V(\pi_F) - V(\pi_{\widehat{Q}}) \leq \frac{10\Delta_r}{\sqrt{\tau}}\sqrt{\frac{\log(|\mathcal{Q}|/\delta)}{N}} + 6\sqrt{\Delta_r}\left(\frac{\log(|\mathcal{Q}|/\delta)}{\tau N}\varepsilon_\mathcal{Q}\right)^{1/4} + 2\sqrt{\varepsilon_\mathcal{Q}/\tau} \tag{47}$$

Proof. To bound the approximation error, we can let $\hat{\pi}$ be the greedy policy associated with a Q-function $\widehat{Q}$ and apply Lemmas 3 and 4. This gives us

$$V(\pi^*) - \sup_{\hat{\pi}\in\Pi_\mathcal{Q}} V(\hat{\pi}) = \inf_{\widehat{Q}\in\mathcal{Q}} [V(\pi^*) - V(\hat{\pi})] \leq \inf_{\widehat{Q}\in\mathcal{Q}} \frac{2}{\sqrt{\tau}}\sqrt{\mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \widehat{Q}(x,a))^2]} = 2\sqrt{\varepsilon_\mathcal{Q}/\tau}. \tag{48}$$

The bound on the estimation error follows the same as before from standard uniform convergence arguments.

The bound on the bandit error follows by again applying Lemmas 3 and 4 and then making the concentration argument from Lemma 16 of (Chen & Jiang, 2019). Explicitly, our Lemmas give us

$$V(\pi_F) - V(\pi_{\widehat{Q}}) \leq V(\pi^*) - V(\pi_{\widehat{Q}}) \leq \frac{2}{\sqrt{\tau}} \sqrt{\mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \widehat{Q}(x,a))^2]}. \tag{49}$$

Then, to bound the squared error term, we can add and subtract:

$$\mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \widehat{Q}(x,a))^2] = \mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \widehat{Q}(x,a))^2] - \inf_{\bar{Q}\in\mathcal{Q}} \mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \bar{Q}(x,a))^2] \tag{50}$$

$$+ \inf_{\bar{Q}\in\mathcal{Q}} \mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \bar{Q}(x,a))^2] \tag{51}$$

$$\leq \mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \widehat{Q}(x,a))^2] - \inf_{\bar{Q}\in\mathcal{Q}} \mathbb{E}_{x,a\sim\mathcal{D},\beta}[(Q(x,a) - \bar{Q}(x,a))^2] \tag{52}$$

$$+ \varepsilon_{\mathcal{Q}}. \tag{53}$$

Now we want to show that the difference in squared error terms concentrates for large N. This is precisely what Lemma 16 from (Chen & Jiang, 2019) does using a one-sided Bernstein inequality. This gives us for any $\delta > 0$ an upper bound with probability $1 - \delta$ of

$$\frac{56\Delta_r^2 \log(|\mathcal{Q}|/\delta)}{3N} + \sqrt{\varepsilon_{\mathcal{Q}} \frac{32\Delta_r^2 \log(|\mathcal{Q}|/\delta)}{N}} \tag{54}$$

Plugging this in and simplifying the constants gives the result. □