A Theoretical Results

Theorem 4.1: Given a policy π_A and demonstrator π_E and environment horizon length H, the distribution shift:

$$D_{\text{KL}}(\rho_{\pi_A}, \rho_{\pi_E}) \leq \frac{1}{H} \sum_{t=0}^{H-1} (H-t) \mathbb{E}_{s \sim \rho^t_{\pi_A}} \left[D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s)) \right]$$

Proof. Using the log-sum inequality:

$$\begin{aligned}
D_{\text{KL}}(\rho^t_{\pi_A}, \rho^t_{\pi_E}) &= \int_{s'} \rho^t_{\pi_A}(s') \log \frac{\rho^t_{\pi_A}(s')}{\rho^t_{\pi_E}(s')} \\
&= \int_{s'} \left(\int_{s,a} \rho^{t-1}_{\pi_A}(s) \pi_A(a|s) \rho(s'|a,s) \right) \log \frac{\int_{s,a} \rho^{t-1}_{\pi_A}(s) \pi_A(a|s) \rho(s'|a,s)}{\int_{s,a} \rho^{t-1}_{\pi_E}(s) \pi_E(a|s) \rho(s'|a,s)} \\
&\leq \int_{s'} \int_{s,a} \rho^{t-1}_{\pi_A}(s) \pi_A(a|s) \rho(s'|a,s) \log \frac{\rho^{t-1}_{\pi_A}(s) \pi_A(a|s) \rho(s'|a,s)}{\rho^{t-1}_{\pi_E}(s) \pi_E(a|s) \rho(s'|a,s)} \\
&\leq \int_{s'} \int_{s,a} \rho^{t-1}_{\pi_A}(s) \pi_A(a|s) \rho(s'|a,s) (\log \frac{\rho^{t-1}_{\pi_A}(s)}{\rho^{t-1}_{\pi_E}(s)} + \log \frac{\pi_A(a|s)}{\pi_E(a|s)}) \\
&\leq \int_{s,a} \rho^{t-1}_{\pi_A}(s) \pi_A(a|s) (\log \frac{\rho^{t-1}_{\pi_A}(s)}{\rho^{t-1}_{\pi_E}(s)} + \log \frac{\pi_A(a|s)}{\pi_E(a|s)}) \\
&\leq \int_{s} \rho^{t-1}_{\pi_A}(s) \log \frac{\rho^{t-1}_{\pi_A}(s)}{\rho^{t-1}_{\pi_E}(s)} + \int_{s,a} \rho^{t-1}_{\pi_A}(s) \pi_A(a|s) \log \frac{\pi_A(a|s)}{\pi_E(a|s)} \\
&\leq D_{\text{KL}}(\rho^{t-1}_{\pi_A}, \rho^{t-1}_{\pi_E}) + \mathbb{E}_{s \sim \rho^t_{\pi_A}} \left[D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s)) \right] \\
&\leq D_{\text{KL}}(\rho^0(\cdot), \rho^0(\cdot)) + \sum_{j=0}^{t-1} \mathbb{E}_{s \sim \rho^j_{\pi_A}} \left[D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s)) \right] \quad \triangleright \text{ Recursive} \\
&\leq \sum_{j=0}^{t-1} \mathbb{E}_{s \sim \rho^j_{\pi_A}} \left[D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s)) \right] \\
D_{\text{KL}}(\rho_{\pi_A}, \rho_{\pi_E}) &= \int_s \left(\frac{1}{H} \sum_{t=1}^{H} \rho^t_{\pi_A}(s) \right) \log \frac{\sum_{t=1}^{H} \rho^t_{\pi_A}(s)}{\sum_{t=1}^{H} \rho^t_{\pi_E}(s)} \\
&\leq \int_s \sum_{t=1}^{H} \rho^t_{\pi_A}(s) \log \frac{\rho^t_{\pi_A}(s)}{\rho^t_{\pi_E}(s)} \\
&\leq \frac{1}{H} \sum_{t=1}^{H} D_{\text{KL}}(\rho^t_{\pi_A}, \rho^t_{\pi_E}) \\
&\leq \frac{1}{H} \sum_{t=1}^{H} \sum_{j=0}^{t-1} \mathbb{E}_{s \sim \rho^j_{\pi_A}} \left[D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s)) \right] \\
&\leq \frac{1}{H} \sum_{t=0}^{H-1} (H-t) \mathbb{E}_{s \sim \rho^t_{\pi_A}} \left[D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s)) \right]
\end{aligned}$$

□

Note that Ke et al. show a similar theorem, however they make the strong assumption of the f-divergence satisfying the triangle inequality, which is not true for the KL divergence we use in Theorem 4.1, and also yields a different bound. Furthermore, the implications of the divergence relationship on data quality (i.e. the data generating policy distribution) is not examined within this prior work. They are focusing more on the algorithmic perspective, as is common in prior work.

Lemma 4.2: Given learned policy π_A and expert π_E, define assume that the policy is learned such that when $s \in \text{supp}(\rho^t_{\pi_E})$, $D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s)) \leq \beta$. Then $\mathbb{E}_{s\sim\rho^t_{\pi_A}}[D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s))] \leq \mathbb{E}_{s\in\rho^t_{\pi_A}}[\beta\mathbb{1}(s \in \text{supp}(\rho^t_{\pi_E})) + \mathbb{1}(s \notin \text{supp}(\rho^t_{\pi_E}))D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s))]$

Proof. This follows by simple substitution:

$$\begin{aligned}\mathbb{E}_{s\sim\rho^t_{\pi_A}}[D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s))] &= \mathbb{E}_{s\in\rho^t_{\pi_A}}[\mathbb{1}(s \in \text{supp}(\rho^t_{\pi_E}))D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s)) \\ &\quad + \mathbb{1}(s \notin \text{supp}(\rho^t_{\pi_E}))D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s))] \\ &\leq \mathbb{E}_{s\in\rho^t_{\pi_A}}[\beta\mathbb{1}(s \in \text{supp}(\rho^t_{\pi_E})) \\ &\quad + \mathbb{1}(s \notin \text{supp}(\rho^t_{\pi_E}))D_{\text{KL}}(\pi_A(\cdot|s), \pi_E(\cdot|s))]\end{aligned}$$

□

Theorem 4.3: Given a policy π_A and demonstrator π_E, assume that for state s, if $\rho^{t-1}_{\pi_E}(s) > 0$, then $\pi_A(a|s) = \pi_E(a|s)$. Assume that transitions are normally distributed with fixed and diagonal variance, $\rho(s'|s,a) = \mathcal{N}(\mu(s,a), \sigma^2 I)$, then the next state coverage probability is $P_S(s; N, \epsilon) \geq 1 - \left(1 - (\frac{c\epsilon}{\sigma})^d \exp\left(-\alpha^2 d\right)\right)^N - \exp\left(-(\alpha-1)^2 d\right)$, where d is the dimensionality of the state, c is a constant, and $\alpha \geq 1$ is a parameter chosen to maximize the bound.

Proof. The proof follows by first discretizing the state space into length ϵ bins. Denote the probability mass of bin b as p_b. First we note that for two independent samples from the same distribution s' and $s'_{i,*}$, we can say that $P(|s' - s'_{i,*}|_\infty \geq \epsilon)$ is upper bounded by the probability that neither sample lands in the same bin. This is because of the fact that if $|s' - s'_{i,*}|_\infty \geq \epsilon$, then that implies the samples did not land in the same bin (if they did the infinity norm would be less than epsilon). Thus, we can also say that $P(\min_i |s' - s'_{i,*}|_\infty \geq \epsilon)$ is upper bounded by the probability that all of the samples $s'_{i,*}$ land in a different bin than s'. Thus:

$$\begin{aligned}P(\min_i |s' - s'_{i,*}|_\infty \geq \epsilon) &\leq P(\text{none lands in same bin}) \\ &\leq \sum_b p_b(1-p_b)^N\end{aligned}$$

Next define a ball of radius R around the mean $\mu(s,a)$ as $\text{Ball}(R)$. Using this ball we can partition the above inequality into two terms to yield an upper bound.

$$\begin{aligned}P(\min_i |s' - s'_{i,*}|_\infty \geq \epsilon) &\leq \sum_b p_b(1-p_b)^N \\ &\leq \sum_{b\in\text{Ball}(R)} p_b(1-p_b)^N + \sum_{b\notin\text{Ball}(R)} p_b(1-p_b)^N\end{aligned}$$

We can upper bound everything inside the ball using the minimum Gaussian mass for the given radius R, or $p_b \geq (\frac{c\epsilon}{\sigma})^d \exp\frac{-R^2}{2\sigma^2}$ where $c = \frac{1}{\sqrt{2\pi}}$. For the second term, if R is large enough, then we can assume p_b is sufficiently small such that $(1-p_b)^N \approx 1$.

$$\begin{aligned}P(\min_i |s' - s'_{i,*}|_\infty \geq \epsilon) &\leq \sum_{b\in\text{Ball}(R)} p_b\left(1 - \left(\frac{c\epsilon}{\sigma}\right)^d \exp\frac{-R^2}{2\sigma^2}\right)^N + \sum_{b\notin\text{Ball}(R)} p_b \\ &\leq \left(1 - \left(\frac{c\epsilon}{\sigma}\right)^d \exp\left(\frac{-R^2}{2\sigma^2}\right)\right)^N \left(\sum_{b\in\text{Ball}(R)} p_b\right) + \sum_{b\notin\text{Ball}(R)} p_b \\ &\leq \left(1 - \left(\frac{c\epsilon}{\sigma}\right)^d \exp\left(\frac{-R^2}{2\sigma^2}\right)\right)^N + \sum_{b\notin\text{Ball}(R)} p_b\end{aligned}$$

The second term is upper bounded by the probability that the d-dim Gaussian s is farther in euclidean distance than R from the mean, which can be written using the tail probability of the χ^2 distribution.

$$P(\min_i |s' - s'_{i,*}|_\infty \geq \epsilon) \leq \left(1 - \left(\frac{c\epsilon}{\sigma}\right)^d \exp\left(\frac{-R^2}{2\sigma^2}\right)\right)^N + P(||s - \mu(s,a)||_2^2 \geq R^2))$$
$$\leq \left(1 - \left(\frac{c\epsilon}{\sigma}\right)^d \exp\left(\frac{-R^2}{2\sigma^2}\right)\right)^N + \left(1 - \chi^2(\frac{R^2}{\sigma^2})\right)$$

We know that for a χ^2 random variable Y over d sub Gaussians, $P(\frac{Y}{d} \geq (1+\delta)^2) \leq \exp\left(\frac{-d\delta^2}{2}\right)$ for $\delta \geq 0$ [52]. Thus assuming $R \geq \sigma\sqrt{d}$, we can write:

$$P(\min_i |s' - s'_{i,*}|_\infty \geq \epsilon) \leq \left(1 - \left(\frac{c\epsilon}{\sigma}\right)^d \exp\frac{-R^2}{2\sigma}\right)^N + \exp\left(-\left(\frac{R}{\sigma} - \sqrt{d}\right)^2\right)$$

Now, for any $\alpha = \frac{R}{\sigma\sqrt{d}} \geq 1$, we can rewrite the above bound as:

$$P(\min_i |s' - s'_{i,*}|_\infty \geq \epsilon) \leq \left(1 - \left(\frac{c\epsilon}{\sigma}\right)^d \exp\left(-\alpha^2 d\right)\right)^N + \exp\left(-(\alpha-1)^2 d\right)$$

Starting from Definition 1:

$$\begin{aligned} P_S(s; N, \epsilon) &= P(\min_i ||s' - s'_{*,i}||_\infty \leq \epsilon) \\ &= 1 - P(\min_i |s' - s'_{i,*}|_\infty \geq \epsilon) \\ &\geq 1 - \left(1 - \left(\frac{c\epsilon}{\sigma}\right)^d \exp\left(-\alpha^2 d\right)\right)^N - \exp\left(-(\alpha-1)^2 d\right) \end{aligned}$$

□

A.1 Generalization under System Noise

Definition 2. *Given a policy π_A, a data generating policy π_E and a starting state s, define the probability of next state coverage $P_B(s, \mu; N) = 1 - \cap_i P(||s'_i - \mu||^2 \geq ||s'_{i,*} - \mu||^2)$, where $s'_{i,*} \sim \rho^t_{\pi_E}(\cdot|s)$ are next state samples from the expert starting at s, and $s'_i \sim \rho^t_{\pi_A}(\cdot|s)$ are next state samples from π_A starting at s.*

Intuitively $P_B(s; N)$ is the probability that given N chances, the hyper-sphere defined by the L2 distance from μ to a sampled data point contains a sample from the learned policy. Here the hyper-sphere represents the set of next states that the policy can generalize to. The N chances approximate the effect of having more data samples to leverage for generalization. For policies learned with neural networks, this aims to represent the "interpolation" capacity of these models among the training samples.

Theorem A.1. *Given a policy π_A and deterministic demonstrator π_E, assume that for state s, if $\rho^{t-1}_{\pi_E}(s) > 0$, then $\pi_A(a|s) = \mathcal{N}(\pi_E(s), \sigma^2_\pi I)$. Assume that transitions are normally distributed with fixed and diagonal variance, $\rho(s'|s,a) = \mathcal{N}(\mu(s,a), \sigma^2 I)$, where $\mu(s,a) = s + \alpha a$ are simplified linear dynamics for scalar $\alpha \in \mathbb{R}$. Then the next state coverage probability is $P_B(s, \mu(s,a); N) = 1 - \left(1 - F_d\left(\frac{1}{1+\alpha^2\left(\frac{\sigma_p}{\sigma}\right)^2}\right)\right)^N$, where $F_d(x)$ is the CDF of the f-distribution of dimension d.*

Proof. First we note that both $\Delta_i = s'_i - \mu(s,a)$ and $\Delta_{i,*} = s'_{i,*} - \mu(s,a)$ are zero mean, and the events for $\Delta_i \geq \Delta_{i,*}$ for all i are independent. Thus:

$$
\begin{aligned}
P_B(s, \mu; N) &= 1 - \cap_i P(||\Delta_i||^2 \geq ||\Delta_{i,*}||^2) \\
&= 1 - \left(1 - P(||\Delta_i||^2 \leq ||\Delta_{i,*}||^2)\right)^N \\
&= 1 - \left(1 - P\left(\frac{||\Delta_i||^2}{||\Delta_{i,*}||^2} \leq 1\right)\right)^N
\end{aligned}
$$

Note that $||\Delta_i||^2$ is just a χ^2_d random variable with d degrees of freedom, and likewise for $||\Delta_{i,*}||^2$. The former has variance $\sigma^2 + (\alpha\sigma_p)^2$ (transition plus learned policy noise), while the latter has variance σ^2 (just transition noise). The f-distribution is defined for two χ^2_d variables X and Y of dimension d as the distribution of $Z = \frac{X/d}{Y/d}$. Thus with a change of variables to X and Y, and denoting F_d as the CDF of the f-distribution of dimension d:

$$
\begin{aligned}
P\left(\frac{||\Delta_i||^2}{||\Delta_{i,*}||^2} \leq 1\right) &= P\left(\frac{(\sigma^2 + (\alpha\sigma_p)^2)X}{\sigma^2 Y} \leq 1\right) \\
&= P\left(\frac{X}{Y} \leq \frac{\sigma^2}{\sigma^2 + (\alpha\sigma_p)^2}\right) \\
&= F_d\left(\frac{\sigma^2}{\sigma^2 + (\alpha\sigma_p)^2}\right) \\
&= F_d\left(\frac{1}{1 + \alpha^2\left(\frac{\sigma_p}{\sigma}\right)^2}\right)
\end{aligned}
$$

Plugging the above expression into the expression for P_B:

$$
P_B(s, \mu; N) = 1 - \left(1 - F_d\left(\frac{1}{1 + \alpha^2\left(\frac{\sigma_p}{\sigma}\right)^2}\right)\right)^N
$$

□

In Theorem A.1, while the probability of next state coverage is not immediately interpretable, we can still intuitively recognize that high σ_s can improve the coverage likelihood of the learned policy even under notable σ_p, and as N gets bigger even less σ_s is needed, despite the fact that system noise is present even under the learned policy. Intuitively, we once again see that increasing N has a significant effect on the coverage likelihood. In terms of noise, what matters is the ratio of policy to system noise, where increasing this ratio leads to sharp drops in performance at some cutoff based on N. We visualize this coverage probability in Fig. 4 under increasing ratios of policy to system noise for different values of N.

B Metrics of Data Quality

Having formalized action divergence and transition diversity in Sec. 4 as two fundamental considerations in a dataset, how can we *measure* these properties in a given dataset?

Action Variance: To measure action consistency, the empirical form of the objective in Eqn. 7 is intractable without access to the underlying expert action distribution π_E. Instead we propose using the empirical variance of the action distribution in the data to approximate the "spread" of the data. In continuous state spaces, we can estimate variance using a coverage distance ϵ to cluster nearby states, and then measuring the per dimension variance across the corresponding actions within said cluster. Defining a cluster to be $C(s, \mathcal{D}) = \{\tilde{s}, \tilde{a}, \tilde{s'} \in \mathcal{D} : ||s - \tilde{s}|| \leq \epsilon\}$, we can compute the variance as:

$$
\text{ActionVariance}(\mathcal{D}) = \frac{1}{|D|} \sum_{s,a \in \mathcal{D}} (a - \sum_{\tilde{s},\tilde{a},\tilde{s'} \in C(s,\mathcal{D})} \tilde{a})^2 \tag{9}
$$

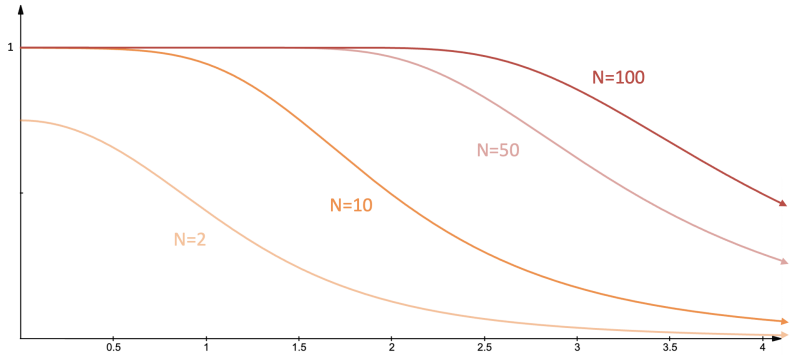

Figure 4: $P_B(s, \mu; N)$ (y-axis) from Theorem A.1 plotted for four-dimensional state under $N \in [2, 10, 50, 100]$, but varying the ratio of policy to system noise (x-axis is $\frac{\sigma_p}{\sigma}$). We see that under this more loose coverage model, with lots of samples, adding system noise can make coverage likely even under double or triple the noise in the learned policy.

The choice in ϵ corresponds to the *generalization* of the learning model to nearby states, similar to the notion of coverage in Definition 1. We use this metric of action consistency in Sec. 5 to study human generated datasets of various quality.

State Similarity: To measure the consistency of states, we approximate the number of "nearby" states using the same clustering process as in the Action Variance metric, and measure the expected cluster size as a fraction of the overall data.

$$\text{StateSimilarity}(\mathcal{D}) = \frac{1}{|D|} \sum_{s,a \in \mathcal{D}} |C(s, \mathcal{D})| \tag{10}$$

While these approximate forms do not encapsulate the full spectrum of possible metrics, we believe these metrics help advance our empirical understanding of data quality for imitation learning. In section 5.2 in the main text, we analyze these metrics of data quality in several environments across different dataset sources.

C Results

The performance results under system noise, policy noise, and both noises are shown with a broader sweep for both *PMObstacle* and *Square* in the tables below.

	$\sigma_s = 0.01$	$\sigma_s = 0.02$	$\sigma_s = 0.03$	$\sigma_s = 0.04$	$\sigma_s = 0.01$	$\sigma_s = 0.02$	$\sigma_s = 0.03$	$\sigma_s = 0.04$
SCRIPTED	100	100	99	96				
$\sigma_s = 0.01$	97.7(1.5)	95.7(0.7)	96.7(1.1)	93.3(0.7)	90.3(7.1)	90.0(8.2)	94.0(2.9)	87.7(3.7)
$\sigma_s = 0.02$	98.7(0.5)	98.0(0.5)	97.7(1.0)	92.7(1.2)	99.7(0.3)	98.0(0.9)	94.3(1.4)	92.3(2.0)
$\sigma_s = 0.03$	98.3(0.7)	98.0(0.8)	99.0(0.5)	95.0(0.9)	99.7(0.3)	98.7(0.5)	97.7(0.5)	95.7(1.1)
$\sigma_s = 0.04$	100.0(0.0)	100.0(0.0)	99.3(0.3)	96.7(0.7)	100.0(0.0)	99.0(0.5)	98.7(0.5)	96.7(1.4)

Table 2: **System Noise**: Success rates (and standard error) for BC in PMObstacle, for 1000 episodes (left) and 10 episodes (right) of data, under system noise. Rows correspond to injecting gaussian system noise (σ_s) into the *dataset* of increasing variance, and columns correspond to injecting noise during *evaluation*. The diagonal in both sub-tables represents evaluating in distribution. **Left**: For large datasets, higher system noise during evaluation tends to decrease the performance of each model (rows left to right), but more system noise during training generally produces the best models (columns top to bottom). **Right**: For small datasets, we observe a similar but exaggerated effect as the left table.

	$\sigma_s = 0.01$	$\sigma_s = 0.02$	$\sigma_s = 0.03$	$\sigma_s = 0.04$	$\sigma_s = 0.01$	$\sigma_s = 0.02$	$\sigma_s = 0.03$	$\sigma_s = 0.04$
SCRIPTED	100	100	99	96				
$\sigma_p = 0.01$	94.0(1.7)	94.0(2.4)	94.7(1.8)	91.3(1.4)	78.0(8.6)	78.7(6.7)	81.3(5.0)	81.3(5.8)
$\sigma_p = 0.02$	87.7(2.0)	92.3(2.0)	90.7(2.0)	91.3(2.2)	88.0(9.4)	78.7(4.4)	80.7(3.1)	80.3(3.2)
$\sigma_p = 0.03$	97.0(0.9)	99.0(0.5)	97.0(0.0)	95.0(0.8)	88.7(3.2)	82.7(5.4)	88.7(5.2)	85.7(4.4)
$\sigma_p = 0.04$	86.7(4.3)	91.0(2.4)	93.3(1.4)	92.7(1.5)	88.3(6.5)	88.0(4.5)	86.0(5.9)	82.3(2.0)

Table 3: **Policy Noise**: Success rates (and standard error) for BC in PMObstacle, for 1000 episodes (left) and 10 episodes (right) of data, under learned policy noise. Rows correspond to injecting gaussian policy noise (σ_p) into the *expert* of increasing variance, and columns correspond to injecting system noise during *evaluation*. **Left**: For large datasets, unlike system noise in Table 2, more policy noise during training often produces the worst models (columns top to bottom). **Right**: For small datasets, adding policy noise produces large variance in performance across runs. Importantly, the datasets in each row have the same observed state diversity as the corresponding row in Table 2, but performance is almost universally lower in both sub-tables here, supporting the idea that state diversity is a coarse metric for success.

	$\sigma_s = 0.01$	$\sigma_s = 0.02$	$\sigma_s = 0.03$	$\sigma_s = 0.04$	$\sigma_s = 0.01$	$\sigma_s = 0.02$	$\sigma_s = 0.03$	$\sigma_s = 0.04$
SCRIPTED	100	100	99	96				
$\sigma_p = 0.01$	96.3(2.2)	99.3(0.5)	97.7(0.3)	93.3(1.0)	99.7(0.3)	98.0(1.2)	96.7(0.5)	96.3(1.4)
$\sigma_p = 0.02$	98.0(0.5)	98.3(0.5)	97.7(0.7)	94.7(1.0)	99.3(0.5)	98.7(1.1)	96.0(2.2)	94.7(1.7)
$\sigma_p = 0.03$	98.0(0.8)	96.7(1.0)	98.3(1.0)	96.3(0.7)	95.0(2.1)	97.7(0.5)	97.7(0.5)	93.3(2.2)
$\sigma_p = 0.04$	98.7(0.5)	99.0(0.5)	97.3(0.3)	95.0(0.8)	100.0(0.0)	99.7(0.3)	99.0(0.8)	93.7(3.1)

Table 4: **System Noise + Policy Noise**: Success rates (and standard error) for BC in PMObstacle, for 1000 episodes (left) and 10 episodes (right) of data, under learned policy noise for a fixed amount of system noise ($\sigma_p = 0.03$). Here we see how system noise improves the robustness of the model to added policy noise.

	$\sigma_s = 0.05$	$\sigma_s = 0.1$	$\sigma_s = 0.2$	$\sigma_s = 0.3$	$\sigma_s = 0.4$
$\sigma_s = 0.05$	55.7(5.3)	50.0(5.4)	27.0(3.9)	12.0(2.4)	7.3(2.2)
$\sigma_s = 0.1$	69.7(5.5)	69.0(3.7)	57.3(6.0)	50.3(6.0)	22.7(3.5)
$\sigma_s = 0.2$	67.7(9.6)	68.7(12.6)	82.0(3.4)	74.3(2.7)	50.3(1.7)
$\sigma_s = 0.3$	47.0(4.9)	53.3(5.9)	54.3(3.0)	50.7(4.3)	38.7(7.3)
$\sigma_s = 0.4$	31.3(5.0)	37.7(8.2)	48.3(9.7)	49.0(8.7)	44.0(5.3)

Table 5: **System Noise, 200ep**: Success rates for BC in Square, for 200 episodes of data, under system noise. Rows correspond to injecting gaussian system noise (σ_s) into the *dataset* of increasing variance, and columns correspond to injecting noise during *evaluation*. The diagonal in both sub-tables represents evaluating in distribution. In both sub-tables we see how policies with low data coverage (low system noise) generalize the worst to increasing noise at test time. More system noise during training generally produces the best models (columns top to bottom).

	$\sigma_s = 0.05$	$\sigma_s = 0.1$	$\sigma_s = 0.2$	$\sigma_s = 0.3$	$\sigma_s = 0.4$
$\sigma_s = 0.05$	40.0(1.2)	33.7(3.1)	16.7(1.4)	4.3(1.0)	2.0(0.5)
$\sigma_s = 0.1$	42.3(4.0)	39.7(3.8)	31.3(3.7)	19.7(3.1)	10.0(1.6)
$\sigma_s = 0.2$	70.0(7.0)	73.7(5.4)	69.7(0.7)	55.3(1.2)	27.0(2.2)
$\sigma_s = 0.3$	57.3(3.1)	58.7(3.1)	64.7(1.4)	60.7(0.3)	44.7(2.2)
$\sigma_s = 0.4$	30.0(7.0)	33.7(6.4)	39.7(5.7)	39.7(6.5)	36.7(6.9)

Table 6: **System Noise, 50ep**: Success rates for BC in Square, for 50 episodes of data, under system noise. Rows correspond to injecting gaussian system noise (σ_s) into the *dataset* of increasing variance, and columns correspond to injecting noise during *evaluation*. The diagonal in both sub-tables represents evaluating in distribution. In both sub-tables we see how policies with low data coverage (low system noise) generalize the worst to increasing noise at test time. More system noise during training generally produces the best models (columns top to bottom).

	$\sigma_s = 0.05$	$\sigma_s = 0.1$	$\sigma_s = 0.2$	$\sigma_s = 0.3$	$\sigma_s = 0.4$
$\sigma_p = 0.005$	69.0(3.7)	59.0(3.7)	34.0(1.7)	20.7(2.4)	7.0(1.6)
$\sigma_p = 0.01$	80.7(3.7)	78.0(4.2)	57.7(2.4)	38.0(1.7)	23.0(2.2)
$\sigma_p = 0.02$	62.3(7.8)	71.7(6.1)	73.0(3.9)	65.3(2.8)	43.3(3.6)

Table 7: **Policy Noise, 200ep**: Success rates for BC in Square, for 200 episodes of data, under learned policy noise. Rows correspond to injecting gaussian policy noise (σ_p) into the *dataset* of increasing variance, and columns correspond to injecting noise during *evaluation*. In the high data regime, we see that more policy noise tends to improve performance (columns top to bottom), since the noise is unbiased so with enough samples from the scripted policy, the model will recover an unbiased policy.

	$\sigma_s = 0.05$	$\sigma_s = 0.1$	$\sigma_s = 0.2$	$\sigma_s = 0.3$	$\sigma_s = 0.4$
$\sigma_p = 0.005$	32.7(3.8)	30.3(3.2)	18.0(3.6)	7.0(0.8)	5.7(1.2)
$\sigma_p = 0.01$	61.3(4.3)	59.0(6.7)	48.3(3.7)	29.7(1.7)	19.3(1.4)
$\sigma_p = 0.02$	57.7(3.2)	58.3(3.1)	49.3(0.5)	41.3(0.5)	28.3(2.4)

Table 8: **Policy Noise, 50ep**: Success rates for BC in Square, for 50 episodes of data, under learned policy noise. Rows correspond to injecting gaussian policy noise (σ_p) into the *dataset* of increasing variance, and columns correspond to injecting noise during *evaluation*. As the amount of data is reduced, there is a significant drop in performance for added policy noise in the dataset, along with higher performance variation compared to 200eps, since the policy can no longer recover an unbiased policy.