

491 **A Overview**

492 The Appendix contains the following content.

- 493 • **Policy Learning Details** (Appendix [B](#)): details on hyperparameters used
- 494 • **Ablation: SPIRE without TAMP** (Appendix [C](#)): ablation study on the effect of removing
495 TAMP-gating and directly running BC and RL fine-tuning
- 496 • **Comparison to Additional Methods** (Appendix [D](#)): comparison to other RL methods that
497 leverage demonstrations
- 498 • **Tasks** (Appendix [E](#)): details on tasks used to evaluate SPIRE
- 499 • **Variance Across Seeds** (Appendix [F](#)): discussion on the variance of results across different
500 seeds and how results are presented

501 B Policy Learning Details

Table 1: DrQ-v2 hyperparameters.

Network structure	CNN
Learning rate	1e-4
Discount	0.99
Batch size	256
n -step returns	3
Action repeat	1
Seed frames	4000
Feature dim	50
Hidden dim	1024
Optimizer	Adam

502 **Hyperparameters.** The base RL algorithm for all our experiments is DrQ-v2 [62]. The specific
 503 hyperparameters are in Table 1.

504 **Observation.** For most tasks, we use one 84×84 RGB image from the wrist camera as the only
 505 observation. For *Tool Hang*, we use a front-view camera instead since the wrist-view is heavily
 506 occluded. For *Tool Hang Broad* and *Coffee Preparation*, we use both camera views plus proprio-
 507 ception state (end-effector pose and gripper finger width). We use the default CNN structure from
 508 DrQ-v2 to encode the image observations. For tasks with multiple observations, we first encode
 509 the image observations each with an independent CNN network, then concatenate the CNN outputs
 510 alongside the low-dimensional observations such as proprioception states to form the feature vector.

511 **Action.** All our tasks share a 7-dimensional (6-DOF delta movement of the end-effector and 1
 512 dimension for finger control) continuous action space. The action is modeled as a normal distribution
 513 with a scheduled standard deviation.

Table 2: Comparing the success rates of *Square* and *Square Broad* with and without TAMP.

Task	BC	RL	Ours
Square w/ TAMP	98%	100%	100%
Square w/o TAMP	2%	0%	94%
Square Broad w/ TAMP	100%	100%	100%
Square Broad w/o TAMP	0%	0%	0%

514 C Ablation: SPIRE without TAMP

515 We provide an additional ablation study on the high-level planner, TAMP. To do so, we treat the
 516 whole task as one handoff section. The agent only receives a reward of one if it completes the whole
 517 task. We collect 200 full demonstrations in *Square*, train a BC policy, and apply SPIRE to fine-
 518 tune the BC policy. Since the trajectory becomes longer and the robot now needs to handle object
 519 transportation, a single local wrist-view becomes insufficient. We thus include both the wrist view
 520 and the global front view, as well as the robot proprioception states in the observation for the w/o
 521 TAMP variant. The result is shown in Table 2.

522 Even though the w/o TAMP variant has more information from observations, the BC and RL policies
 523 are significantly worse than the w/ TAMP counterpart. The increased horizon makes the BC policy
 524 easier to drift away to regions less frequently visited in demonstrations and makes RL exploration
 525 much harder. In *Square*, despite the low starting quality, SPIRE still fine-tunes BC to reach a 94%
 526 success rate, demonstrating the effectiveness of RL fine-tuning. However, when the initialization
 527 range increases in *Square Broad*, even SPIRE fails to find an acceptable policy.

528 In conclusion, TAMP (1) confines the agent-controlled section to a small local area, reducing the
 529 need for global information, and (2) decreases the horizon (11.6 w/ TAMP, 101.7 w/o TAMP in
 530 *Square*) for the learned agent, reducing compounding errors and exploration difficulty.

D Comparison to Additional Methods

In each handoff section from TAMP, SPIRE utilizes the demonstrations by training a behavior cloning agent and using RL to fine-tune it. There are alternative methods to combine expert demonstrations and RL, which can be readily plugged in as replacements to SPIRE. In this section, we make connections from our method to GAIL [47]. The discriminator-based IRL reward in GAIL serves the same purpose as our KL penalty term - preventing the current policy from deviating from the expert policy. We draw further connection by showing that our KL penalty is the same as the IRL reward function in GAIL with an alternative discriminator objective and a different reward form.

Let π_E be the expert policy. The IRL reward function in GAIL is $-\log(1 - D(s, a))$, where $D : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is the discriminator that maximizes

$$J(D) := \mathbb{E}_{\tau \sim \pi}[\log(1 - D(s, a))] + \mathbb{E}_{\tau \sim \pi_E}[\log(D(s, a))] \quad (1)$$

If we use an alternative objective:

$$\hat{J}(D) := \mathbb{E}_{s \sim \pi_E, a \sim \text{Unif}}[-D(s, a)] + \mathbb{E}_{\tau \sim \pi_E}[\log(D(s, a))] \quad (2)$$

The alternative objective discriminates π_E from a fixed policy rather than the current learned policy π . Assume π_E has full support, then maximizing $\hat{J}(D)$ is equivalent to maximize for every $s \in \mathcal{S}$:

$$\hat{J}_s(D) := \mathbb{E}_{a \sim \text{Unif}}[-D(s, a)] + \mathbb{E}_{a \sim \pi_E(\cdot | s)}[\log(D(s, a))] \quad (3)$$

$$= - \left(\int D(s, a) da \right) + \left(\int \pi_E(a | s) \log(D(s, a)) da \right) \quad (4)$$

$$= - \left(\int D(s, a) da \right) + \left(\int \pi_E(a | s) \log \pi_E(a | s) da \right) + \left(\int \pi_E(a | s) \log \frac{D(s, a)}{\pi_E(a | s)} da \right) \quad (5)$$

$$= - \left(\int D(s, a) da \right) + H(\pi_E(\cdot | s)) + \left(\int \pi_E(a | s) \log \frac{D(s, a)}{\pi_E(a | s)} da \right) \quad (6)$$

$$\leq - \left(\int D(s, a) da \right) + H(\pi_E(\cdot | s)) + \left(\int \pi_E(a | s) \left(\frac{D(s, a)}{\pi_E(a | s)} - 1 \right) da \right) \quad (7)$$

$$= - \left(\int D(s, a) da \right) + H(\pi_E(\cdot | s)) + \left(\int D(s, a) da \right) - \left(\int \pi_E(a | s) da \right) \quad (8)$$

$$= H(\pi_E(\cdot | s)) - 1 \quad (9)$$

where H is the entropy. (7) holds since $\log x \leq x - 1$ for all $x > 0$, and only equates when $x = 1$, i.e., $\hat{D}(s, a) = \pi_E(a | s)$. Since (9) is a constant, the maximum of $\hat{J}(D)$ can be taken when (7) equates, which means the optimal solution of $\hat{J}(D)$ is $\hat{D}(s, a) = \pi_E(a | s)$. Our KL penalty then is equivalent to using an IRL reward of $\log(\hat{D}(s, a)) = \log \pi_E(a | s)$.

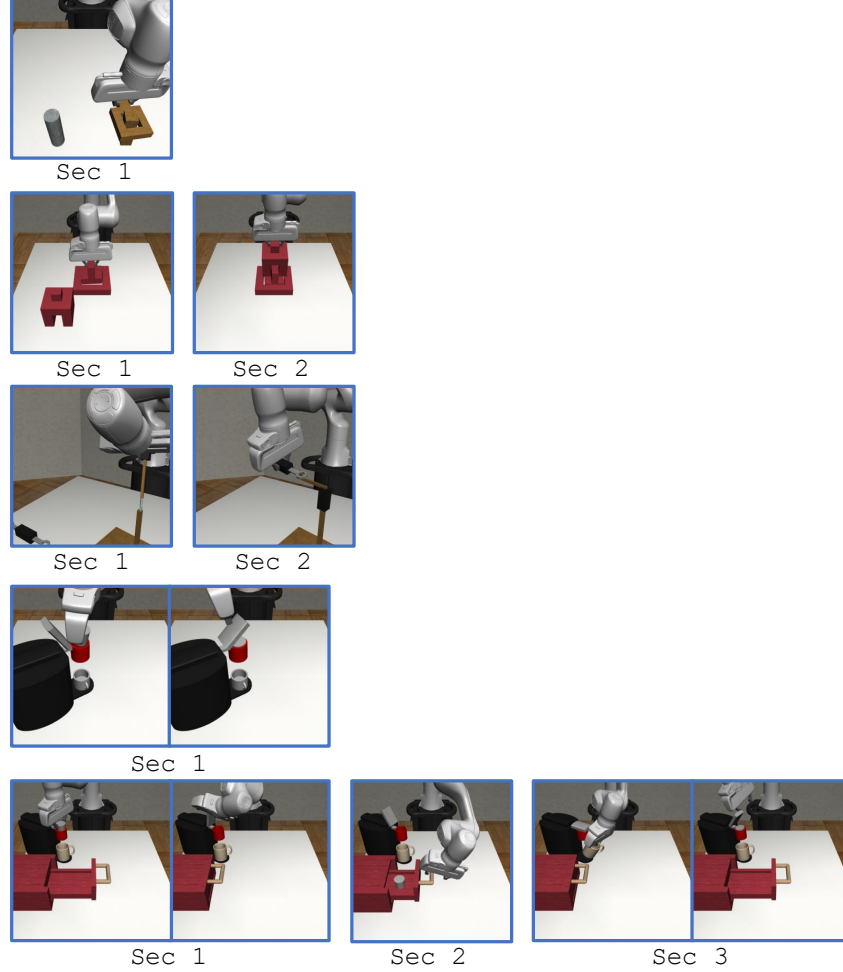


Figure 7: Handoff sections of every task. The tasks from top to bottom are: *Square*, *Three Piece*, *Tool Hang*, *Coffee*, *Coffee Preparation*.

E Tasks

We describe the nine tasks in the main paper in more detail.

Square and **Square Broad**. The robot must pick up a nut and place it onto a peg. This task has 1 handoff section, where the learned agent places the nut. The **Broad** version increases the initialization range of both the nut and the peg.

Three Piece and **Three Piece Broad**. The robot must assemble a structure by inserting one piece into a base and placing another piece on top of the first. This task has 2 handoff sections, where the learned agent places the two pieces. The **Broad** version increases the initialization range of all three pieces including the base.

Tool Hang and **Tool Hang Broad**. The robot must first insert a L-shaped piece into a base to assemble a frame, then hang a wrench off of the frame. This task has 2 handoff sections, where the learned agent inserts the L-shaped piece and hangs the wrench. The **Broad** version increases the initialization range of all three pieces (base, L-shaped hook, and wrench).

Coffee and **Coffee Broad**. The robot must pick up a coffee pod, insert it into a coffee machine, and close the lid. This task has 1 handoff section where the learned agent inserts the pod and closes the lid. The **Broad** version increases the initialization range of the pod and the coffee machine.

Coffee Preparation. This is an extended version of **Coffee**. The robot must place a mug onto the coffee machine, open the lid, open the drawer where the coffee pod is placed, pick up the pod, insert

566 the pod into the coffee machine, and finally close the lid. This task has 3 handoff sections where the
567 learned agent (1) places the mug and opens the lid, (2) opens the drawer, and (3) inserts the pod and
568 closes the lid.
569 See Figure 7 for an illustration of all the handoff sections.

Table 3: Mean and standard deviation (in parenthesis) of success rates out of 5 seeds.

	BC	RL [15]	Ours
Square	92.4 (5.5)	83.6 (36.7)	99.2 (1.8)
Square Broad	96.4 (4.1)	100.0 (0.0)	96.4 (5.4)
Coffee	96.8 (4.1)	40.0 (52.1)	88.0 (26.8)
Coffee Broad	41.6 (6.7)	23.2 (12.1)	84.4 (8.3)
Three Piece	63.6 (6.7)	0.0 (0.0)	84.0 (34.7)
Three Piece Broad	25.2 (7.7)	0.0 (0.0)	78.4 (5.0)
Tool Hang	9.2 (4.6)	0.0 (0.0)	54.0 (46.8)

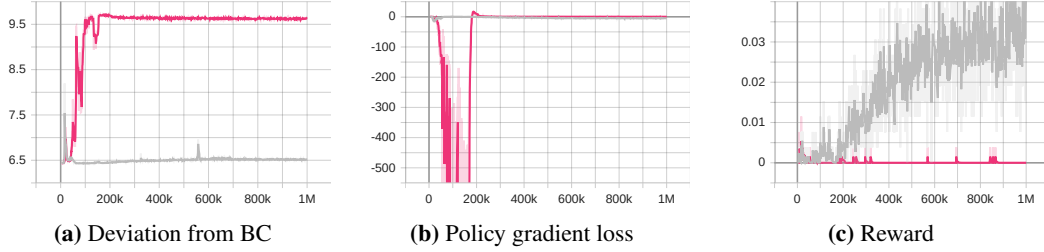


Figure 8: Comparing the (a) *Deviation from BC*, (b) *policy gradient loss*, and (c) *reward* training curves of a successful run (marked as grey) and a failed run (marked as red) in *Tool Hang*.

F Variance Across Seeds

In Figure 3, we show the best run out of 5 seeds. Here we provide the mean and standard deviation of the success rates in Table 3. We observe that although SPIRE still outperforms BC in terms of mean success rate in most of the tasks, our method exhibits unusually high variances in some of the tasks, for example, *Coffee*, *Three Piece*, and *Tool Hang*. In those tasks, one or more runs result in a performance significantly lower than the rest. Specifically,

- In *Coffee*, one run has 40% success rate, while the rest are all 100%;
- In *Three Piece*, one run has 22% success rate, while the rest are at least 98%;
- In *Tool Hang*, one run has 0% success rate and one has 6%, while the rest are at least 82%.

Reinforcement learning methods are known to have high variances, especially in sparse reward settings. SPIRE partially alleviates this problem by enforcing the KL penalty for deviating from an anchor policy. However, in practice, such deviation can still happen.

Figure 8 compares the training curve of a successful run (with 88% final success rate) and a failed run (with 0% final success rate). The policy in the failed run drastically deviated from the BC policy early on in the training. This is likely related to the unusually large policy gradient loss, which the KL penalty term was unable to match and failed to constrain the policy.

In our experiments, such an abrupt decrease in policy gradient loss happens frequently, with varying scales and timing, causing the training results to have high variance. Using an adaptive weight of the KL penalty might be a potential solution, which we wish to investigate in future work.

We do not believe 5 seeds are enough to quantitatively reflect the chance of such sudden deviation happening. An alternative solution would be to compare only the results where such deviation did not happen, which is why we chose to report the top-1 performing seed in our main paper.