A Environment Details

Here we provide additional details for our simulation environments. An unabridged version of the description from Section 5 is as follows:

- `Push (2D)`: Push a round puck using the tool such that it stops at the specified goal location. The goal space is a subset of 2D final puck locations $\mathcal{G} \subset \mathbb{R}^2$, and the control action space $\mathcal{A}_C \in \mathbb{R}^2$ specifies the x and y tool velocities.
- `Catch balls (2D)`: Use the tool to catch three balls that fall from the sky. The agent's goal is to catch all three balls, which start from varying locations on the x-y plane. We use a 1-dimensional control action space that specifies the x velocity of the tool at each step.
- `Scoop (2D)`: Use the tool to scoop balls out of a reservoir containing 40 total balls. Here we specify goals of scooping $x \in \{1, 2, ..., 7\}$ balls. The control action space $\mathcal{A}_C \in \mathbb{R}^3$ specifies the velocity of the rigid tool in x and y directions, along with its angular velocity.
- `Fetch cube (3D)`: Use the tool to retrieve an object randomly positioned beneath a vertical overhang. This task is additionally challenging because the position of the robot end effector is restricted to a rectangular region in the x-y plane of dimensions $0.8\text{m} \times 0.2\text{m}$ to avoid collision with the overhang. The tool is a three-link chain where each link is a box parameterized by its width, length, and height. The design space also includes the relative angle between two connected links, with a total of $n = 11$ parameters. The control action space $\mathcal{A}_C \in \mathbb{R}^3$ represents a change in end-effector position.
- `Lift cup (3D)`: Use the tool to lift a cup of randomized geometry from the ground into the air. This task requires careful design of the tool to match cup geometry. The tool is a four-link fork with two prongs parameterized by the separation, tilt angle, width, length, and height of the prongs. The same parameters are applied to both prongs to maintain symmetry. The handle dimensions are fixed. The design space has $n = 5$ parameters. $\mathcal{A}_C \in \mathbb{R}^3$ represents a change in end-effector position.
- `Scoop (3D)`: A 3D analog of the 2D `scoop` task. This task has the same goal space as the 2D `scoop` task, but the tool in 3D is a six-link scoop composed of a rectangular bottom plate parameterized by its width and length, and four rectangular side plates attached to each side of the bottom plate. Each side plate is parameterized by its height and relative angle to the bottom plate. A handle with fixed dimensions is attached to one of the side plates. There are $n = 10$ total design parameters. $\mathcal{A}_C \in \mathbb{R}^6$ represents a change in end-effector pose.

B Experimental Details

B.1 Training Hyperparameters & Architecture Details for Our Framework

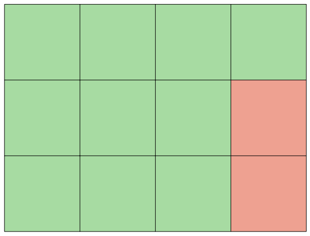
In Tables 5, 6, 7, 8, 9, and 10, we provide detailed hyperparameters for our framework for each environment. Unless otherwise specified, we use the neural network architectures for the design policy, control policy, and value function from [7].

B.2 Training Hyperparameters & Architecture Details for Baselines

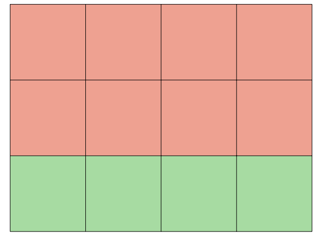
For the CMA-ES baseline, we perform hyperparameter sweeps for a fair comparison with our framework. For the CMA-RL baseline, we use the same set of best performing hyperparameters for the outer CMA-ES loop. The tested hyperparameter configurations for each baseline are listed in Table 2. Except model architecture differences, we use the same optimization hyperparameters for Ours, Ours(shared arch.), and HWasP-minimal.

B.3 Generalization to Unseen Goals Experiment Details

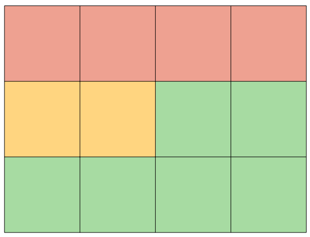
For `Fetch cube`, the rectangular region of initial poses is defined by $x \in [-0.395, 0.395]$ and $y \in [0.4, 0.7]$. The cutout region corresponds to two disconnected rectangular patches contained in



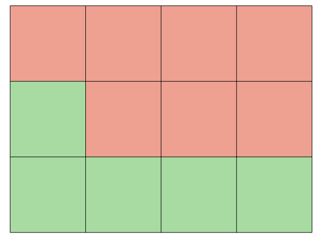
(a) Tool 1



(b) Tool 2



(c) Tool 3



(d) Tool 4

Figure 10: Heatmaps of success and failures for trials where fixed tools are used for a range of initializations. The grid here directly maps to the set of 2D initial cube locations tested for each tool, from the point of view of the robot. Here green indicates a success, red indicates failure, and orange indicates a failure where the final cube position is within 5cm of success.

Method	**Hyperparameters**	**Values**
CMA-ES	Population Size	10, **24**, 100, 1000
	Initial Stdev	**0.1**, 1.0, 10.0
	Center Learning Rate	0.01, 0.1, **1.0**
	Covariance Learning Rate	0.01, 0.1, **1.0**
	Rank μ Learning Rate	0.01, 0.1, **1.0**
	Rank One Learning Rate	0.01, 0.1, **1.0**
CMA-RL	Poicy Net	(256, 256, 256, 256, 256)
	Value Net	(256, 256, 256, 256, 256)
	Learning Rate	3e-4
	Batch Size	(50000, 20000(3D scoop))
	Minibatch Size	2000
Ours(shared arch.)	Poicy Net	(256, 256, 256, 256, 256)
	Value Net	(256, 256, 256, 256, 256)
	Learning Rate	1e-4
	Batch Size	(50000, 20000(3D scoop))
	Minibatch Size	2000

Table 2: We tune over these values for hyperparameters of baseline methods. Bolded values indicate the best performing settings for CMA-ES, which we use in our comparisons.

the training region defined by $x_1 \in [-0.350, -0.045]$, $y_1 \in [0.434, 0.666]$ and $x_2 \in [0.045, 0.350]$, $y_2 \in [0.434, 0.666]$ respectively.

Zero-shot performance. We train six policies using our framework where the cutout region removes a fraction of the total training area equal to 0.1, 0.2, 0.4, 0.6, 0.8, and 0.9 respectively.

Fine-tuning performance. For the fine-tuning experiment, we specifically select four initializations that we find our policies do not complete successfully zero-shot: $\{(-0.167, 0.367), (-0.129, 0.357), (0.430, 0.493), (0.415, 0.610)\}$.

B.4 Trading off Design and Control Complexity Experiment Details

We train four agents independently on `catch balls`, setting the value of α, the tradeoff reward parameter defined in Equation 1, to 0, 0.3, 0.7, and 1.0 respectively.

B.5 Real Robot Experiment Details

For our real-world experiments, we use a Franka Emika Panda arm. We control the robot using an impedance controller from the Polymetis [38] library.

Tools are 3D printed using polylactic acid (PLA) on commercially available Ender 3 and Ender 5 printers with nozzle diameter 0.4mm. We print using a layer height of 0.3mm and 10% infill. We perform slicing using the Ultimaker CURA software.

We roll out each policy for 100 environment steps or until a success is detected.

For the `fetch cube` task, we measure the success based on whether the center of mass of the 5cm cube is closer than 0.5m from the base of the robot. Please see Table 3 for per-tool details. The tool images are shown in Figure 9, from left to right: Tools 4, 2, 3, 1 respectively.

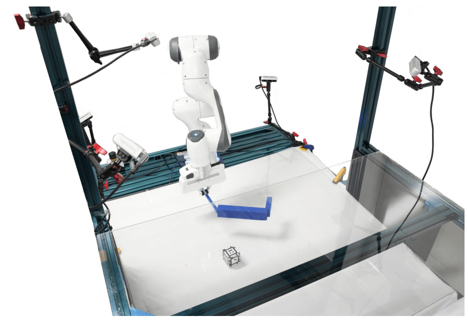


Figure 11: Real world setup. We use a Franka Panda arm and five RealSense D435 cameras for tracking.

Tool	Initial cube position (x, y, z)
Tool 1	(-0.110, -0.803, 0.025)
Tool 2	(-0.339, -0.588, 0.025)
Tool 3	(0.155, -0.731, 0.025)
Tool 4	(0.211, -0.633, 0.025)

Table 3: Initial cube positions corresponding to tools fabricated in real experiments.

For the `lift cup` task, we measure success based on whether the cup has been lifted higher than 0.4m off of the plane of the workspace. Please see Table 4 for per-tool details. The tool images are shown in Figure 9, from left to right: Tools 1, 2, 3, 4.

Tool	Cup geometry parameters (length/width, height)
Tool 1	(0.3, 0.6)
Tool 2	(0.3, 0.9)
Tool 3	(0.5, 0.8)
Tool 4	(0.9, 0.6)

Table 4: Cup geometry parameters corresponding to tools fabricated in real experiments. Note that the length and width parameters share a single value.

We also present detailed results for the `fetch cube` experiments using tools generated for a specific initial position for a range of initializations. Recall that we test the policies on a 3×4 grid of initial positions that span a range of 12cm $\times$ 85.6cm, for a total of 12 trials per tool. We plot the successes and failures for each tool according to geometric position in Figure 10. We can see that the control policy is able to use each tool to solve the task for several initializations, but each tool is specialized for particular regions.

Hyperparameter	Value
Tool Position Init.	(20, 10)
Control Steps Per Action	1
Max Episode Steps	150
Slack Reward	-0.001
Tool Length Ratio	(-0.5, 0.5)
Tool Length Init.	(2.0, 2.0, 2.0)
Tool Angle Init.	(0.0, 0.0, 0.0)
Tool Angle Ratio	(-1.0, 1.0)
Tool Angle Scale	90.0
Control GNN	(64, 64, 64)
Control Index MLP	(128, 128)
Design GNN	(64, 64, 64)
Design Index MLP	(128, 128)
Control Log Std.	-1.0
Design Log Std.	-2.3
Fix Design & Control Std.	True
Policy Learning Rate	2e-5
Entropy β	0.01
Value Learning Rate	1e-4
KL Divergence Threshold	0.005
Batch Size	50000
Minibatch Size	2000
PPO Steps Per Batch	10

Table 5: Hyperparameters used for our framework on the `push` task.

Hyperparameter	Value
Tool Position Init.	(20, 10)
Control Steps Per Action	1
Max Episode Steps	150
Slack Reward	-0.001
Tool Length Ratio	(-0.5, 2.0)
Tool Length Init.	(2.0, 1.0, 1.0)
Tool Angle Init.	(0.0, 0.0, 0.0)
Tool Angle Ratio	(-1.0, 1.0)
Tool Angle Scale	60.0
Control GNN	(64, 64, 64)
Control Index MLP	(128, 128)
Design GNN	(64, 64, 64)
Design Index MLP	(128, 128)
Control Log Std.	0.0
Design Log Std.	0.0
Fix Design & Control Std.	True
Policy Learning Rate	2e-5
Entropy β	0.01
Value Learning Rate	1e-4
KL Divergence Threshold	0.002
Batch Size	50000
Minibatch Size	2000
PPO Steps Per Batch	10

Table 6: Hyperparameters used for our framework on the `catch balls` task.

Hyperparameter	Value
Tool Position Init.	(15, 10)
Control Steps Per Action	5
Max Episode Steps	30
Slack Reward	-0.001
Tool Length Ratio	(-0.7, 0.2)
Tool Length Init.	(6.0, 3.0, 3.0)
Tool Angle Init.	(0.0, 0.0, 0.0)
Tool Angle Ratio	(-0.1, 0.7)
Tool Angle Scale	90.0
Control GNN	(64, 64, 64)
Control Index MLP	(128, 128)
Design GNN	(64, 64, 64)
Design Index MLP	(128, 128)
Control Log Std.	0.0
Design Log Std.	0.0
Fix Design & Control Std.	True
Policy Learning Rate	2e-5
Entropy β	0.01
Value Learning Rate	3e-4
KL Divergence Threshold	0.1
Batch Size	50000
Minibatch Size	2000
PPO Steps Per Batch	10

Table 7: Hyperparameters used for our framework on the `scoop` task.

Hyperparameter	Value
Tool Position Init.	(0.0, 0.5, 0.02)
Control Steps Per Action	10
Max Episode Steps	100
Slack Reward	-0.001
Success Reward	10.0
Box Dimensions Min	(0.005, 0.05, 0.005)
Box Dimensions Max	(0.015, 0.1, 0.02)
Tool Angle Min	(-90.0, -90.0, -90.0)
Tool Angle Max	(90.0, 90.0, 90.0)
Control Action Min	(-1.0, -1.0, -1.0, -0.2, -0.2, -0.2)
Control Action Max	(1.0, 1.0, 1.0, 0.2, 0.2, 0.2)
Control Action Scale	0.1
Control GNN	(128, 128, 128)
Control Index MLP	(128, 128)
Design GNN	(128, 128, 128)
Design Index MLP	(128, 128)
Control Log Std.	0.0
Design Log Std.	0.0
Fix Design & Control Std.	False
Policy Learning Rate	1e-4
Entropy β	0.0
Value Learning Rate	3e-4
KL Divergence Threshold	0.5
Batch Size	50000
Minibatch Size	2000
PPO Steps Per Batch	10

Table 8: Hyperparameters used for our framework on the `fetch cube` task.

Hyperparameter	Value
Tool Position Init.	(0.0, 1.2, 0.05)
Control Steps Per Action	10
Max Episode Steps	150
Slack Reward	-0.001
Success Reward	10.0
Box Dimensions Min	(0.005, 0.02, 0.01)
Box Dimensions Max	(0.01, 0.1, 0.03)
Tool Angle Min	(-30.0, -30.0, -30.0)
Tool Angle Max	(30.0, 30.0, 30.0)
Control Action Min	(-1.0, -1.0, -1.0, -1.57, -1.57, -1.57)
Control Action Max	(1.0, 1.0, 1.0, 1.57, 1.57, 1.57)
Control Actioin Scale	0.1
Control GNN	(128, 128, 128)
Control Index MLP	(128, 128)
Design GNN	(128, 128, 128)
Design Index MLP	(128, 128)
Control Log Std.	0.0
Design Log Std.	-1.0
Fix Design & Control Std.	True
Policy Learning Rate	2e-5
Entropy β	0.01
Value Learning Rate	3e-4
KL Divergence Threshold	0.5
Batch Size	50000
Minibatch Size	2000
PPO Steps Per Batch	5

Table 9: Hyperparameters used for our framework on the `lift cup` task.

Hyperparameter	**Value**
Tool Position Init.	(0.0, 0.05, 0.1)
Control Steps Per Action	10
Max Episode Steps	100
Slack Reward	-0.001
Success Reward	10.0
Box Dimensions Min	(0.04, 0.005, 0.02)
Box Dimensions Max	(0.08, 0.005, 0.05)
Tool Angle Min	(-15.0, -15.0, -15.0)
Tool Angle Max	(15.0, 15.0, 15.0)
Control Action Min	(-1.0, -1.0, -1.0, -1.57, -1.57, -1.57)
Control Action Max	(1.0, 1.0, 1.0, 1.57, 1.57, 1.57)
Control Action Scale	0.05
Control GNN	(128, 128, 128)
Control Index MLP	(128, 128)
Design GNN	(128, 128, 128)
Design Index MLP	(128, 128)
Control Log Std.	0.0
Design Log Std.	0.0
Fix Design & Control Std.	False
Policy Learning Rate	1e-4
Entropy β	0.01
Value Learning Rate	3e-4
KL Divergence Threshold	0.5
Batch Size	20000
Minibatch Size	2000
PPO Steps Per Batch	5

Table 10: Hyperparameters used for our framework on the `3D scoop` task.