

A Proofs

Here we provide the derivations for the generalization bound in Section 3. The proof requires a non-trivial combination of previous results in provable imitation learning, mostly from [13], and the generalization guarantees of information bottleneck [20]. These results are reported in Lemma A.3 and A.4. Before going through the proofs, we state the main theorem again for convenience.

Theorem A.1 (Theorem 3.1). *For a confidence $\delta \in (0, 1)$, it holds with probability at least $1 - 2\delta$*

$$\begin{aligned} & \mathbb{E}_{T \sim P_0} [J(\pi_T^*) - J(\hat{\pi})] \\ & \lesssim RH \left(\mathcal{E}_{gen}(\pi^E) + \mathcal{E}_{opt}(\hat{\pi}) + C(E, \pi^E, \hat{\pi}) \sqrt{\frac{I_{T;Z} |\mathcal{A}| \log(|\mathcal{A}|/\delta)}{m}} + \frac{8 \log(|\Pi|m/\delta)}{n} \right) \end{aligned}$$

where

- R is an upper bound to the cumulative reward of any policy in any task $\theta \in \Theta$ (Section 2);
- $\mathcal{E}_{gen}(\pi^E)$ is the generalization error of the expert's policy (Asm. 2);
- $\mathcal{E}_{opt}(\hat{\pi})$ is a bound to the optimization error of solving (1) (Asm. 4);
- $C(E, \pi^E, \hat{\pi})$ is a constant that depends on the training data E , the expert's policy π^E , the cloned policy $\hat{\pi}$, and other absolute constants as detailed in Lemma A.4.

Proof. We derive the result as follows

$$\begin{aligned} & \mathbb{E}_{T \sim P_0} [J(\pi_T^*) - J(\hat{\pi})] \\ & \leq RH \mathbb{E}_{T \sim P_0} \mathbb{E}_{\tau \sim \mathbb{P}_T^{\pi^*}} [\mathbf{1}(\pi_T^*(a^h|x^h) \neq \hat{\pi}(a^h|x^h))] \end{aligned} \quad (4)$$

$$\leq RH \mathcal{E}_{gen}(\hat{\pi}) \quad (5)$$

$$\leq RH \left(\mathcal{E}_{gen}(\pi^E) + \mathbb{E}_{T \sim P_0} \mathbb{E}_{X A \sim \mathbb{P}_T^{\pi^E}} [\mathbf{1}(\hat{\pi}(X) \neq A)] \right) \quad (6)$$

$$\lesssim RH \left(\mathcal{E}_{gen}(\pi^E) + \mathcal{E}_{opt}(\hat{\pi}) + \sqrt{\frac{I_{T;Z} |\mathcal{A}| \log(|\mathcal{A}|/\delta)}{m}} + \frac{8 \log(|\Pi|m/\delta)}{n} \right) \quad (7)$$

where (4) and (5) are straightforward from the definitions of the performance $J(\pi)$ and the generalization error $\mathcal{E}_{gen}(\pi)$ (see Section 2 and (3) respectively), (6) follows from Assumption 2, and (7) holds with probability at least $1 - 2\delta$ through Lemma A.2. \square

We provide below the lemmas we need to prove the result above.

Lemma A.2. *For a confidence $\delta \in (0, 1)$, it holds with probability at least $1 - 2\delta$*

$$\mathbb{E}_{T \sim P_0} \mathbb{E}_{X A \sim \mathbb{P}_T^{\pi^E}} [\mathbf{1}(\hat{\pi}(X) \neq A)] \lesssim \sqrt{\frac{I_{T;Z} |\mathcal{A}| \log(|\mathcal{A}|/\delta)}{m}} + \frac{8 \log(|\Pi|m/\delta)}{n} + \mathcal{E}_{opt}(\hat{\pi})$$

where $I_{T;Z}$ is the mutual information between the task T and the internal representation of the demonstrator Z .

Proof. We derive the result as follows

$$\begin{aligned} & \mathbb{E}_{T \sim P_0} \mathbb{E}_{X A \sim \mathbb{P}_T^{\pi^E}} [\mathbf{1}(\hat{\pi}(X) \neq A)] \\ & \leq \mathbb{E}_{T \sim P_0} \mathbb{E}_{X A \sim \mathbb{P}_T^{\pi^E}} [\mathbf{1}(\hat{\pi}(X) \neq A)] - \frac{1}{mnH} \sum_{i=1}^m \sum_{j=1}^n \sum_{h=0}^{H-1} \mathbf{1}(\hat{\pi}(\tau_{ij}^h) \neq a_{ij}^h) + \mathcal{E}_{opt}(\hat{\pi}) \quad (8) \\ & \leq \mathbb{E}_{T \sim P_0} \mathbb{E}_{X A \sim \mathbb{P}_T^{\pi^E}} [\mathbf{1}(\hat{\pi}(X) \neq A)] - \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{X A \sim \mathbb{P}_{\theta_i}^{\pi^E}} [\mathbf{1}(\hat{\pi}(X) \neq A)] + \frac{8 \log(|\Pi|m/\delta)}{n} + \mathcal{E}_{opt}(\hat{\pi}) \quad (9) \end{aligned}$$

$$\lesssim \sqrt{\frac{I_{T;Z|\mathcal{A}} \log(|\mathcal{A}|/\delta)}{m}} + \frac{8 \log(|\Pi|m/\delta)}{n} + \mathcal{E}_{opt}(\hat{\pi}) \quad (10)$$

where (8) is a trivial consequence of Assumption 4 on the optimization error for solving (1), (9) holds with probability at least $1 - \delta$ through Lemma A.3 and a union bound on the m training tasks, and (10) holds with probability $1 - 2\delta$ from Lemma A.4 by omitting constant and lower order terms and applying a union bound. \square

Lemma A.3 (Sample complexity of behavioral cloning [13]). *For a confidence $\delta \in (0, 1)$, an MDP θ , and a deterministic expert's policy π^E , it holds with probability at least $1 - \delta$*

$$\mathbb{E}_{XA \sim \mathbb{P}_{\theta}^{\pi^E}} [\mathbf{1}(\hat{\pi}(X) \neq A)] \leq \frac{8 \log(|\Pi|/\delta)}{n}.$$

Proof. This result can be obtained through a combination of results in [13]. First, for a policy $\hat{\pi}$ obtained by minimizing the negative log likelihood of the data, as in (1), from Proposition 2.1 [13] we have with probability at least $1 - \delta$ that

$$D_H^2(\mathbb{P}_{\theta}^{\hat{\pi}}, \mathbb{P}_{\theta}^{\pi^E}) \leq \frac{2 \log(|\Pi|/\delta)}{n}$$

where $D_H^2(\mathbb{P}, \mathbb{Q}) = \int (\sqrt{d\mathbb{P}} - \sqrt{d\mathbb{Q}})^2$ is the squared Hellinger distance between the probability measures \mathbb{P} and \mathbb{Q} . Then, through Lemma F.3 [13] we have

$$\mathbb{E}_{XA \sim \mathbb{P}_{\theta}^{\pi^E}} [\mathbf{1}(\hat{\pi}(X) \neq A)] \leq 4D_H^2(\mathbb{P}_{\theta}^{\hat{\pi}}, \mathbb{P}_{\theta}^{\pi^E})$$

which concludes the proof. \square

Lemma A.4 (Information bottleneck generalization gap [20]). *For a dataset $E = \{\theta_i \sim P_0\}_{i=1}^m$ of m tasks and a single-point convex loss $\ell(\hat{\pi}(X), A)$, let us define the generalization gap across the prior P_0 as*

$$\Gamma(E) := \mathbb{E}_{T \sim P_0} \mathbb{E}_{XA \sim \mathbb{P}_T^{\pi^E}} [\ell(\hat{\pi}(X), A)] - \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{XA \sim \mathbb{P}_{\theta_i}^{\pi^E}} [\ell(\hat{\pi}(X), A)].$$

For a confidence $\delta \in (0, 1)$, $\Gamma(E)$ is upper bounded with probability at least $1 - \delta$ by

$$\beta \sqrt{\frac{I_{T;Z|A} \log 2 + \alpha_{\gamma} \log 2 + H(Z|T, A) + \log(2|\mathcal{A}|/\delta)}{m}} + \frac{f(\hat{\pi}) \sqrt{2\gamma|\mathcal{A}| \log(2|\mathcal{A}|/\delta)}}{m^{3/4}} + \frac{\gamma g(\hat{\pi})}{m^{1/2}}$$

where $\alpha_{\gamma}, \beta, \gamma$ are constant values, $f(\hat{\pi}) = \max_{i \in [m]} \mathbb{E}_{XA \sim \mathbb{P}_{\theta_i}^{\pi^E}} [\ell(\hat{\pi}(X), A)]$ is the maximum training loss, $g(\hat{\pi}) = \sup_{XA} \ell(\hat{\pi}(X), A)$ is the maximum generalization loss, and $H(Z|T, A)$ is the entropy of the demonstrator internal representation given TA .

Proof. This result is based on Theorem 1 in [20], in which the notation adapted to our setting of interest. All of the derivations can be found in [20]. A more coarse version of the bound is given as

$$\Delta(E) \leq \left(\beta \sqrt{\alpha_{\gamma} \log 2 + H(Z|T, A) + f(\hat{\pi}) \sqrt{2\gamma} + \gamma g(\hat{\pi})} \right) \sqrt{\frac{I_{T;Z|A} \log(2|\mathcal{A}|/\delta)}{m}}$$

where the first factor can be incorporated into a constant $C(E, \pi^E, \hat{\pi})$. We note that the term $H(Z|T, A) = 0$ whenever the demonstrator internal representation is deterministic, which is a fair assumption in our setting. Further, the maximum training error $f(\hat{\pi})$ is close to zero and upper bounded by the optimization error $\mathcal{E}_{opt}(\hat{\pi})$. The value of $g(\hat{\pi})$ is upper bounded by 1 for the indicator loss $\ell(\hat{\pi}(X), A) = \mathbf{1}(\hat{\pi}(X) \neq A)$. Finally, we note that $I_{T;Z} \geq I_{T;Z|A}$. With these considerations, by omitting all of the constants, we have

$$\Delta(E) \lesssim \sqrt{\frac{I_{T;Z|A} \log(|\mathcal{A}|/\delta)}{m}}$$

as it is reported elsewhere in the paper. \square

B Peg insertion extended results

This section describes in detail the setting, hyperparameters, and constants used for the peg insertion task, as well as provides extended evaluation results.

B.1 Experimental setup

Figure 7 shows a close-up view of all peg shapes (10 shapes in total) and their corresponding boards, where the training shapes are in the top row and the test shapes are in the bottom row. Each peg insertion attempt starts from 10cm above the hole (Z axis) and a random reset position within a 0.5cm box in the XY plane, centered above the hole position. In this experiment, initial rotations about the X and Y axes are fixed (0 degrees), while the Z angle starts from a random rotation ranging from -60 to 60 degrees. Two Realsense cameras are mounted on the robot’s wrist. For the blindfolded expert experiment, we mask out the hole to hide its orientation from the experts, such that they cannot infer the orientation of the peg and must explore the domain in order to insert the peg.

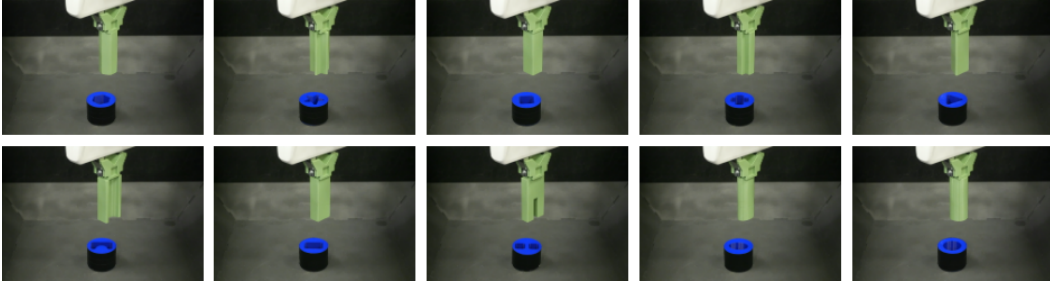


Figure 7: Close-up view of all peg insertion tasks. Top row: Training pegs. Bottom row: Test pegs.

B.2 Hyperparameters and constants

As described in section 4.2, the same architecture is used for learning both π_{BC} and π_{BF-BC} . Specifically, we use ResNet-10 [17] encoder pretrained on the ImageNet dataset [10], and a GRU of 1024, which we found to produce the best performance for both policies π_{BC} and π_{BF-BC} independently. Throughout our experiments, we train our networks using the Adam optimizer. The hyperparameters of the networks are the learning rate, learning rate decay, and the batch size. To ensure that the best performance of each approach is achieved, we perform a separate hyperparameter search for each policy, trained on each subset of shapes. The best hyperparameters are listed in Table 3. The network outputs a 6-dimensional vector for the mean and a 6-dimensional vector for the diagonal covariance matrix of a Gaussian policy (for 6-DoF action space). We train our networks using the log-likelihood loss. In all our evaluations, the action is chosen as the maximum likelihood of the distribution.

Table 3: List of hyperparameters used in the peg insertion experiment. The learning rate (lr) schedule indicates the iteration number for multiplying the lr by 0.5.

Hyperparameter	π_{BC}	π_{BF-BC}
batch size	1024	1024
hidden size	1024	1024
initial lr	0.0003	0.0003
lr schedule	$\text{lr} \times 0.5$ at $\{10, 100, 150, 200\}K$	$\text{lr} \times 0.5$ at $\{50, 100, 150, 200\}K$

B.3 Measuring exploratory behavior

We compute two additional measures for the exploratory behavior of the different experts: the map coverage score (Table 4) and the entropy of state visitation (Table 5).

Map coverage score is the ratio $C = N_v / N_{total}$ given by the number of visited states N_v divided by all accessible states N_{total} , averaged over all episodes. For the peg insertion experiment, we consider

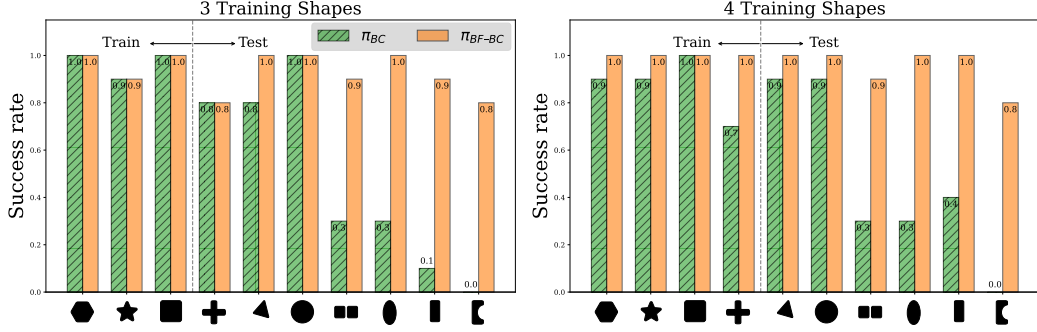


Figure 8: Success rate of robotic peg insertion for 10 peg shapes (horizontal axis). We train on a subset of shapes $k = \{3, 4\}$, with the remaining shapes withheld as test-set. Both for $k = 3$ (left) and $k = 4$ (right), cloning blindfolded experts generalizes better than cloning standard experts.

the rotation of the robotic arm around the Z-axis as the crucial component of the state space for obtaining the correct articulation for insertion. We compute the ratio of the rotation performed (in radians) divided by 2π in each trajectory, averaged over all trajectories.

The entropy of state visitation is defined by $H = -\sum_s p(s) \log p(s)$. We calculate $p(s)$ using a histogram (with 20 bins) of rotation angles along the trajectory, averaged over all trajectories.

The results confirm that blindfolded experts explore a larger portion of the state space to compensate for the redacted information in the observations.

Table 4: Map coverage score of the trajectories demonstrated by fully-informed experts (Experts) and blindfolded experts (BF-Experts).

Mode	hexagon	star	square	plus	triangle	Average
Experts	0.078	0.113	0.150	0.153	0.191	0.137
BF-Experts	0.114	0.259	0.267	0.270	0.327	0.247

Table 5: Entropy of the state visitation of the trajectories demonstrated by fully-informed experts (Experts) and blindfolded experts (BF-Experts).

Mode	hexagon	star	square	plus	triangle	Average
Experts	2.876	3.121	3.266	3.228	3.418	3.182
BF-Experts	3.100	3.416	3.546	3.577	3.631	3.454

B.4 Results for different combinations of training shapes

Figure 8 shows the success rate for $k = 3, 4$ training shapes (and the rest serve as a test set, out of a total of 10 peg shapes). The results on the varying amounts of training shapes, further support that cloning blindfolded experts generalizes better than the standard BC approach.

C Progen maze and heist extended results

This section describes in detail the hyperparameters and constants used for the Progen maze task.

C.1 Hyperparameters and constants

For a fair comparison, we conduct a separate hyperparameter search for both π_{BC} and π_{BF-BC} . We perform a hyperparameter search for the batch size $b \in \{128, 256, 512, 1024\}$, for the learning rate $\text{lr} \in \{1e^{-3}, 1e^{-4}, 1e^{-5}, 5e^{-3}, 5e^{-4}, 5e^{-5}\}$ and for hidden size $h \in \{128, 256, 512, 1024\}$. We also evaluate the performance with and without using a learning decay schedule. Our networks are trained using the Adam optimizer. The best hyperparameters are chosen based on the lowest training loss and the highest training success rate. We evaluate performance over an average of 10 different random

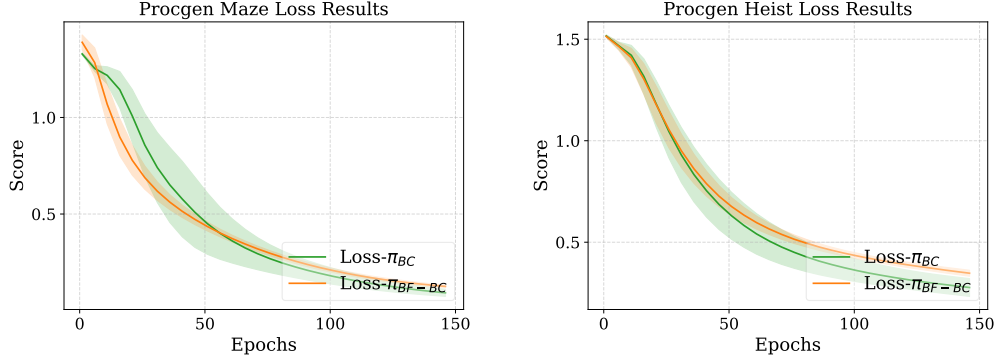


Figure 9: Loss as a function of training epochs for the Procgen maze (left) and heist (right). Mean (color line) and std (shaded region) are computed across 10 seeds.

training seeds. Figure 9 and Table 6 show the loss function and the chosen hyperparameters. Note that in most cases, the best hyperparameters for π_{BC} and π_{BF-BC} turned out to be fairly similar.

Table 6: List of hyperparameters used in the Procgen Maze experiment.

Hyperparameter	π_{BC}	π_{BF-BC}
batch size	256	256
hidden size	1024	1024
learning rate	10^{-5}	10^{-5}
learning rate schedule?	No	No

Table 7: List of hyperparameters used in the Procgen Heist experiment.

Hyperparameter	π_{BC}	π_{BF-BC}
batch size	128	128
hidden size	512	1024
learning rate	10^{-5}	10^{-5}
learning rate schedule?	No	No

C.2 Number of steps vs. number of trajectories

As described in Table 1, the blindfolded expert takes more steps on average to complete each trajectory. When comparing the different approaches, we match the number of trajectories, which leads to a greater total number of environment steps for the blindfolded expert. Table 8 shows the total number of steps available for training the different BC policies, alongside their performance. We also compare our results to a standard BC approach with twice the number of trajectories (from the same 100 seeds) to match the number of environment steps produced by the blindfolded expert. In addition, we compare our results to the results reported by [27], who train a BC policy on the Procgen maze with a dataset of $1M$ environment steps taken from a trained PPO expert, on 200 training seeds⁸.

We can see in Table 8 that π_{BF-BC} achieves better performance than all other contending policies. When compared to the results reported in [27], we can see that our results are better despite significantly less training data (an order of magnitude fewer trajectories) and half the number of training seeds.

⁸For $1M$ expert dataset, we report the results from [27] who evaluated over 5 seeds.

Table 8: Performance comparison on the Procgen maze experiment. Our results are reported at epoch 40 (early stopping) when training performance plateaus. The mean and std are computed over 10 seeds. Top performer in bold.

Parameter	1M Expert Dataset in [27]	π_{BC}	π_{BC-ext}	π_{BF-BC}
# of trajectories	15385	2000	4608	2000
# of total env steps	1000000	57166	115290	103238
# of seeds	200	100	100	100
Test performance	4.46 ± 0.16	3.35 ± 0.29	3.65 ± 0.3	6.37 ± 0.47

D Data collection

Peg insertion. For both π_{BC} and π_{BF-BC} , we use 400 trajectories for each of the training shapes. We, the authors, collected the data by operating the robot manually using a Spacemouse control. Recall that the blindfolded expert observes a masked-out view of the board (through the robot wrist cameras) such that the orientation of the peg is not directly visible and must be inferred through exploration. However, recorded observations in favor of cloning the blindfolded policy π_{BF-BC} are unmasked, i.e., only the human expert is blindfolded. In addition, we rescale the images from 480×480 to 128×128 for both π_{BC} and π_{BF-BC} to facilitate computations.

Procgen maze and heist.

To train our experts (BC) and blindfolded experts (BF-BC) policies, we collected 4000 human demonstrations on 200 levels. We conducted crowd-sourced data collection for the maze and heist videogames by recruiting 20 volunteers who played the games. Each expert played all 200 levels of maze and all 200 levels of heist twice, once with the mask and once without the mask. Their game trajectories were recorded to serve toward the imitation-learning of the experts’ policy π_{BC} and blindfolded experts’ policy π_{BF-BC} . The participants moved the mouse using the keyboard’s arrow keys and relied on the Procgen “interactive” GUI for maze and heist observations in full resolution (512×512). For the blindfolded experts’ data, their observations are modified to reveal only the agent’s immediate surroundings (a diameter of $\frac{1}{8}$ of the width for maze and $\frac{1}{6}$ for heist) with the rest of the observation masked out. Note that the state observations that are provided to the cloning networks are a lower resolution of 64×64 of the unmasked observations.

All participants were compensated with vouchers for their efforts. The experiment’s GUI environment, alongside the training code and the recorded data, are available at:
<https://github.com/EvZissel/blindfolded-experts/>