Appendix

A. Stage 1: Details

In simulation, during the collect phase, we obtain 50 expert policies per task, corresponding to different layouts, so a total of 900 task and layout specific policies, which can be replayed in stage 2. For the real robot environment, we collect 8 trajectories per task through kinesthetic demonstration, so a total of 40 expert trajectories, which can be replayed.

B. Stage 2: Details

For augmenting the real-robot kinesthetic demos collected by experts, we replay the trajectories while varying different attributes of the scene, and recording per-timestep image observations during the replays. The visual augmentations in the real-robot setting correspond to color jitters of the observation images. In addition, we incorporate three different semantic augmentations. The first is action noise during replays to ensure wider coverage in mitigating covariate shift issues. Second, we manually shuffle the positions of distractor objects across the scene, and swap some objects in and out of the scene. Finally, we develop a novel method for incorporating automatic semantic scene variations, *without physically modifying objects* in the scene. We use latest advances in generative modeling [2], [38] that lets us perform controlled scene re-generations. This is at the dataset level, and doesn't require additional robot operation hours. We specifically consider the open-sourced Stable Diffusion trained model [38], and run inference through it. The model takes as input an image of the scene, and a region for modification, specified in pixel coordinates. Controlled generation lets us keep the rest of the scene unchanged, and introduce new plausible objects in the region specified. By automating this process, we can obtain several visually augmented demos with zero extra human effort for data collection. Fig. 4 shows a visualization of what controlled generation looks like for a scene from our real robot environment. The generated images place plausible objects like mugs, cups, and glasses on locations of the white-colored table that are unoccupied.

C. Stage 3: Details

The pre-trained visual representations for R3M are obtained through training on egocentric human videos [48], with a combination of time-contrastive loss, and losses for video-language alignment. We use the exact pre-trained model from the original paper, and do not introduce any additional loss for fine-tuning with our own collected data. Fine-tuning simply corresponds to backpropagating through the layers of the pre-trained encoder to update its weights, while performing imitation learning in stage 4.

D. Stage 4: Details

For visual goals, the embeddings obtained from stage 3 are 1024x1 dimensional, and are concatnetaed with the observation embedding, which is also of the same dimensions, is concatenated, before feeding the concatenated vector to the policy MLP. In additon, we also concatenate the roobt joint velocity, and joint pose vectors (each of dimension 8x1), so the combined embedding that goes as input to the policy MLP is of dimension 2064x1. The output of the policy MLP is a mean and standard deviation vector, such that they represent a Gaussian action distribution of 8x1 dimension.

E. Experiment setup details

*1) **Simulation environment**:* Each episode is evaluated for a horizon length of 100, and success criteria is determined by checking whether the final pose of the target object is within a 5% error bound from the specified goal-pose during evaluation.

*2) **Real-robot environment**:* Each episode is evaluated for a horizon length of 100 time-steps. At the beginning of each evaluation episode, a goal image is first collected by manually setting the target object to a fixed goal location with organic variations; then, the target object is set back and the agent takes in both the captured goal image and current visual observations as input. We define an episode as success when the robot is able to move the target object to within a range of 3cm error from the given goal location.

F. Additional results

Fig. 8 shows detailed results for the **CACTI -Sim-50 Benchmark** that was forward referenced, with aggregate values in Fig. 6 of the main paper.

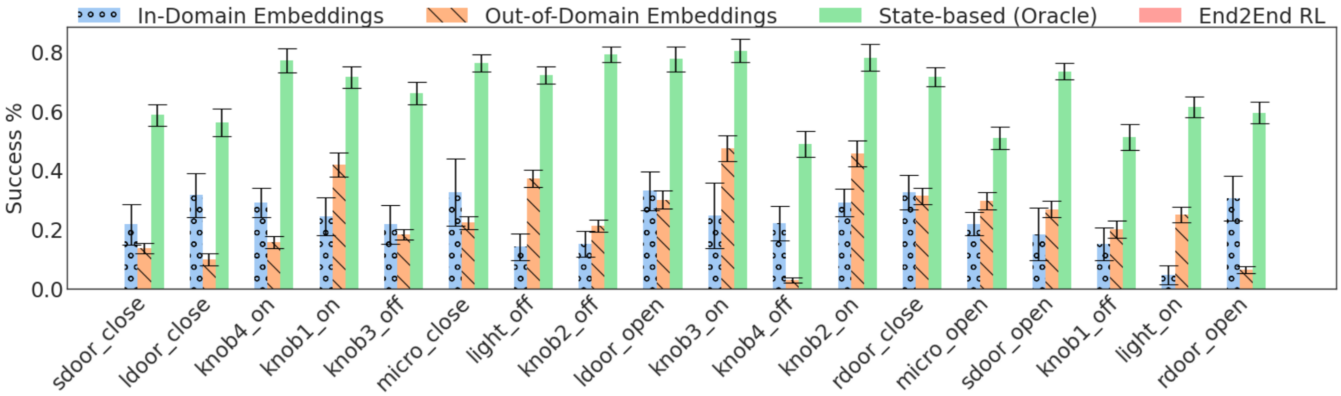

Fig. 8. **CACTI -Sim-50 Benchmark results.** The bar plot shows evaluation success rates on each of the 18 semantic simulated kitchen tasks with 50 layout variations per task. Fig. 6 in the main paper shows aggregate results, and detailed results for CACTI -Sim-10 Benchmark.