

# Appendix

## Table of Contents

---

<b>A</b>	<b>Limitation and Future Work</b>	<b>14</b>
<b>B</b>	<b>Clarification on EMBR’s Relationship with Hierarchical RL</b>	<b>15</b>
<b>C</b>	<b>Experimental Details</b>	<b>15</b>
C.1	Robot Observation and Action Space . . . . .	15
C.2	Environment Reset . . . . .	15
C.3	Skills, Symbolic Actions, Predicates, Pre- and Post-conditions . . . . .	16
C.4	Experimental Evaluation Protocol . . . . .	16
C.5	Explanation of Prior Method Comparison . . . . .	16
C.6	Short-horizon Skill Descriptions . . . . .	17
C.7	Long-horizon Task Descriptions . . . . .	18
C.8	Explanation of Ablation Experiments . . . . .	19
<b>D</b>	<b>Additional Ablations: EMBR vs. Original BEE</b>	<b>19</b>
<b>E</b>	<b>Additional Ablations: EMBR with Stage-Specific Rewards vs. EMBR with Skill-Specific Rewards Trained from Scratch</b>	<b>19</b>
<b>F</b>	<b>Additional Ablations: Task Performance With vs. Without Any of the Five Cameras</b>	<b>19</b>
F.1	Excluding Wrist Camera . . . . .	20
F.2	Excluding Left-Shoulder Camera . . . . .	20
F.3	Excluding Left-Shoulder Camera . . . . .	20
F.4	Excluding Left-Waist Camera . . . . .	20
F.5	Excluding Right-Waist Camera . . . . .	21
<b>G</b>	<b>Amount of Human Supervision and Robot Training Required</b>	<b>21</b>
<b>H</b>	<b>Hyperparameters</b>	<b>21</b>

---

### A Limitation and Future Work

While EMBR significantly improves upon prior methods, there are a number of important limitations that we hope future work can address.

**Task scope.** The current system does not learn skills that pay attention to object semantics. For example, the ‘grasp object from desk’ skill involve grasping any object from the desk, rather than grasping a particular object of interest. Therefore, the current system is limited in the scope of long-horizon tasks that it can perform. Training skills that are semantic in nature is in principle possible with EMBR; doing so may require learning more powerful success classifiers for certain visuomotor skills, as they would need to pay attention to both spatial and semantic information.

**Amount of human supervision.** While the presented system was shown to complete 14 skills and 3 challenging long-horizon tasks, EMBR completes tasks with a relatively large amount of human guidance: a human needs to determine the set of primitive skills, define the symbolic relationships, determine the stages for each primitive skill, and provide positive examples for each stage of each skill. Reducing the amount of human guidance without sacrificing on the complexity of skills that can be learned is an interesting avenue for future work.

**Environment generalization.** While EMBR generalizes to new objects, it does not in its current form provide skills that generalize to new scenes or desks. This would likely require collecting

data with greater variety in scenes and desks. Scaling data towards such generalization is also an interesting future direction.

**Planning.** While the technical focus of this work was to develop an RL algorithm that provides robust and grounded skills and evaluate how such skills can be sequenced, the symbolic planner employed in our system also has limitations. In particular, the symbolic planner we used does not handle partial observability. For example, the planner cannot tell whether an object is in the left drawer without opening the left drawer in the first place. We hope that future work can develop more sophisticated grounding mechanisms to handle such partial observability.

## B Clarification on EMBR’s Relationship with Hierarchical RL

EMBR is not a hierarchical RL algorithm because it does not learn a model over options, especially since in our case, the “model over options” is the off-the-shelf symbolic planner itself (i.e., the symbolic planner autonomously chooses which option to execute). Instead, we are learning a skill neural network for each of the 14 skills. These skills or options are not defined by the expert, but rather learned autonomously by using images of the terminal states as the only supervision signal. We recognize that different parts of the robotics and reinforcement learning community have different interpretations of the term “model-based reinforcement learning”. In this paper, “model-based RL” only refers to learning each option efficiently by learning a visual dynamics model, rather than learning a high-level model over options.

## C Experimental Details

### C.1 Robot Observation and Action Space

**Observation Space.** In all robotic experiments in this work, the robot is equipped with five  $64 \times 64$  RGB cameras and no other observations. In practice, the robot’s observation space has four out of five cameras:  $\mathcal{S} = \mathbb{R}^{4 \times 64 \times 64 \times 3}$ .

**Action Space.** The robot’s action space is a five-dimensional vector that specifies the translation and rotation of the Panda robot’s end-effector as well as its gripper:  $\mathcal{A} = \mathbb{R}^5$ . The first three scalars in this vector denotes the  $x, y, z$  translation of the end-effector in centimeters. The fourth scalar in this vector denotes the wrist rotation of the end-effector in radians. Finally, the five, last scalar in this vector is a binary scalar that commands the robot’s gripper: 0 for fully opening the gripper, and 1 for fully closing the gripper.

### C.2 Environment Reset

Different from simulated learning, real-world robotic learning requires environment reset. In our experiments, we use the following reset strategies to reset each skill during data collection.

**Opening drawers.** For opening the left or right drawer, once the drawer is successfully opened, a forward pushing cartesian motion to re-close the drawer is sufficient to reset the environment.

**Closing drawers.** For closing the left or right drawer, once the drawer is successfully closed, a backward pulling cartesian motion to re-open the drawer is sufficient to reset the environment.

**Opening cabinet.** For opening the cabinet, once the drawer is successfully opened, a forward pushing cartesian motion to re-close the cabinet is sufficient to reset the environment.

**Closing cabinet.** For closing the cabinet, we use the open-cabinet skill to re-open the cabinet that has just been closed. Therefore, this skill should be learned after the “Open cabinet” skill is learned.

**Inserting marker.** For inserting a marker, once the marker is successfully inserted, an upward cartesian motion to pull-out a marker is sufficient to reset the environment.

**Grasping object.** For grasping object skills, we throw a clutter of many training objects onto the desk or into the drawers. Once the grasp is successful, re-opening the closed gripper will be sufficient to reset the environment. In the case where the objects on the desk fall off the desk, a human picks them up once every few hours.

**Placing object.** For placing an object, re-closing the gripper and an upward cartesian motion to lift the object that has just been placed to the desk, drawers or cabinet is sufficient to reset the environment.

Table 3: Types and entities

	Types	Entities
1	robot	robot
2	location	desk, left-drawer, right-drawer, cabinet
3	objects	object, marker

Table 4: Predicates

	Predicates	Description
1	(grasped robot)	robot’s gripper is closed
2	(at object location)	object is at location
3	(opened location)	location is opened
4	(in object robot)	object is in robot’s gripper
5	(inserted object)	object is inserted into the marker holder

### C.3 Skills, Symbolic Actions, Predicates, Pre- and Post-conditions

Types and entities defined symbolically are in Table 3. Predicates that build on these types are in Table 4. The mapping between the 14 learned primitive skills and their corresponding symbolic actions are in Table 5.

### C.4 Experimental Evaluation Protocol

#### C.4.1 Short-horizon Skill Evaluation

For evaluation of primitive skills, all methods under comparison are given 100 timesteps to complete each skill. Inability to complete the skill under this timed budget denotes a failure and otherwise a success. The VAE and the latent dynamics model architecture we used is the GHVAE architecture [51]. The success classifiers and Q-functions are five-layer deep neural networks.

#### C.4.2 Long-horizon Task Evaluation

At the beginning of each long-horizon task, all drawers and cabinet are closed. It is also required that all drawers and cabinets be closed at the end of each task to achieve task success. During evaluation, all methods are given  $100 \times K^*$  timesteps to complete each long-horizon task, where  $K^*$  is the minimum number of skills required to complete the task, as elaborated in Table 2. In effect, this is giving the robot 100 timesteps per skill that needs to be used in the task to complete the entire task. Inability to complete the skill under this timed budget denotes a task failure and otherwise a success.

### C.5 Explanation of Prior Method Comparison

#### C.5.1 Prior method 2: “EMBR w/o $f_{\text{vae}}, f_{\mathcal{T}}$ [20]”

We note that while we name the second prior method “EMBR w/o  $f_{\text{vae}}, f_{\mathcal{T}}$  [20]” by citing Qt-Opt [20], this prior method is similar to Qt-Opt except in the following ways:

1. Qt-Opt uses high-resolution images  $472 \times 472$  as image observation while EMBR use  $64 \times 64$  images
2. Qt-Opt uses distributed machine learning mechanisms while “EMBR w/o  $f_{\text{vae}}, f_{\mathcal{T}}$  [20]” does not use distributed mechanisms
3. “EMBR w/o  $f_{\text{vae}}, f_{\mathcal{T}}$  [20]” uses only positive example images and does not use negative example images
4. The original Qt-Opt algorithm [20] uses a specialized CNN architecture, while “EMBR w/o  $f_{\text{vae}}, f_{\mathcal{T}}$  [20]” uses the “ResNet152” architecture in both the success classifier and the Q-function, which is the largest CNN we can fit into our computational resources.

#### C.5.2 Prior method 3: “BEE [56]”

We also note that for the third prior method “BEE [56]”, we use the same rollout timesteps of  $L = 5$  as in EMBR for fair comparison.

Table 5: Mapping between primitive skills and symbolic actions

$k$	Primitive Skill	Corresponding Symbolic Action
1	Insert marker into a marker holder	insert(marker robot)
2	Grasp object from left drawer	grasp(object left-drawer robot)
3	Grasp object from right drawer	grasp(object right-drawer robot)
4	Grasp object from desk	grasp(object desk robot)
5	Open right drawer	open(right-drawer robot)
6	Open left drawer	open(left-drawer robot)
7	Open cabinet	open(cabinet robot)
8	Close right drawer	close(right-drawer robot)
9	Close left drawer	close(left-drawer robot)
10	Close cabinet door	close(cabinet robot)
11	Place object into left drawer	place(object left-drawer robot)
12	Place object into right drawer	place(object right-drawer robot)
13	Place object on desk	place(object desk robot)
14	Place object in cabinet	place(object cabinet robot)

## C.6 Short-horizon Skill Descriptions

**Insert a marker into a marker holder.** In this skill, the robot needs to insert a marker of unseen texture or geometry (while the exact geometry is unseen and novel, markers in general are quite geometrically similar) into a marker holder. At the beginning of the skill, the marker is assumed to be in the robot’s gripper and oriented vertically. The skill is considered a success when the marker is successfully inserted into the marker holder and a failure otherwise.

Before the insert-marker skill is executed, we also equip the robot with an additional scripted skill of changing the orientation of the marker from horizontal to vertical. This skill uses the edge of the desk as a supporting point to turn a marker of any texture (since it is scripted) from horizontal to vertical. While one can also acquire this skill using EMBR, we found that the scripted version of the skill performs best primarily due to the high force-torque intensity of this skill.

**Grasp object from left drawer (single).** In this skill, the robot needs to pick-up a single object of novel geometry and/or texture from the left drawer. At the beginning of each trial, the object is already in the left drawer. The initial position is randomized but not at the corners of the left drawer, in which case grasping is not mechanically possible using the Franka Emika Panda parallel-jaw gripper. The initial orientation of each object is also randomized. The skill is considered a success when the object is successfully picked up and a failure otherwise.

**Grasp object from left drawer (clutter).** In this skill, the robot needs to pick-up a single object from a three-object clutter, each with novel geometry and/or texture, in the left drawer. At the beginning of each trial, three novel objects are already in the left drawer. The initial position of each object is randomized but not at the corners of the left drawer, in which case grasping is not mechanically possible using the Franka Emika Panda parallel-jaw gripper. The initial orientation of each object is also randomized. The skill is considered a success when any one object is successfully picked up and a failure otherwise.

**Grasp object from right drawer (single).** In this skill, the robot needs to pick-up a single object of novel geometry and/or texture from the right drawer. At the beginning of each trial, the object is already in the right drawer. The initial position is randomized but not at the corners of the right drawer, in which case grasping is not mechanically possible using the Franka Emika Panda parallel-jaw gripper. The initial orientation of each object is also randomized. The skill is considered a success when the object is successfully picked up and a failure otherwise.

**Grasp object from right drawer (clutter).** In this skill, the robot needs to pick-up a single object from a three-object clutter, each with novel geometry and/or texture, in the right drawer. At the beginning of each trial, three novel objects are already in the right drawer. The initial position of each object is randomized but not at the corners of the right drawer, in which case grasping is not mechanically possible using the Franka Emika Panda parallel-jaw gripper. The initial orientation of each object is also randomized. The skill is considered a success when any one object is successfully picked up and a failure otherwise.

**Grasp object from desk (single).** In this skill, the robot needs to pick-up a single object of novel geometry and/or texture from the desk. At the beginning of each trial, the object is already on the

desk. The initial position and orientation of the object is randomized. The skill is considered a success when the object is successfully picked up and a failure otherwise.

**Grasp object from desk (clutter).** In this skill, the robot needs to pick-up a single object from a three-object clutter, each with novel geometry and/or texture, from the desk. At the beginning of each trial, the object is already on the desk. The initial position and orientation of the object is randomized. The skill is considered a success when the object is successfully picked up and a failure otherwise. If any of the three objects on the desk is pushed off the desk, the episode is also considered a failure.

**Place an object into left drawer.** In this skill, the robot needs to place a single object of novel geometry and/or texture into the left drawer. In the beginning of each trial, the object is already in the gripper and the left drawer is already open. The skill is considered a success when the object is successfully placed into the left drawer and a failure otherwise.

**Place an object into right drawer.** In this skill, the robot needs to place a single object of novel geometry and/or texture into the right drawer. In the beginning of each trial, the object is already in the gripper and the right drawer is already open. The skill is considered a success when the object is successfully placed into the right drawer and a failure otherwise.

**Place an object on desk.** In this skill, the robot needs to place a single object of novel geometry and/or texture on the desk. In the beginning of each trial, the object is already in the gripper. The skill is considered a success when the object is successfully placed on the desk and a failure otherwise.

**Place an object in cabinet.** In this skill, the robot needs to place a single object of novel geometry and/or texture into the cabinet. In the beginning of each trial, the object is already in the gripper and the cabinet is already open. The skill is considered a success when the object is successfully placed in the cabinet and a failure otherwise.

## C.7 Long-horizon Task Descriptions

**Organize Desk and Cabinet.** In this task, the robot needs to pick-up all three novel objects (unseen during training) cluttered on the desk in any order and place them in the cabinet. After placing, the robot is also required to close all drawers and cabinet in the end. In this task, there will always be three cluttered objects on the desk unseen during training, and the position and orientation of each object is non-deterministic. All three cluttered objects have novel geometries and/or textures that are unseen during training. These novel objects were sampled randomly from Fig. 5 (d). While one can test the algorithm across a flexible number of objects at the desk, we fix the number of objects per experiment to three, considering that the overall task performance depends on the total number of skills required to complete the entire task. Some of these objects are also semi-transparent, such as the scotch tape. The task is only considered successful when all three objects are successfully placed into the cabinet and the cabinet and the drawers are closed.

**Organize Markers.** In this task, the robot needs to pick-up three unseen markers (unseen texture, seen geometry) and insert it into a marker holder in the left drawer. All three markers have novel textures that are unseen during training. The marker holder is not mounted to the drawer and can move freely in the left drawer. Each marker can either be on the desk or in the right drawer. Although we could have also placed some markers in the left drawer to make the task more difficult, a marker in the left drawer is not mechanically graspable by the Frank Emika robot gripper once the marker holder is placed in the same drawer (the left drawer will not be wide enough for the gripper to grasp anything). For example, at the beginning of a trial, two markers can be randomly placed on the desk (random position and orientation), and one marker can be randomly placed in the right drawer (random position and orientation).

To re-orient each marker, we also equip all methods with a scripted skill that uses the edge of the desk as a supporting point to re-orient the marker in the gripper. This skill is executed right before executing the "Insert marker into a marker holder" skill. We emphasize that only this one skill is scripted in the robot's skill repertoire. All other 14 skills in the robot's skill repertoire is both visuomotor and learned.

**Rearrange Objects.** In this task, the robot needs to rearrange three objects to their respective desired locations. All three objects have novel geometries and/or textures that are unseen during training. Initially, each of the three novel objects will be placed on the desk (object1), in the left drawer (object2), and in the right drawer (object3) respectively. We evenly place these objects to

Table 6: Ablation 4: EMBR without Stages (EMBR Data) vs. EMBR without Stages (Recollect Data). Successful trials (out of 20) and success rates of visuomotor primitive skills.

$K$	Primitive Skill	Precision	EMBR	EMBR w/o Stages (EMBR Data)	EMBR w/o Stages (Recollect Data)
1	Insert marker into a marker holder	High	20 (100%)	20 (100%)	20 (100%)
5	Open right drawer	Medium	20 (100%)	17 (85%)	0 (0%)
6	Open left drawer	Medium	20 (100%)	16 (80%)	0 (0%)
7	Open cabinet	Medium	20 (100%)	18 (90%)	0 (0%)
11	Place object into left drawer	Low	20 (100%)	20 (100%)	20 (100%)
12	Place object into right drawer	Low	20 (100%)	20 (100%)	20 (100%)
13	Place object on desk	Low	20 (100%)	20 (100%)	20 (100%)
14	Place object in cabinet	Low	20 (100%)	20 (100%)	20 (100%)

maximize the number of unique visuomotor skills the robot needs to use in order to complete the task. In other words, there will always be one object on the desk, one object in the left drawer, and one object in the right drawer, so that the robot needs to use all three visuomotor skills to complete the entire long-horizon task: "Grasp object from left drawer", "Grasp object from right drawer", and "Grasp object from desk". However, the identity of each novel object is random, so are their positions and orientations.

### C.8 Explanation of Ablation Experiments

**Ablation 1: Stage-specific vs. skill-specific reward function** In this ablation, we re-train EMBR’s Q-functions using just one stage per skill using the same dataset  $\mathcal{D}$  for fair comparison. Therefore, for skills that only have one stage, the success rates would be the same for using stage-specific and skill-specific reward functions.

**Ablation 2: Task performance with vs. without classifier-based replanning.** In this ablation, we disable classifier-based replanning and thus didn’t need to re-train any of EMBR’s neural networks, i.e. the VAE, the dynamics model, the success classifiers, and the Q-functions. Concretely, we execute each skill for 100 timesteps without early termination and move on to the next skill without success detection. Notice that in this case, the success classifiers are still used for training the Q-function; they are just not used for replanning during long-horizon task execution.

## D Additional Ablations: EMBR vs. Original BEE

The only changes to the BEE comparison are to give it the same architecture as EMBR, specifically to make the comparison of algorithms more fair. We compared to the original BEE algorithm using the code from <https://github.com/stanford-iris-lab/batch-exploration>. For all grasping skills, we observe that the original BEE achieves 0% success rate. This shouldn’t be surprising at all: the VAE in BEE is much smaller than the version of BEE we compared to in our pre-revision paper. The VAE learned from the original BEE algorithm cannot reconstruct any details of graspable objects, which makes the original BEE perform much worse than either our re-implemented version of BEE in our pre-revision paper, or the EMBR algorithm.

## E Additional Ablations: EMBR with Stage-Specific Rewards vs. EMBR with Skill-Specific Rewards Trained from Scratch

In addition to comparing to EMBR w/o Stages (EMBR Data) (i.e. EMBR without Stage-specific Rewards but using data already collected from the original EMBR algorithm), we also compared to EMBR without Stages (Recollect Data) (i.e. an additional ablated version of "EMBR without Stage-specific Rewards" that collects data from scratch). In Table 6, we observe that "EMBR w/o Stages (Recollect Data)" cannot explore medium-precision skills that have multiple stages. This includes all cabinet and drawer-opening skills. For skills that only have one stage, the performance is the same as "EMBR w/o Stages (EMBR data)".

## F Additional Ablations: Task Performance With vs. Without Any of the Five Cameras

In our experiments, the robot has access to RGB images from five cameras, one of which is mounted on the wrist of the robot. To measure the contribution of each camera to the observed performance, we re-trained EMBR’s success classifier and Q-function using only four of the five cameras, excluding the wrist, left-shoulder, right-shoulder, left-waist or the right-waist camera. Note that in all ablations, we exclude the particular camera observation from the robot’s observation space and then re-train EMBR, using the same dataset  $\mathcal{D}$ , for fair comparison. In Table 7, we found that each of the

Table 7: Ablation 3: Skill performance with vs. without any of the five cameras. Successful trials (out of 20) and success rates of visuomotor primitive skills.

$K$	Primitive Skill	EMBR With Wrist Camera	EMBR Without Wrist Camera
2	Grasp object from left drawer (single)	18 (90%)	14 (70%)
	Grasp object from left drawer (clutter)	18 (90%)	15 (75%)
3	Grasp object from right drawer (single)	18 (90%)	12 (60%)
	Grasp object from right drawer (clutter)	18 (90%)	14 (70%)
4	Grasp object from desk (single)	18 (90%)	12 (60%)
	Grasp object from desk (clutter)	18 (90%)	13 (65%)
$K$	Primitive Skill	EMBR With Left Shoulder Camera	EMBR Without Left Shoulder Camera
7	Open Cabinet	20 (100%)	3 (15%)
$K$	Primitive Skill	EMBR With Right Shoulder Camera	EMBR Without Right Shoulder Camera
5	Open Right Drawer	20 (100%)	11 (55%)
6	Open Left Drawer	20 (100%)	13 (65%)
$K$	Primitive Skill	EMBR With Left Waist Camera	EMBR Without Left Waist Camera
2	Grasp object from left drawer (single)	18 (90%)	11 (55%)
$K$	Primitive Skill	EMBR With Right Waist Camera	EMBR Without Right Waist Camera
3	Grasp object from right drawer (single)	18 (90%)	6 (30%)

five cameras is important to the overall performance in that it improves the success rate of specific skills by 15-85%.

The reliance on a large set of perception systems is mainly the result of the narrow field of view (70 degrees) of each of the five cameras, such that:

1. Only the left shoulder camera can see the entire cabinet
2. Only the right shoulder camera can see the entire left and right drawers when the robot is pulling the drawer handle simultaneously - this is especially important during our long-horizon task experiments since the robot needs to know whether both drawers are open or closed.
3. Only the left waist camera can clearly see the entire marker holder for “insert marker” skills and all individual objects in the left drawer
4. Only the right waist camera can clearly see all individual objects in the right drawer

The reason why as many as five cameras are needed is because there are 14 different skills with varying degrees of precision (from inserting a marker to opening a cabinet on the other end of the desk). Learning to do all of 14 skills requires a global view of the scene. The EMBR algorithm is compatible with an arbitrary number of cameras, and in principle, a single camera with a sufficient field-of-view would work as well. Using five narrow-field-of-view cameras in our particular office desk setup doesn’t affect the applicability, transfer, and replicability of the EMBR framework, because EMBR is fully flexible and compatible with an arbitrary number of cameras, and one can add or remove cameras as they wish.

### F.1 Excluding Wrist Camera

In Table 7, we find that including the wrist camera observation leads to a 15-30% improvement in grasping skill success rate, highlighting the importance of the wrist camera observation.

### F.2 Excluding Left-Shoulder Camera

In Table 7, we find that including the left-shoulder camera observation leads to a 85% improvement in the success rate of opening the cabinet on the office desk, because the left-shoulder camera is the only camera that can fully observe the cabinet.

### F.3 Excluding Right-Shoulder Camera

In Table 7, we find that including the right-shoulder camera observation leads to a 35-45% improvement in the success rate of opening the drawers of the office desk, because the right-shoulder camera is the only camera that can see the entire left and right drawer when the robot is pulling the drawer handle.

### F.4 Excluding Left-Waist Camera

In Table 7, we find that including the left-waist camera observation leads to a 35% improvement in the success rate of grasping an object from the left drawer, because only the left-waist camera can clearly see individual objects in the left drawer.

### F.5 Excluding Right-Waist Camera

In Table 7, we find that including the right-waist camera observation leads to a 60% improvement in the success rate of grasping an object from the right drawer, because only the right-waist camera can clearly see individual objects in the right drawer.

## G Amount of Human Supervision and Robot Training Required

For each stage of a skill, providing a set of example images typically takes 15 minutes. Providing a set of example images for a skill takes  $15 \times N_k$  minutes, where  $N_k$  is the total number of stages for the skill. The amount of robot training time is 16 hours on average per skill, regardless of how many stages the skill has. The number of trajectories per skill is 300-400 trajectories on average.

## H Hyperparameters

Table 8 details the hyperparameters for EMBR.

Table 8: Hyperparameters		
Category	Hyper-parameter	Value
Train Only	Episode Length / Horizon $H$	50
Test Only	# rollout trajectories $m_0$	150
	Episode Length / Horizon $H$	100
Train and Test	Discount Rate $\gamma$	0.98
	Rollout Horizon $L$	5
	Number of action samples during Q-maximization $m_1$	200
	Number of Camera Observations	4