A Training pseudocode

Algorithm 1 SEQUENTIAL DEXTERITY: A bi-directional optimization framework for skill chaining

Require: sub-task MDPs $\mathcal{M}_1,...,\mathcal{M}_K$

Initialize sub-policies $\pi_\theta^1,...,\pi_\theta^K$, transition feasibility function $F_\omega^1,...,F_\omega^K$, ternimal state buffers $\mathcal{B}_\mathcal{I}^1,...,\mathcal{B}_\mathcal{I}^K$, the sum of reward buffers $\mathcal{B}_\mathcal{R}^1,...,\mathcal{B}_\mathcal{R}^K$
for iteration $m=0,1,...,M$ **do**
  for each subtask $i=1,...,K$ **do**
    while until convergence of π_θ^i **do**
      Rollout trajectories $\tau=(s_0,a_0,r_0,...,s_T)$ with π_θ^i
      Update π_θ^i by maximizing $\mathbb{E}_{\pi^i}[\sum_{t=0}^{T-1}\gamma^t r_t^i]$
    end while
  end for ▷ Forward initialization
  for each subtask $i=K,...,1$ **do**
    while until convergence of π_θ^i **do**
      Sample s_0 from environment or $\mathcal{B}_\beta^{i-1}$
      Rollout trajectories $\tau=(s_0,a_0,r_0,...,s_T)$ with π_θ^i
      if $\boldsymbol{c}_t^i > h_t^i$ or terminate from environment **then**
        $\mathcal{B}_\mathcal{T}^i \leftarrow \mathcal{B}_\mathcal{T}^i \cup s_{[T-10:T]}, \mathcal{B}_\mathcal{R}^i \leftarrow \mathcal{B}_\mathcal{R}^i \cup [\sum_{t=0}^{T-1} r_t^i]$
      end if
      Update F^i with $s_{[T-10:T]} \sim \mathcal{B}_\mathcal{T}^{i-1}$ and $[\sum_{t=0}^{T-1}\boldsymbol{r}_t] \sim \mathcal{B}_\mathcal{R}^i$
      Update π_θ^i by maximizing $\mathbb{E}_{\pi^i}[\sum_{t=0}^{T-1}\gamma^t r_t^i]$
    end while
  end for ▷ Backward finetuning
end for

B Real-world system setups

During real-world deployment, some observations used in the simulation are hard to accurately estimate (e.g., joint velocity, object velocity, etc.). We use the teacher-student policy distillation framework [6, 7, 49] to abstract away these observation inputs from the policy model. In each policy rollout, our system first uses the top-down camera view to perform a color-based segmentation to localize the target block piece given by the manual. Then, the robot calls motion planning API to move to the target location with OSC controller [50]. After that, our system uses the wrist camera view to track the segmentation and 6D pose of the object with a combination of color-based initial segmentation, Xmem segmentation tracker [51], and Densefusion pose estimator [52]. If the target object is deeply buried (as the case in the top left corner of Fig. 4), the transition feasibility function will inform the robot to execute the searching policy until the target appears. During the last insertion stage, the estimated 6D object pose will guide the robot policy to adjust its finger and wrist motion to align with the goal location as it learned in the simulation. Since simulating contact-rich insertion is still a research challenge in graphics, after the robot has placed the block to the target location, we perform a scripted pressing motion (spread out the entire hand and press down) on the target location to ensure a firm insert. More details about real-world system setups and results can be found in the Supplementary video.

C State Space in Simulation

C.1 Building Blocks

Searching Table.4 gives the specific information of the state space of the searching task.

Orienting Table.5 gives the specific information of the state space of the orienting task.

Table 4: Observation space of Search task.

Index	Description
0 - 23	dof position
23 - 46	dof velocity
46 - 98	fingertip pose, linear velocity, angle velocity (4 x 13)
98 - 111	hand base pose, linear velocity, angle velocity
111 - 124	object base pose, linear velocity, angle velocity
124 - 143	the actions of the last timestep
143 - 159	motor tactile
159 - 160	the number of pixels of the object exposed under the camera

Table 5: Observation space of Orient and Grasp task.

Index	Description
0 - 23	dof position
23 - 46	dof velocity
46 - 98	fingertip pose, linear velocity, angle velocity (4 x 13)
98 - 111	hand base pose, linear velocity, angle velocity
111 - 124	object base pose, linear velocity, angle velocity
124 - 143	the actions of the last timestep
143 - 159	motor tactile

Grasping Table.5 gives the specific information of the state space of the grasping task.

Inserting Table.6 gives the specific information of the state space of the inserting task.

Table 6: Observation space of Insert task.

Index	Description
0 - 23	dof position
23 - 46	dof velocity
46 - 98	fingertip pose, linear velocity, angle velocity (4 x 13)
98 - 111	hand base pose, linear velocity, angle velocity
111 - 124	object base pose, linear velocity, angle velocity
124 - 143	the actions of the last timestep
143 - 159	motor tactile
159 - 166	goal pose
166 - 169	goal position - object position
169 - 173	goal rotation - object rotation

C.2 Tool positioning

Grasping Table.5 gives the specific information of the state space of the grasping task.

In-hand Orientation Table.6 gives the specific information of the state space of the in-hand orientation task.

D Reward functions

D.1 Building Blocks

Searching Denote the τ is the commanded torques at each timestep, the number of pixels of the object exposed under the camera as $\boldsymbol{P}$, the sum of the distance between each fingertip and the object as

Table 7: Domain randomization of all the sub-tasks.

Parameter	Type	Distribution	Initial Range
Robot			
Mass	Scaling	uniform	[0.5, 1.5]
Friction	Scaling	uniform	[0.7, 1.3]
Joint Lower Limit	Scaling	loguniform	[0.0, 0.01]
Joint Upper Limit	Scaling	loguniform	[0.0, 0.01]
Joint Stiffness	Scaling	loguniform	[0.0, 0.01]
Joint Damping	Scaling	loguniform	[0.0, 0.01]
Object			
Mass	Scaling	uniform	[0.5, 1.5]
Friction	Scaling	uniform	[0.5, 1.5]
Scale	Scaling	uniform	[0.95, 1.05]
Observation			
Obs Correlated. Noise	Additive	gaussian	[0.0, 0.001]
Obs Uncorrelated. Noise	Additive	gaussian	[0.0, 0.002]
Action			
Action Correlated Noise	Additive	gaussian	[0.0, 0.015]
Action Uncorrelated Noise	Additive	gaussian	[0.0, 0.05]
Environment			
Gravity	Additive	normal	[0, 0.4]

$\sum_{i=0}^{4}\boldsymbol{f}_i$, the action penalty as $\|\boldsymbol{a}\|_2^2$, and the torque penalty as $\|\tau\|_2^2$. Finally, the rewards are given by the following specific formula:

$$r = \lambda_1 * \boldsymbol{P} + \lambda_2 * min(\sum_{i=0}^{4}\boldsymbol{f}_i - e_0, 0) + \lambda_3 * \|\boldsymbol{a}\|_2^2 + \lambda_4 * \|\tau\|_2^2 \tag{3}$$

where $\lambda_1 = 5.0$, $\lambda_2 = 1.0$, $\lambda_3 = -0.001$, $\lambda_4 = -0.003$, and $e_0 = 0.2$.

Orienting Denote the τ is the commanded torques at each timestep, the angle of rotation of the object as θ, the sum of the distance between each fingertip and the object as $\sum_{i=0}^{4}\boldsymbol{f}_i$, the action penalty as $\|\boldsymbol{a}\|_2^2$, and the torque penalty as $\|\tau\|_2^2$. Finally, the rewards are given by the following specific formula:

$$r = \lambda_1 * \theta + \lambda_2 * min(\sum_{i=0}^{4}\boldsymbol{f}_i - e_0, 0) + \lambda_3 * \|\boldsymbol{a}\|_2^2 + \lambda_4 * \|\tau\|_2^2 \tag{4}$$

where $\lambda_1 = 1.0$, $\lambda_2 = 1.0$, $\lambda_3 = -0.001$, $\lambda_4 = -0.003$, and $e_0 = 0.6$.

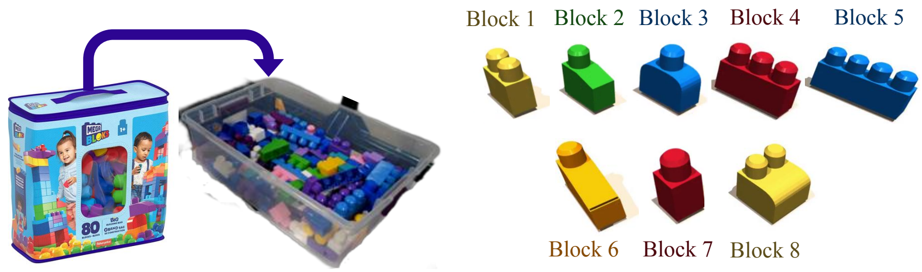
Grasping Denote the τ is the commanded torques at each timestep, the sum of the distance between each fingertip and the object as $\sum_{i=0}^{4}\boldsymbol{f}_i$, the action penalty as $\|\boldsymbol{a}\|_2^2$, and the torque penalty as $\|\tau\|_2^2$. Finally, the rewards are given by the following specific formula:

$$r = \lambda_1 * exp[\alpha_0 * min(\sum_{i=0}^{4}\boldsymbol{f}_i - e_0, 0)] + \lambda_2 * \|\boldsymbol{a}\|_2^2 + \lambda_3 * \|\tau\|_2^2 \tag{5}$$

where $\lambda_1 = 1.0$, $\lambda_2 = -0.001$, $\lambda_3 = -0.003$, $\alpha_0 = -5.0$, and $e_0 = 0.1$. It is worth noting that in the latter half of our grasping training, we force the hand to lift, so if the grip is unstable, the object will drop and the reward will decrease.

Inserting Denote the τ is the commanded torques at each timestep, the object and goal position as x_o and x_g, the angular position difference between the object and the goal as d_a, the sum of the distance between each fingertip and the object as $\sum_{i=0}^{4}\boldsymbol{f}_i$, the action penalty as $\|\boldsymbol{a}\|_2^2$, and the torque penalty as $\|\tau\|_2^2$. Finally, the rewards are given by the following specific formula:

$$\begin{aligned} r = &\lambda_1 * exp[-(\alpha_0 * \|x_o - x_g\|_2 + \alpha_1 * 2 * \arcsin(clamp(\|d_a\|_2, 0, 1)))] + \\ &\lambda_2 * min(\sum_{i=0}^{4}\boldsymbol{f}_i - e_0, 0) + \lambda_3 * \|\boldsymbol{a}\|_2^2 + \lambda_4 * \|\tau\|_2^2 \end{aligned} \tag{6}$$



(a) Real world (b) Simulation

Figure 6: The block model we use in simulation and real-world. (b) is the eight blocks used in our building blocks task. The upper Block 1-5 is the training block, and the lower Block 6-8 is the unseen block for testing.

where $\lambda_1 = 1.0$, $\lambda_2 = 0.0$, $\lambda_3 = -0.001$, $\lambda_4 = -0.003$, $\alpha_0 = 1.0$, $\alpha_1 = 20.0$, and $e_0 = 0.06$.

D.2 Tool positioning

Grasping Denote the τ is the commanded torques at each timestep, the sum of the distance between each fingertip and the object as $\sum_{i=0}^{4} \boldsymbol{f}_i$, the action penalty as $\|\boldsymbol{a}\|_2^2$, and the torque penalty as $\|\tau\|_2^2$. Finally, the rewards are given by the following specific formula:

$$r = \lambda_1 * exp[\alpha_0 * min(\sum_{i=0}^{4} \boldsymbol{f}_i - e_0, 0)] + \lambda_2 * \|\boldsymbol{a}\|_2^2 + \lambda_3 * \|\tau\|_2^2 \tag{7}$$

where $\lambda_1 = 1.0$, $\lambda_2 = -0.001$, $\lambda_3 = -0.003$, $\alpha_0 = -5.0$, and $e_0 = 0.1$. It is worth noting that in the latter half of our grasping training, we force the hand to lift, so if the grip is unstable, the object will drop and the reward will decrease.

In-hand Orientation Denote the τ is the commanded torques at each timestep, the object and goal position as x_o and x_g, the angular position difference between the object and the goal as d_a, the sum of the distance between each fingertip and the object as $\sum_{i=0}^{4} \boldsymbol{f}_i$, the action penalty as $\|\boldsymbol{a}\|_2^2$, and the torque penalty as $\|\tau\|_2^2$. Finally, the rewards are given by the following specific formula:

$$\begin{aligned} r = &\lambda_1 * exp[-(\alpha_0 * \|x_o - x_g\|_2 + \alpha_1 * 2 * \arcsin(clamp(\|d_a\|_2, 0, 1)))] + \\ &\lambda_2 * min(\sum_{i=0}^{4} \boldsymbol{f}_i - e_0, 0) + \lambda_3 * \|\boldsymbol{a}\|_2^2 + \lambda_4 * \|\tau\|_2^2 \end{aligned} \tag{8}$$

where $\lambda_1 = 1.0$, $\lambda_2 = 0.0$, $\lambda_3 = -0.001$, $\lambda_4 = -0.003$, $\alpha_0 = 1.0$, $\alpha_1 = 20.0$, and $e_0 = 0.06$.

E Domain Randomization

Isaac Gym provides lots of domain randomization functions for RL training. We add the randomization for all the sub-tasks as shown in Table. 7 for each environment. we generate new randomization every 1000 simulation steps.

F Task Setups

F.1 Building Blocks

Block model. For the building blocks task, we use the same model as Mega Bloks[1] as our blocks. It is a range of large, stackable construction blocks designed specifically for the small hands of the children.

[1]https://www.megabrands.com/en-us/mega-bloks.

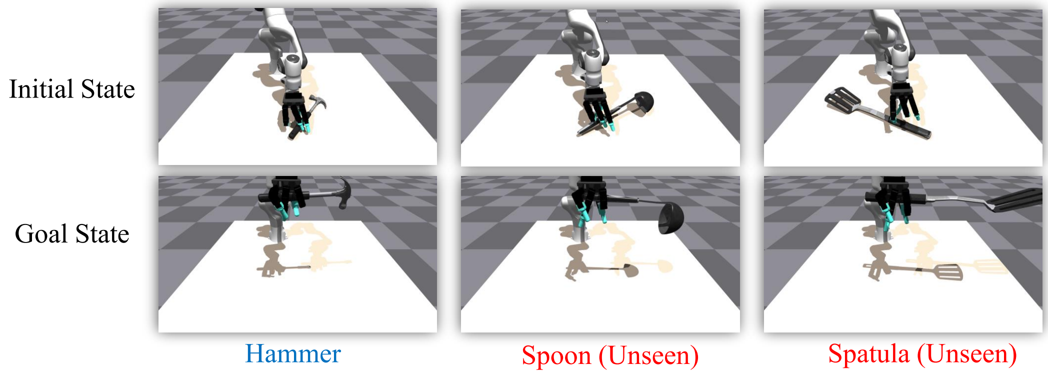


Figure 7: Visualization of the three tools we use in Tool Positioning task. The Hammer is use for training and the Spoon and Spatula is only use for testing. We also show the goal pose of the tools.

Figure 8: Snapshot of the searching task.

Figure 9: Snapshot of the orienting task.

We take eight different types of blocks (denoted as Block 1, Block 2,..., Block 8) as the models of our block, and carefully measured the dimensions to ensure that they were the same as in the real world. The block datasets is shown in Figure. 6. For all building block sub-tasks, we use Block 1-5 as the training object and Block 6-8 as the unseen object for testing.

Physics in insertion between two blocks. It is difficult to simulate the realistic insertion in the simulator, and it is easy to explode or model penetration when the two models are in frequent contact. Therefore, we want the plug and slot between the two blocks can be inserted without frequent friction. We reduced the diameter of all block plugs and convex decomposed them via VHACD method when loaded into Isaac Gym. Finally, we made one block possible to insert another block through free fall to verify the final effect.

F.2 Tool positioning

For the tool positioning task, we have a total of three tools: hammer, spatula, and spoon. We use the hammer for training and test both in the hammer, spatula, and spoon. This long-horizon task involves grasp a tool and re-orient it onto a pose suitable for its use. Fig.7 shows what they look like and the initial and goal state of the each three tools.

F.3 Typical frames of all sub-tasks

For the convenience of readers, we show some typical frames of all the sub-tasks in simulation.

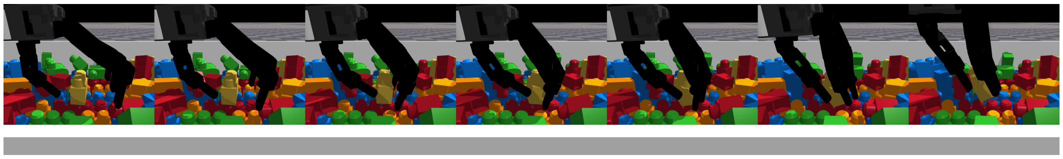


Figure 10: Snapshot of the grasping task.

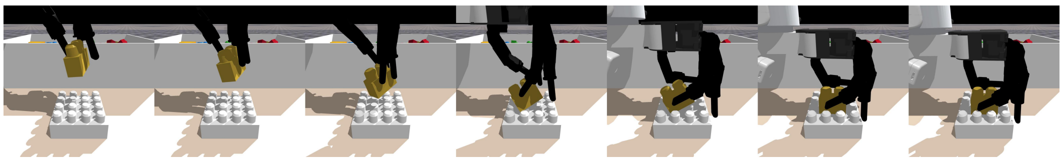


Figure 11: Snapshot of the inserting task.

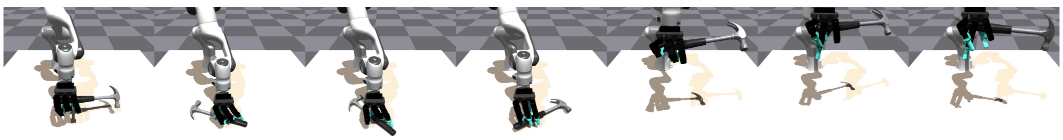


Figure 12: Snapshot of the hammer positioning.

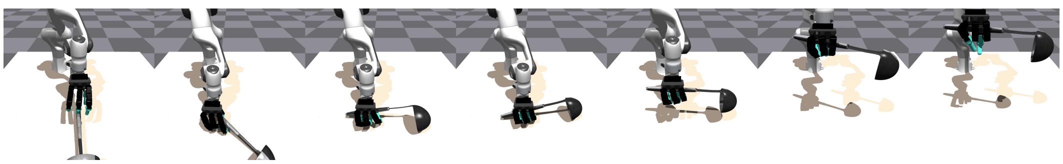


Figure 13: Snapshot of the spoon positioning.

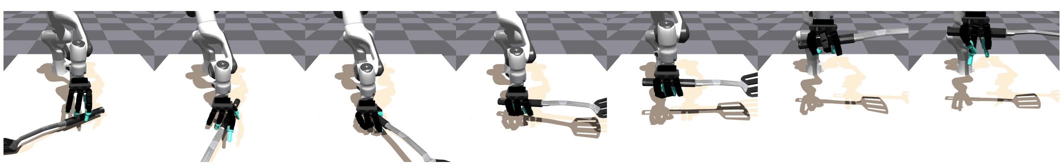


Figure 14: Snapshot of the spatula positioning.

	Trained	Unseen	All
Ours w/o belief state	$0.40_{\pm 0.08}$	$0.16_{\pm 0.07}$	$0.29_{\pm 0.06}$
Ours w/o tactile	$\mathbf{0.43}_{\pm \mathbf{0.04}}$	$0.33_{\pm 0.00}$	$0.37_{\pm 0.02}$
Ours w/o both	$0.26_{\pm 0.05}$	$0.02_{\pm 0.01}$	$0.14_{\pm 0.02}$
Ours	$\mathbf{0.43}_{\pm \mathbf{0.04}}$	$\mathbf{0.36}_{\pm \mathbf{0.04}}$	$\mathbf{0.38}_{\pm \mathbf{0.04}}$

Table 8: Ablation study on the system choices in single-step **Orient** task.

F.3.1 Building Blocks

F.3.2 Tool Positioning

G Motor tactile and belief state.

We found that motor tactile and belief state are beneficial for dexterous in-hand manipulation. Tab. 8 is the ablation study of the design choices of our input state space. We modify the goal of the Orient sub-task in the building blocks task to a pre-defined goal orientation and train each ablation method only on this sub-policy. We find the belief state pose estimator has the highest improvement (9% in task success rate), which highlights its effects on in-hand manipulation.

G.1 Hyperparameters of the PPO

G.1.1 Building Blocks

Table 9: Hyperparameters of PPO in Building Blocks.

Hyperparameters	Searching	Orienting	Grasping & Inserting
Num mini-batches	4	4	8
Num opt-epochs	5	10	2
Num episode-length	8	20	8
Hidden size	[1024, 1024, 512]	[1024, 1024, 512]	[1024, 1024, 512]
Clip range	0.2	0.2	0.2
Max grad norm	1	1	1
Learning rate	3.e-4	3.e-4	3.e-4
Discount (γ)	0.96	0.96	0.9
GAE lambda (λ)	0.95	0.95	0.95
Init noise std	0.8	0.8	0.8
Desired kl	0.016	0.016	0.016
Ent-coef	0	0	0

G.1.2 Tool Positioning

Table 10: Hyperparameters of PPO in Tool Positioning.

Hyperparameters	Grasping	In-hand Orienting
Num mini-batches	4	4
Num opt-epochs	5	10
Num episode-length	8	20
Hidden size	[1024, 1024, 512]	[1024, 1024, 512]
Clip range	0.2	0.2
Max grad norm	1	1
Learning rate	3.e-4	3.e-4
Discount (γ)	0.96	0.96
GAE lambda (λ)	0.95	0.95
Init noise std	0.8	0.8
Desired kl	0.016	0.016
Ent-coef	0	0