A Detail Description of Environments

A.1 Open-Loop Environments

Block Stacking In the Block Stacking environment (Figure 6a), there are N cubic blocks with a size of $3cm \times 3cm \times 3cm$. The blocks are randomly initialized in the workspace. The goal of this task is to stack all blocks in a stack. An optimal policy requires $2(N-1)$ steps to finish this task. The number of blocks N is configurable. By default, $N = 4$, and the maximal number of steps per episode is 10.

House Building 1 In the House Building 1 environment (Figure 6b), there are $N-1$ cubic blocks with a size of $3cm \times 3cm \times 3cm$ and one triangle block with a bounding box size of around $3cm \times 3cm \times 3cm$. The blocks are randomly initialized in the workspace. The goal of this task is to first form a stack using the $N-1$ cubic blocks, then place the triangle block on top of the stack. An optimal policy requires $2(N-1)$ steps to finish this task. The number of blocks N is configurable. By default, $N = 4$, and the maximal number of steps per episode is 10.

House Building 2 In the House Building 2 environment (Figure 6c), there are two cubic blocks with a size of $3cm \times 3cm \times 3cm$, and a roof block with a bounding box size of around $12cm \times 3cm \times 3cm$. The blocks are randomly initialized in the workspace. The goal of this task is to place the two cubic blocks next to each other, then place the roof block on top of the two cubic blocks. An optimal policy requires 4 steps to finish this task. The default maximal number of steps per episode is 10.

House Building 3 In the House Building 3 environment (Figure 6d), there are two cubic blocks with a size of $3cm \times 3cm \times 3cm$, one cuboid block with a size of $12cm \times 3cm \times 3cm$, and a roof block with a bounding box size of around $12cm \times 3cm \times 3cm$. The blocks are randomly initialized in the workspace. The goal of this task is to first place the two cubic blocks next to each other, place the cuboid block on top of the two cubic blocks, then place the roof block on top of the cuboid block. An optimal policy requires 6 steps to finish this task. The default maximal number of steps per episode is 10.

House Building 4 In the House Building 4 environment (Figure 6e), there are four cubic blocks with a size of $3cm \times 3cm \times 3cm$, one cuboid block with a size of $12cm \times 3cm \times 3cm$, and a roof block with a bounding box size of around $12cm \times 3cm \times 3cm$. The blocks are randomly initialized in the workspace. The goal of this task is to first place two cubic blocks next to each other, place the cuboid block on top of the two cubic blocks, place another two cubic blocks on top of the cuboid block, then place the roof block on top of the two cubic blocks. An optimal policy requires 10 steps to finish this task. The default maximal number of steps per episode is 20.

Improvise House Building 2 In the Improvise House Building 2 environment (Figure 6f), there are two random blocks and a roof block. The shapes of the random blocks are sampled from Figure 12. The blocks are randomly initialized in the workspace. The goal of this task is to place the two random blocks next to each other, then place the roof block on top of the two random blocks. An optimal policy requires 4 steps to finish this task. The default maximal number of steps per episode is 10.

Fig. 12. The object set in the Improvise House Building 2 and Improvise House Building 3 environment.

Improvise House Building 3 In the Improvise House Building 3 environment (Figure 6g), there are two random blocks, a cuboid block, and a roof block. The shapes of the random blocks are sampled from Figure 12. The blocks are randomly initialized in the workspace. The goal of this task is to place the two random blocks next to each other, place the cuboid block on top of the two random blocks, then place the roof block on top of the cuboid block. An optimal policy requires 6 steps to finish this task. The default maximal number of steps per episode is 10.

Bin Packing In the Bin Packing task (Figure 6h), N objects and a bin are randomly placed in the workspace. The shapes of the objects are randomly sampled from Figure 13 (Object models are derived from [52]) with a maximum size of $8cm \times 4cm \times 4cm$ and a minimum size of $4cm \times 4cm \times 2cm$. The bin has a size of $17.6cm \times 14.4cm \times 8cm$. The goal of this task is to pack all N objects in the bin. An optimal policy requires $2N$ steps to finish the task. The number of objects N is configurable. By default, $N = 8$, and the maximal number of steps per episode is 20.

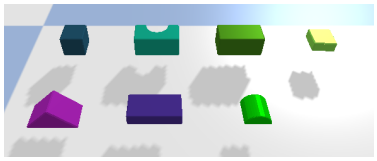


Fig. 13. The object set in the Bin Packing environment.

Bottle Arrangement In the Bottle Arrangement task (Figure 6i), six bottles with random shapes (sampled from 8 different shapes shown in Figure 14. The bottle shapes are generated from the 3DNet dataset [48]. The sizes of each bottle are around $5cm \times 5cm \times 14cm$), and a tray with a size of $24cm \times 16cm \times 5cm$ are randomly placed in the workspace. The goal is to arrange all six bottles in the tray. An optimal policy requires 12 steps to finish this task. By default, the maximal number of steps per episode is 20.

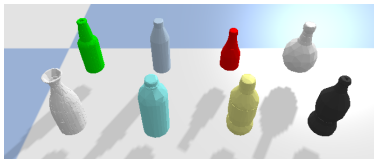


Fig. 14. The object set in the Bottle arrangement environment.

Box Palletizing In the Box Palletizing task (Figure 6j) (some object models are derived from [51]), a pallet with a size of $23.2cm \times 19.2cm \times 3cm$ is randomly placed in the

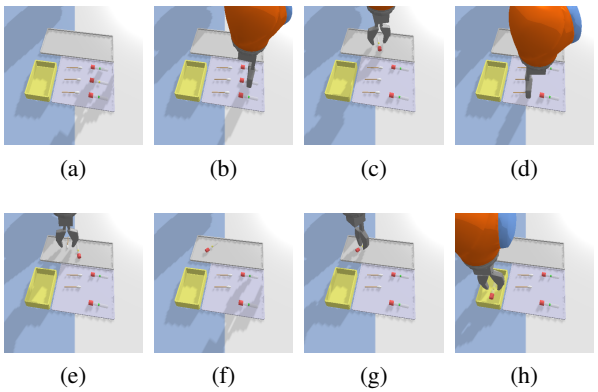


Fig. 15. An example of one COVID test process.

workspace. The goal is to stack N boxes with a size of $7.2cm \times 4.5cm \times 4.5cm$ as shown in Fig 6j. At the beginning of each episode and after the agent correctly places a box on the pallet, a new box will be randomly placed in the empty workspace. An optimal policy requires $2N$ steps to finish this task. The number of boxes N is configurable (6, 12, or 18). By default, $N = 18$, and the maximal number of steps per episode is 40.

Covid Test In the Covid Test task (Figure 6k), there is a new tube box (purple), a test area (gray), and a used tube box (yellow) placed arbitrarily in the workspace but adjacent to one another. Three swabs with a size of $7cm \times 1cm \times 1cm$ and three tubes with a size of $8cm \times 1.7cm \times 1.7cm$ are initialized in the new tube box. To supervise a COVID test, the robot needs to present a pair of a new swab and a new tube from the new tube box to the test area. The simulator simulates the user testing COVID by putting the swab into the tube and randomly placing the used tube in the test area. Then the robot needs to re-collect the used tube into the used tube box. See one example of this process in Figure 15. Each episode includes three rounds of COVID test. An optimal policy requires 18 steps to finish this task. By default, the maximal number of steps per episode is 30.

Object Grasping In the Object Grasping task (Figure 6l), the robot needs to grasp an object from a clutter of at most N objects. At the start of training, N random objects are initialized with random position and orientation. The shapes of the objects are randomly sampled from the object set shown in Figure 16. The object set contains 86 objects derived from the GraspNet1B [11] dataset. Every time the agent successfully grasps all N objects, the environment will re-generate N random objects with random positions and orientations. The maximal number of steps per episode is 1. The number of objects N in this environment is configurable. By default, there will be 15 objects.

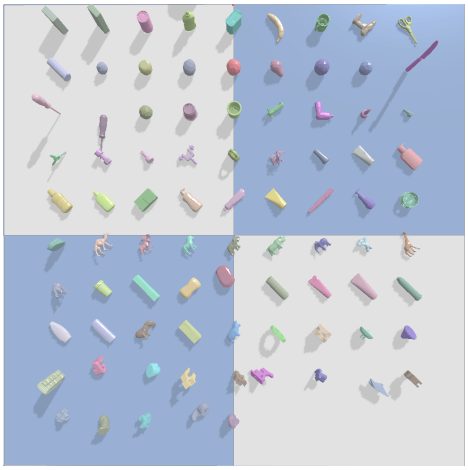


Fig. 16. The object set in the Object Grasping environment.

A.2 Close-Loop Environments

Block Reaching In the Block Reaching environment (Figure 9a), there is a cubic block with a size of $5cm \times 5cm \times 5cm$. The block is randomly initialized in the workspace. The goal of this task is to move the gripper towards the block such that the distance of the fingertip and the block is within $3cm$. By default, the maximal number of steps per episode is 50.

Block Picking In the Block Picking environment (Figure 9b), there is a cubic block with a size of $5cm \times 5cm \times 5cm$. The block is randomly initialized in the workspace. The goal of this task is to grasp the block and raise the gripper such that the gripper is $15cm$ above the ground. By default, the maximal number of steps per episode is 50.

Block Pushing In the Block Pushing environment (Figure 9c), there is a cubic block with a size of $5cm \times 5cm \times 5cm$ and a goal area with a size of $9cm \times 9cm$. The block and the goal area are randomly initialized in the workspace. The goal of this task is to push the block such that the distance between the block's center and the goal's center is within $5cm$. By default, the maximal number of steps per episode is 50.

Block Pulling In the Block Pulling environment (Figure 9d), there are two cuboid blocks with a size of $8cm \times 8cm \times 5cm$. The blocks are randomly initialized in the workspace. The goal of this task is to pull one of the two blocks such that it makes contact with another block. By default, the maximal number of steps per episode is 50.

Block in Bowl In the Block in Bowl environment (Figure 9e), there is a cubic block with a size of $5cm \times 5cm \times 5cm$, and a Bowl with a bounding box size of $16cm \times$

$16cm \times 7cm$. The block and the bowl are randomly initialized in the workspace. The goal of this task is to pick up the block and place it inside the bowl. By default, the maximal number of steps per episode is 50.

Block Stacking In the Block Stacking environment (Figure 9f), there are N cubic blocks with a size of $5cm \times 5cm \times 5cm$. The blocks are randomly initialized in the workspace. The goal of this task is to form a stack using the N blocks. By default, $N = 2$, the maximum number of steps per episode is 50.

House Building In the House Building environment (Figure 9g), there are $N-1$ cubic blocks with a size of $5cm \times 5cm \times 5cm$ and one triangle with a bounding box size of $5cm \times 5cm \times 5cm$. The blocks are randomly initialized in the workspace. The goal of this task is to first form a stack using the $N-1$ cubic blocks, then place the triangle block on top. By default, $N = 2$, the maximum number of steps per episode is 50.

Corner Picking In the Corner Picking environment (Figure 9h), there is a cubic block with a size of $5cm \times 5cm \times 5cm$ and a corner formed by two walls. The poses of the block and the corner are randomly initialized with a fixed relative pose between them so that the block is right next to the two walls. The wall is fixed in the workspace and not movable. The goal of this task is to nudge the block out from the corner and then pick it up at least $15cm$ above the ground. By default, the maximum number of steps per episode is 50.

Drawer Opening In the Drawer Opening environment (Figure 9i), there is a drawer with a random pose in the workspace. The outer part of the drawer is fixed and not movable. The goal of this task is to pull the drawer handle to open the drawer. By default, the maximum number of steps per episode is 50.

Object Grasping In the Object Grasping task (Figure 6l), the robot needs to grasp an object from a clutter of at most N objects. At the start of training, N random objects are initialized with random position and orientation. The shapes of the objects are randomly sampled from the object set shown in Figure 16. The object set contains 86 objects derived from the GraspNet1B [11] dataset. Every time the agent successfully grasps all N objects, the environment will re-generate N random objects with random positions and orientations. If an episode terminates with any remaining objects in the bin, the objects will not be re-initialized. The goal of this task is to grasp any object and lift it such that the gripper is at least $0.15m$ above the ground. The number of objects N in this environment is configurable. By default, there will be 5 objects, and the maximum number of steps per episode is 50.

B List of Configurable Environment Parameters

Table 4 shows a list of configuration parameters in our framework.

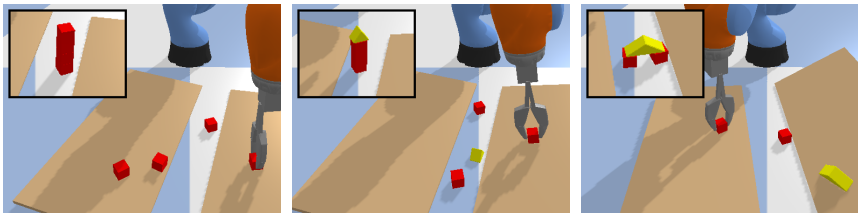
Parameter	Example	Description
robot	kuka	the robot to use in the experiment (Section 3.4).
action_sequence	pxyzr	The action space. 'pxyzr' means the action space a 5-vector, including the gripper action (p), the position of the gripper (x, y, z), and its top-down rotation (r).
workspace	array([[0.25, 0.65], [-0.2, 0.2], [0, 1]])	The workspace in terms of the range in x, y, and z.
object_scale_range	0.6	The scale of the size of the objects in the environment.
max_steps	10	The maximal steps per episode.
num_objects	1	The number of objects in the environment.
obs_size	128	The pixel size of the observation I.
in_hand_size	24	The pixel size of the in-hand image H.
fast_mode	True	If True, teleport the arm when possible to speed up the simulation.
render	False	If True, render the PyBullet GUI.
random_orientation	True	If True, the objects in the environments will be initialized with random orientations.
half_rotation	True	If True, constrain the gripper rotation between 0 and π.
workspace_check	point/bounding_box	Check object out of bound using the object center of mass or the bounding box
close_loop_tray	False	If True, generate a tray like in the Object Grasping (Figure 9j) in the close-loop environment.

Table 4. List of example configurable parameters in our framework.

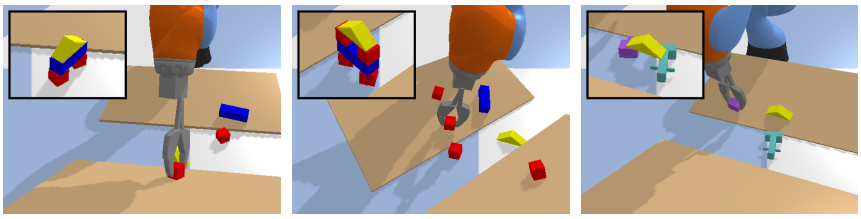
C Open-Loop 6DoF Environments

Most of the 6DoF environments mirror those in Figure 6, but the workspace is initialized with two ramps in the ramp environments or with a bumpy surface in the bump environments.

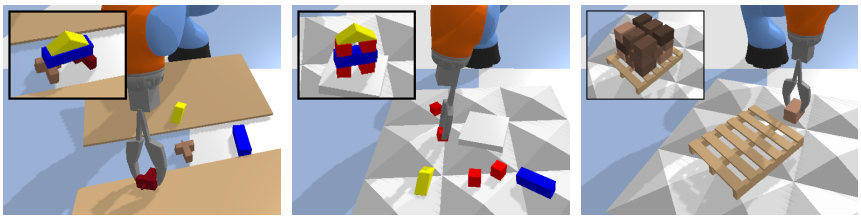
In the ramp environments (Figure 17a-Figure 17g), the two ramps are always parallel to each other. The distance between the ramps is randomly sampled between $4cm$ and $20cm$. The orientation of the two ramps along the z-axis is randomly sampled between 0 and 2π. The slope of each ramp is randomly sampled between 0 and $\frac{\pi}{6}$. The height of each ramp above the ground is randomly sampled between $0cm$ and $1cm$. In addition, the relevant objects are initialized with random positions and orientations either on the ramps or on the ground.



(a) Ramp Block Stacking (b) Ramp House Building 1 (c) Ramp House Building 2



(d) Ramp House Building 3 (e) Ramp House Building 4 (f) Ramp Improvise House Building 2



(g) Ramp Improvise House Building 3 (h) Bump House Building 4 (i) Bump Box Palletizing

Fig. 17. The open-loop 6DoF environments. The window on the top-left corner of each sub-figure shows the goal state of each task.

In the bump environments (Figure 17h and Figure 17i), bumpy surface is generated by nine pyramid shapes with a random slop sampled from 0 to $\frac{\pi}{12}$ degrees. The orientation of the bumpy surface along the z-axis is randomly sampled at the beginning of each episode.

D Additional Benchmarks

This section demonstrates the Open-Loop 2D Benchmark, the Open-Loop 6D Benchmark, and the Close-loop 3D Benchmark (Table 1).

D.1 Open-Loop 2D Benchmark

The Open-Loop 2D Benchmark requires the agent to solve the open-loop tasks in Figure 6 without random orientations, i.e., all of the objects in the environment will be

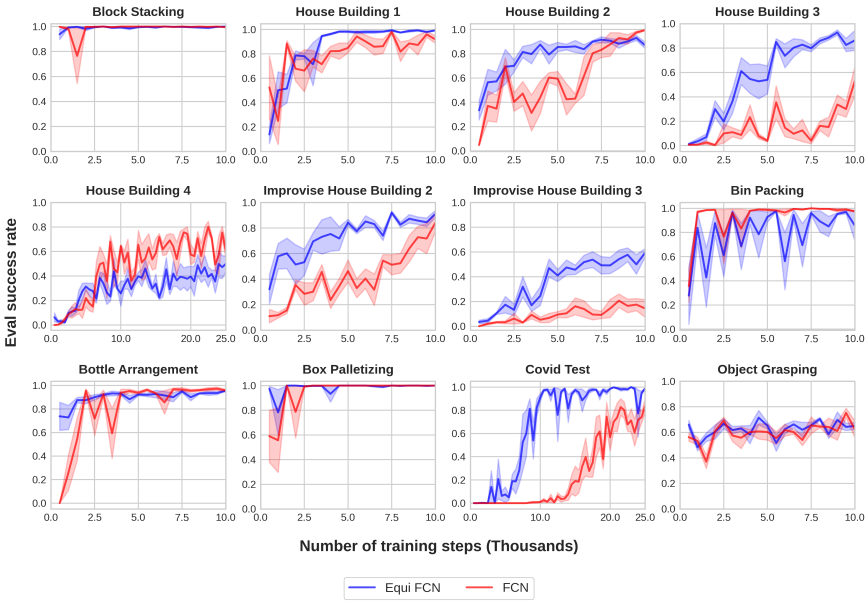


Fig. 18. The Open-Loop 2D benchmark result. The plots show the evaluation performance of the greedy policy in terms of the task success rate. The evaluation is performed every 500 training steps. Results are averaged over four runs. Shading denotes standard error.

initialized with a fixed orientation. The action space in this benchmark is $A_g \times A^{xy}$, i.e., the agent only controls the target (x, y) position of the gripper, while θ is fixed at 0 degree. Other environment parameters mirror the Open-Loop 3D Benchmark in Section 5.1.

Similar as in Section 5.1, we provide DQN, ADET, DQfD, and SDQfD algorithms with FCN and Equivariant FCN (Equi FCN) network architectures (the other architectures do not apply to this benchmark because the agent does not control θ). In this section, we show the performance of SDQfD equipped with FCN and Equivariant FCN. Figure 18 shows the result. Equivariant FCN (blue) generally shows a better performance compared with standard FCN (red).

D.2 Open-Loop 6D Benchmark

In the Open-Loop 6D Benchmark, the agent needs to solve the open-loop 6DoF environments (Appendix C) in an action space of $A_g \times A^{\mathrm{SE}(3)}$, i.e., the position (x, y, z) of the gripper and the rotation (θ, ϕ, ψ) of the gripper along the z, y, x axes.

We provide two baselines in this benchmark: 1) ASR [44]: a hierarchical approach that selects the actions in a sequence of $((x, y), \theta, z, \phi, \psi)$ using 5 networks; 2) Equivariant Deictic ASR [46] (Equi Deictic ASR): similar as 1), but replace the standard networks with equivariant networks and the deictic encoding to improve the sample

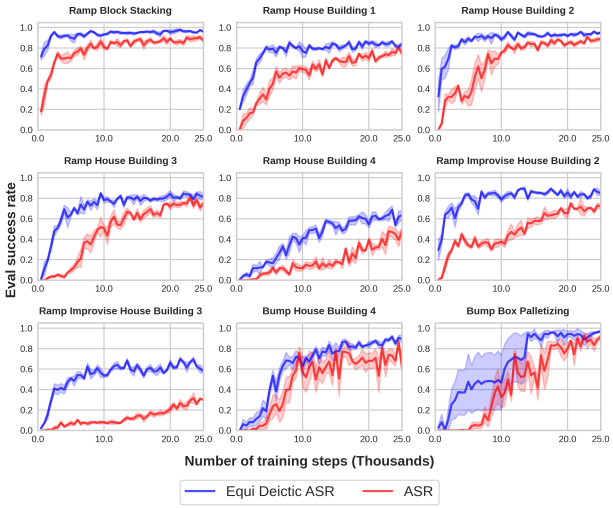


Fig. 19. The Open-Loop 6D benchmark result. The plots show the evaluation performance of the greedy policy in terms of the task success rate. The evaluation is performed every 500 training steps. Results are averaged over four runs. Shading denotes standard error.

efficiency. We use 1000 planner episodes for the ramp environments and 200 planner episodes for the bump environments. The in-hand image H in this experiment is a 3-channel orthographic projection image of a voxel grid generated from the point cloud at the previous pick pose. Other environment parameters mirror the Open-Loop 3D Benchmark in Section 5.1.

Figure 19 shows the results. Equivariant Deictic ASR (blue) demonstrates a stronger performance compared with standard ASR (red).

D.3 Close-Loop 3D Benchmark

The Close-Loop 3D Benchmark is similar as the Close-Loop 4D Benchmark (Section 5.2), but with the following two changes: first, the environments are initialized with a fixed orientation; second, the action space is $A_\lambda^{xyz} \in \mathbb{R}^4$ instead of $A_\lambda^{xyz\theta} \in \mathbb{R}^5$, i.e., the agent only controls the delta (x, y, z) position of the end-effector and the open-width λ of the gripper.

We provide the same baseline algorithms as in Section 5.2. In this section, we show the performance of SDQfD, Equivariant SDQfD (Equi SDQfD), and FERM SDQfD. Figure 20 shows the result. Equivariant SACfD (blue) shows the best performance across all tasks. FERM SACfD (green) and SACfD (red) has similar performance, except for Block Reaching, where FERM SACfD outperforms standard SACfD.

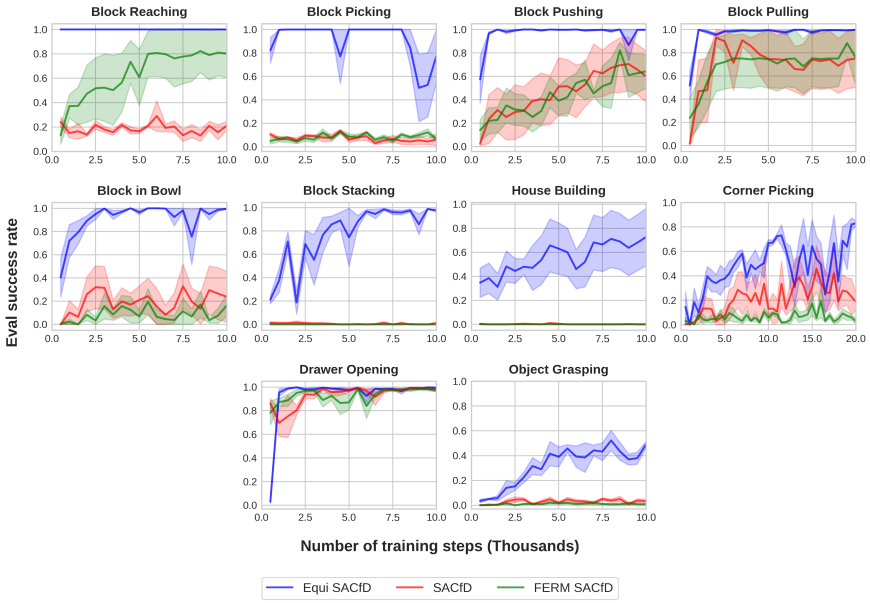


Fig. 20. The Close-Loop 3D benchmark result. The plots show the evaluation performance of the greedy policy in terms of the task success rate. The evaluation is performed every 500 training steps. Results are averaged over four runs. Shading denotes standard error.

E Additional Baselines for Open-Loop 3D Benchmark

In this section, we show the performance of three additional baseline algorithms in the Open-Loop 3D Benchmark (Section 5.1): DQfD, ADET, and DQN. We compare them with SDQfD (the algorithm used in Section 5.1). All algorithms are equipped with the Equivariant ASR architecture. Figure 21 shows the result. Notice that SDQfD and DQfD generally perform the best, while SDQfD has a marginal advantage compared with DQfD. ADET learns faster in some tasks (e.g., House Building 1), but normally converges to a lower performance compared with SDQfD and DQfD. DQN performs the worst across all environments because of the lack of imitation loss.

F Additional Baselines for Close-Loop 4D Benchmark

In this section, we show the performance of two additional baseline algorithms in the Close-Loop 4D Benchmark (Section 5.2): RAD SACfD and DrQ SACfD. As is shown in Figure 22, RAD SACfD (yellow) performs poorly in all 10 environments. DrQ SACfD (brown) outperforms FERM SACfD (purple) in Block Picking and Block Pulling, but still underperforms the equivariant methods (blue and red).

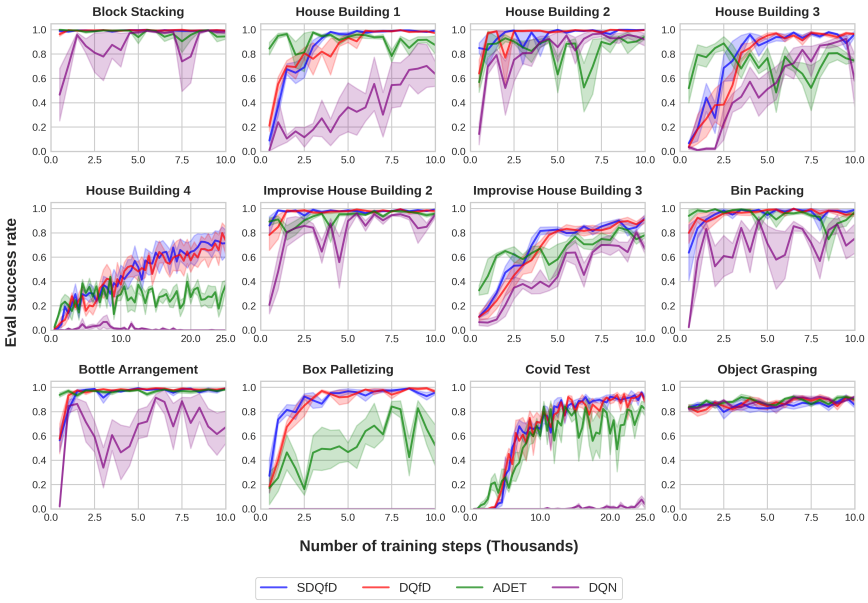


Fig. 21. The Open-Loop 3D benchmark result with additional baselines. The plots show the evaluation performance of the greedy policy in terms of the task success rate. The evaluation is performed every 500 training steps. Results are averaged over four runs. Shading denotes standard error.

G Benchmark Details

G.1 Open-Loop Benchmark

In all environments, the kuka arm is used as the manipulator. The workspace has a size of $0.4m \times 0.4m$. The top-down observation I covers the workspace with a size of 128×128 pixels. (In the Rot FCN baseline, I's size is 90×90 pixels, and is padded with 0 to 128×128 pixels. This is padding required for the Rot FCN baseline because it needs to rotate the image to encode θ.) The size of the in-hand image H is 24×24 pixels for the Open-Loop 2D and Open-Loop 3D benchmarks. In the Open-Loop 6D Benchmark, H is a 3-channel orthographic projection image, with a shape of $3 \times 24 \times 24$ in the ramp environments, and $3 \times 40 \times 40$ in the bump environments. We train our models using PyTorch [33] with the Adam optimizer [23] with a learning rate of 10^{-4} and weight decay of 10^{-5}. We use Huber loss [17] for the TD loss. The discount factor γ is 0.95. The mini-batch size is 16. The replay buffer has a size of 100,000 transitions. At each training step, the replay buffer will separately draw half of the samples from the expert data and half of the samples from the online transitions. The weight w for the margin loss term of SDQfD is 0.1, and the margin $l = 0.1$. We use the greedy policy as the behavior policy. We use 5 environments running in parallel.

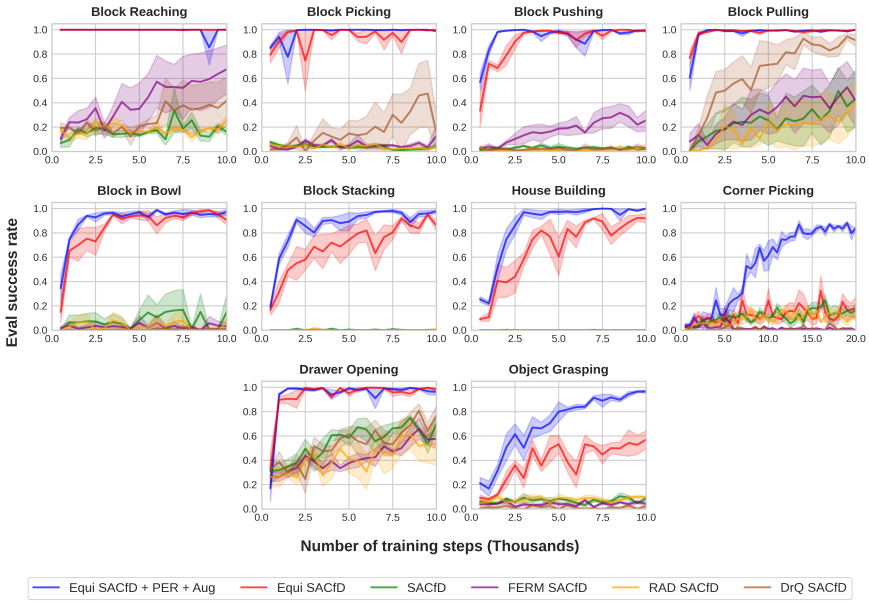


Fig. 22. The Close-Loop 4D benchmark result with additional baselines. The plots show the evaluation performance of the greedy policy in terms of the task success rate. The evaluation is performed every 500 training steps. Results are averaged over four runs. Shading denotes standard error.

G.2 Close-Loop Benchmark

In all environments, the kuka arm is used as the manipulator. The workspace has a size of $0.3m \times 0.3m \times 0.24m$. The pixel size of the top-down depth image O is 128×128 (except for the FERM baseline, where I's size is 142×142 and will be cropped to 128×128). I's FOV is $0.45m \times 0.45m$. We use the Adam optimizer with a learning rate of 10^{-3}. The entropy temperature α is initialized at 10^{-2}. The target entropy is -5. The discount factor $\gamma = 0.99$. The batch size is 64. The buffer has a capacity of 100,000 transitions. In baselines using the prioritized replay buffer (PER), PER has a prioritized replay exponent of 0.6 and prioritized importance sampling exponent $\beta_0 = 0.4$ as in [36]. The expert transitions are given a priority bonus of $\epsilon_d = 1$. The FERM baseline's contrastive encoder is pretrained for 1.6k steps using the expert data as in [54]. We use 5 environments running in parallel.