CACTI: A Framework for Scalable Multi-Task Multi-Scene Visual Imitation Learning

Anonymous Author(s)
Affiliation
Address
`email`

Abstract: Developing robots that possess a diverse repertoire of behaviors and exhibit generalization in unknown scenarios requires progress on two fronts: efficient collection of large-scale and diverse datasets, and training of high-capacity policies on the collected data. While large and diverse datasets unlock generalization capabilities, like that observed in computer vision and natural language processing, collection of such datasets is particularly challenging for physical systems like robotics. In this work, we propose a framework to bridge this gap and scale robot learning, under the lens of multi-task, multi-scene robot manipulation in kitchen environments. Our framework, named CACTI, has four stages that separately handle data collection, data augmentation, visual representation learning, and imitation policy training. We demonstrate that, in a simulated kitchen environment, CACTI enables training a single policy on 18 semantic tasks across up to 50 layout variations per task. When instantiated on a real robot setup, CACTI results in a policy capable of 5 manipulation tasks involving kitchen objects, and robust to varying distractor layouts. The simulation task benchmark and augmented datasets in both real and simulated environments will be released to facilitate future research.

1 Introduction

Inspite of recent advances in learning based control, developing a general-purpose embodied agent with human-level abilities for generalizable skills is still a distant goal. Since the internet generates quality datasets, not random sets of words or images, and so large-scale internet data has shown significantly improved results even with the same underlying algorithm in natural language processing (NLP) and computer vision (CV) [1, 2, 3]. However, in embodied AI, especially robotics, not just quality data, but even random data is not possible to collect at scale due to operational challenges: unlike the abundant textual data from the internet and single-image annotations, tele-operating robots to collect demonstrations is much more laborious and time-consuming. Another challenge lies in incorporating diversity to the data: in robot manipulation, for example, covering a wide range of objects and scenes demands a large amount of physical resources.

In this work, we set out to address the above challenges by developing a framework for a *single* embodied agent to learn to solve a repertoire of tasks in multiple-scenes. We instantiate the framework in a robot manipulation setting with visual observations instead of state-based representations in order to help with generalization to changing scenes during deployment, where the states of objects might not be precisely available. There are several design decisions with respect to data collection, and learning policies to operate in scenes based on the collected data. End-to-end approaches like reinforcement learning (RL) that interleave data collection with policy learning are not ideal as they rely on deploying sub-optimal policies to collect data. On the other extreme, imitation learning (IL) by collecting a large dataset of expert demonstrations is infeasible due to constraints on availability of diverse experts, and challenges in fitting end-to-end neural networks to diverse datasets. Instead of

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

developing monolithic frameworks based on traditional RL and IL, we develop a four staged approach that breaks down monolithic blocks into manageable pieces in accordance with their expense.

Incorporating the above considerations, we propose a framework, namely CACTI , that can be divided into four stages, with the following decomposition: *Collect* - gather data with task specific experts, *Augment* - multiply data to boost experience diversity, *Compress* - project to a informative but low dimensional latent space, and *TraIn* – recover a general multi-task agent. Concretely, the four stages involve limited collection of data by either a human expert or a task-specific learned expert, data multiplication by augmenting the expert data with visual scene and layout variations, out-of-domain visual embedding learning, and training a single policy that utilizes the visual embeddings to imitate expert behavior on augmented data across multiple tasks. Figure 1 shows a schematic overview of the framework. We demonstrate in section 2 that it is possible to instantiate this framework both in sim and in real world using standard techniques.

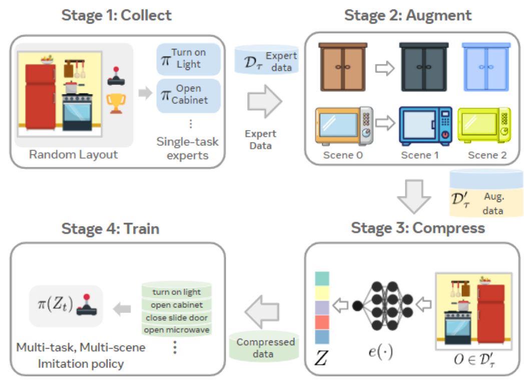


Figure 1: **Framework overview.** Schematic representation of the proposed framework, CACTI 's four stages.

In summary, we present a framework for large-scale, vision-based multi-task imitation learning with the following contributions: 1) fast limited in-domain data collection with in-domain experts, 2) efficient multiplication of data with diverse augmentations, 3) single visual policy learning with compressed representations, that generalizes across diverse task and scene variations, 4) multi-layout multi-task simulation framework with different benchmarks that we open-source to the community

2 A framework for Multi-Task Multi-Scene Visual Imitation Learning

Conceptually, CACTI involves four stages, as illustrated in Fig. 1: *Collect* - gather limited in-domain data with task specific experts, *Augment* - multiply data to boost the number of trajectories and diversity across them, *Compress* - project to an informative but low dimensional latent space that disentangles some factors of variations in the observations, and *TraIn* - recover a general muti-task agent on the augmented dataset, using compressed observation representations with a single policy. The subsequent subsections elaborate on each of the four stage in CACTI and their implementation in both simulation and the real world.

2.1 Collect: Small in-domain expert data collection

The goal of this stage is to collect a limited amount of expert demonstrations, while minimizing the cost of data collection in terms of both human labor (tele-operationg real robot) and computational cost (training RL experts in simulation).

We set up a toy kitchen tabletop with a Franka robot arm; the objects we use are shown in Fig. 6. Since it is much more cost expensive to train RL expert policies on the real robot, we opt to incorporate kinesthetic teaching by a human expert as a means of collecting trajectories. We define 5 tasks that involve manipulating the tabletop objects, and expert demonstrations are collected in a single-object, single-task setting. A human holds the robot and guides it to perform a task, and we save the joint pose and end-effector information of the robot at each time-step. For each of the 5 tasks, the demonstrator collects 8 trajectories of kinesthetic demonstrations.

Please see Appendix for details on data collection in our simulation environment.

2.2 Augment: Semantic scene variations for augmentation

In this stage, we aim to increase the diversity of data collected in stage one before using it for visual policy learning. To do so, we introduce two types of augmentations, *visual* and *semantic*. Visual augmentations involve changing attributes like color and texture of all the objects, and scene lighting. Semantic augmentations involve changing the layout of objects in the scene, namely their positions and orientations. Together, these augmentations help significantly multiply the limited data $\mathcal{D}_\tau$ collected by task-specific experts in stage one, and yield the augmented dataset $\mathcal{D}'_\tau \forall \tau$.

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. 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, 4], specificially the open-sourced Stable Diffusion trained model [4], 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. The generated images place plausible objects like mugs, cups, and glasses on locations of the white-colored table that are unoccupied. Please refer to Appendix section A.2 for details of augmentations in simulation and the real-world.

2.3 Compress: Representations pre-trained on internet data

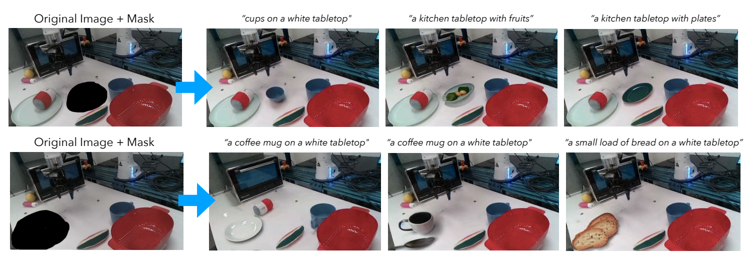


Figure 2: **In-painting augmentations.** Visualization of automatic data augmentation based on controlled generation on a scene from our real-robot environment. We specify a region of the image to be edited (a mask), and a text prompt, and sample several resulting model generations. We use the latest stable-diffusion model [5] that's fine-tuned specifically for image inpainting.

The Compress stage of our framework involves encoding image observations into low-dimensional embeddings, which makes it easier for the downstream policy to learn across complex semantic variations in the scene, and potentially generalize to new scenes with different attributes. This also helps to decouple representation and policy learning, and independently optimizing for each component through separate methods and architectures. We explore the use of representation networks trained with large-scale out-of-domain internet data, as well as representation models trained with only in-domain data from the simulator. For the former case, we use the R3M model [6] which has demonstrated strong empirical performance in various imitation learning tasks. For the latter, we train a ResNet-50 model using MoCo [7] on the in-domain data.

2.4 Multi-Task Multi-Scene Visual Policy Learning

The final stage is about learning a single policy with the visual backbone from stage three, trained on the entire multi-task multi-scene data respectively in simulation and the real environment. The overall goal-conditioned policy architecture, and the deployment protocol after stage 4 is shown in Fig. 8. During training, the goals o_g are sampled from the last 10 steps in each augmented trajectory, and during deployment, are provided by the experimenters. At time-step t, the input observation o_t and goal observation o_g are respectively embedded to latent representations z_t, z_g by the encoder from stage three. The embeddings are concatenated and fed to an MLP that eventually outputs the mean and co-variance of a Gaussian action distribution. The policy training loss is the usual behavior cloning loss that maximizes log-likelihood of the policy under the data distribution.

3 Experiments

Through experiments on simulated and real-robot environments, we aim to understand the following research questions: 1) How effective is CACTI in learning task behaviors for diverse scenes, compared to monolithic approaches? 2) How do variations in instantiation details of the different stages of CACTI affect the behavior of the final policy? 3) How do the learned policies in CACTI generalize to scenes with different objects, and variations compared to the training datasets?

3.1 Environment and Evaluation details

We setup a simulated kitchen environment with 18 tasks involving eight objects: four burner knobs, one light switch, one kettle, one cabinet with sliding door, one cabinet with a left and a right door, and one microwave. A multi-task agent gets communicated about which task to execute through a task embedding that contains both the targeted object pose and the object arrangement information that's unique to each layout. We have a similar real-robot setup as the simulated kitchen but on a smaller scale. Fig. 6 shows all the objects we have in the real scene, that include toasters, plates, mugs, strainers, cans, ketchup bottles, and several fruits. Based on these objects, we define each task to be the manipulation of an object from an initial location to a goal location. We define five tasks in this environment described visually in Fig. 7. Additional details are in the Appendix.

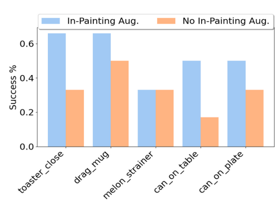
3.2 Framework Ablations and policy baselines

In the real robot environment, we evaluate the novel in-painting based semantic augmentation, by training two visual multi-task policies: one with data augmented with in-painted trajectory images, and the other without this augmentation. For the real robot experiments, we use the out-of-domain pre-trained R3M model, which we fine-tune during stage 4 of learning the policy.

Additional details about the variants and experiment settings for simulation, and real environments are mentioned in Appendix section A.5.

3.3 Results

Fig. 3 shows results for the real-robot experiments, where both the evaluated variants achieve reasonable success rates across all the tasks, demonstrating utility of the overall framework . We observe that the policy trained with in-painted data augmentations achieves on average around 20% absolute and 60% relative higher success rates compared to the one trained without these augmentations. This shows the importance of the in-painted augmentations in scaling up useful data without human hours being used, and potentially opens up interesting research directions at the intersection of generative modeling and robot learning.



CACTI -Real	Avg.
In-Paint Aug.	53.4
No In-Paint Aug.	33.2

Figure 3: **Real world evaluation.** We report results from the real robot environment tasks using the evaluation setup described in section 3.1. The two compared multi-task policies were both evaluated for 30 episodes on each of the 5 tasks. The bar chart (left) shows success rates averaged within each task, and the final results in the Table show (right) are averaged over all episodes in all 5 tasks.

4 Discussion and Conclusion

In this paper, we developed a framework for multi-task multi-scene visual imitation learning, and instantiated it both in simulation and in the real world. Our framework incorporates several components like fast and efficient data collection, novel data augmentation, compressed visual representations, and a single control policy trained over augmented datasets. We demonstrate efficacy of the framework in a large-scale simulated kitchen environment with several variations in the tasks, type of objects, and randomizations in the scene, and in the real-world tasks, we show the efficacy of novel augmentations like in-painting images based on prompting a deep generative model.

References

[1] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[3] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[6] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[7] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta. The unsurprising effectiveness of pre-trained vision models for control. *arXiv preprint arXiv:2203.03580*, 2022.

[8] S. M. Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[9] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[10] X. Chen, C. Wang, Z. Zhou, and K. Ross. Randomized ensembled double q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982*, 2021.

[11] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

[12] A. Nichol and J. Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

[13] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

[14] L. Pinto and A. Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2161–2168. IEEE, 2017.

[15] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pages 1407–1416. PMLR, 2018.

[16] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Wiele, V. Mnih, N. Heess, and J. T. Springenberg. Learning by playing solving sparse reward tasks from scratch. In *International conference on machine learning*, pages 4344–4353. PMLR, 2018.

[17] Z. Mandi, P. Abbeel, and S. James. On the effectiveness of fine-tuning versus meta-reinforcement learning. *arXiv preprint arXiv:2206.03271*, 2022.

[18] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[19] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

[20] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[21] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine. Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31, 2018.

[22] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[23] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR, 2017.

[24] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto. Visual imitation made easy. In *Conference on Robot Learning (CoRL)*, 2020.

[25] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[26] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[27] A. Srinivas, M. Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[28] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020.

[29] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[30] C. G. Rivera, D. A. Handelman, C. R. Ratto, D. Patrone, and B. L. Paulhamus. Visual goal-directed meta-imitation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3767–3773, 2022.

[31] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

[32] N. M. M. Shafiullah, Z. J. Cui, A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *arXiv preprint arXiv:2206.11251*, 2022.

[33] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 23–30. IEEE, 2017.

[34] S. James, A. J. Davison, and E. Johns. Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. *CoRR*, abs/1707.02267, 2017. URL http://arxiv.org/abs/1707.02267.

[35] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *arXiv preprint arXiv:1710.06537*, 2017.

[36] E. Tzeng, C. Devin, J. Hoffman, C. Finn, P. Abbeel, S. Levine, K. Saenko, and T. Darrell. Adapting deep visuomotor representations with weak pairwise constraints. *arXiv preprint arXiv:1511.07111*, 2015.

[37] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel. Asymmetric actor critic for image-based robot learning. *CoRR*, abs/1710.06542, 2017. URL http://arxiv.org/abs/1710.06542.

[38] E. Tzeng, C. Devin, J. Hoffman, C. Finn, X. Peng, S. Levine, K. Saenko, and T. Darrell. Towards adapting deep visuomotor representations from simulated to real environments. *CoRR*, abs/1511.07111, 2015.

[39] F. Sadeghi and S. Levine. (cad)$ˆ2$rl: Real single-image flight without a single real image. *CoRR*, abs/1611.04201, 2016. URL http://arxiv.org/abs/1611.04201.

[40] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[41] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[42] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *arXiv preprint arXiv:1506.07365*, 2015.

[43] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[44] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[45] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[46] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

[47] K. Xie, H. Bharadhwaj, D. Hafner, A. Garg, and F. Shkurti. Latent skill planning for exploration and transfer. In *International Conference on Learning Representations*, 2020.

[48] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019.

[49] K. Gregor, D. J. Rezende, F. Besse, Y. Wu, H. Merzic, and A. v. d. Oord. Shaping belief states with generative environment models for rl. *arXiv preprint arXiv:1906.09237*, 2019.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[51] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta. The unsurprising effectiveness of pre-trained vision models for control. *arXiv preprint arXiv:2203.03580*, 2022.

A More details on framework design

A.1 Stage 1: Details

We create a simulated environment that supports 18 semantic tasks and randomly-generated layout variations. Each layout has a different arrangement of the main kitchen objects (for example, placing the microwave next to the sink v.s. in the top shelf next to the cabinet). We use a standard on-policy RL algorithm, namely NPG [8], to train single-task, single-layout expert policies $\pi(s_t)$ from state-based input observations s_t. For each of the 18 tasks, we gather 50 expert policies each corresponding to a different layout, hence a total of 900 policies. In implementation, we initialize a large batch of parallel RL training runs, and use a threshold of 90% success rate to filter converged policies as experts. 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.

A.2 Stage 2: Details

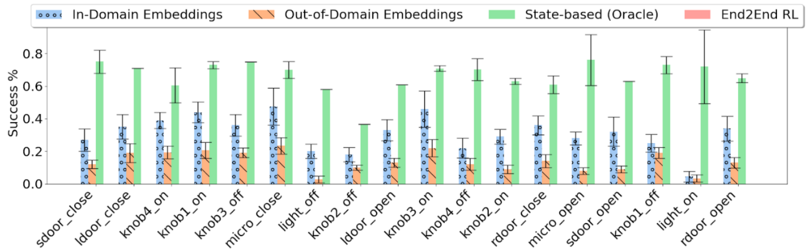
For augmenting the real-robot kinesthetic demos collected by experts, we replay the trajectories while varying different attributes of the scene, and recording per-timestep image observations during the replays. The visual augmentations in the real-robot setting correspond to color jitters of the observation 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, 4] 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 [4], 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. 2 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.

A.3 Stage 3: Details

The pre-trained visual representations for R3M are obtained through training on egocentric human videos [9], 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.

A.4 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.



CACTI -Sim-10	Avg. Success
State-based	68.8 ± 2.9
Out-of-Domain	17.5 ± 1.2
In-Domain	40.3 ± 2.5
End2End RL	0.0 ± 0.0
CACTI -Sim-50	**Avg. Success**
State-based	66.8 ± 2.3
Out-of-Domain	24.5 ± 0.1
In-Domain	23.8 ± 2.5
End2End RL	0.0 ± 0.0

Figure 4: **CACTI -Sim-10 Benchmark results.** The bar plot shows evaluation success rates on each of the 18 semantic simulated kitchen tasks with 10 layout variations per task. The table shows results averaged over all the tasks for CACTI -Sim-10 and CACTI -Sim-50 respectively. Detailed results of CACTI -Sim-50 are in Appendix section A.6.

A.5 Experiment setup details

A.5.1 Simulation environment

For the simulation benchmark, we compare against a state-based agent (simulator states as input instead of scene images) that is trained through the stage four procedure across all 18 tasks, in 10 layouts per task (CACTI -Sim-10) and 50 layouts per task (CACTI -Sim-50). By design, this policy is agnostic to visual scene augmentations, but must learn to generalize across the semantic layout variations. The performance of this agent is an approximation of the upper bound on visual policy learning behavior in this benchmark. We evaluate two different choices for stage three, namely out-of-domain embeddings, in-domain embeddings [7] trained on the augmented data. In addition, we evaluate CACTI against a monolithic framework of end-to-end RL training across the same set of task and layout variations. We use a REDQ agent [10] for RL training, and report results after training across 1M environment steps per-task.

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.

A.5.2 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.

A.6 Additional results

Fig. 5 shows detailed results for the **CACTI -Sim-50 Benchmark** that was forward referenced, with aggregate values in Fig. 6 of the main paper.

B Results on Simulation Benchmark

Fig. 4 shows results of the different variants on the CACTI -Sim-10 benchmark (bar graph) and also average results across tasks in the Table. We see that the state-based visual imitation policy achieves an average success rate of 65-70% across all the tasks. This oracle serves as an upper bound for the the visual policy variants. The policy trained with in-domain embeddings achieves on average 40% success rate in CACTI -Sim-10 while the policy with out-of-domain embeddings achieves around 18%. The out-of-domain embedding version is comparable to in-domain for CACTI -Sim-50 that requires generalization to more diverse variations. Interestingly, both these variants significantly outperform the monolithic RL baseline, trained from scratch for upto 1M environment steps per task,

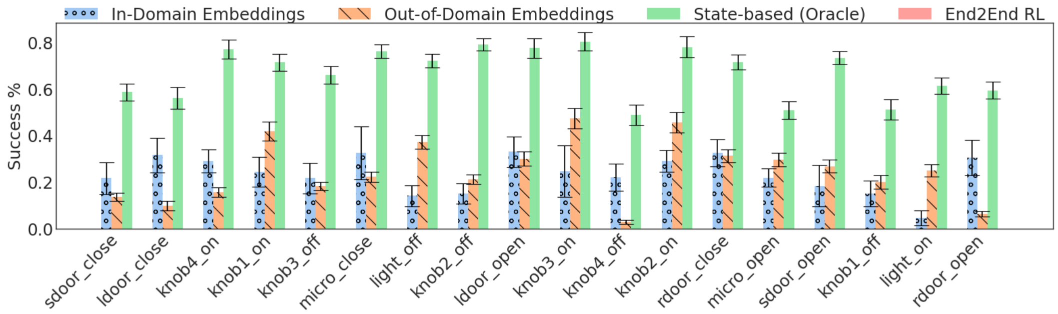


Figure 5: **CACTI -Sim-50 Benchmark results.** The bar plot shows evaluation success rates on each of the 18 semantic simulated kitchen tasks with 50 layout variations per task. Fig. 6 in the main paper shows aggregate results, and detailed results for CACTI -Sim-10 Benchmark.

which obtains a success rate of 0. This also suggests the non-triviality of the CACTI -Sim benchmark, which we will open-source to the community for future frameworks to evaluate their approaches.

C Related Work

Scaling robot learning frameworks. Prior works on scaling robot learning have largely focused on the RL paradigm, either through multi-task RL [11] or meta-RL [12, 13] and shown that shared learning among tasks amortizes the cost of acquiring diverse behaviors compared to training single policies for individual tasks [14, 15, 16]. The main reason for success in these settings has been that most tasks share some common structure (for example reaching and grasping behavior components), and such structures can be discovered through the learning of shared policy. This is useful from the perspective of designing frameworks that are scalable with efficient re-use of data across tasks. Recent work [17] has found that learning pre-trained representations and simple multi-task learning outperforms most meta RL approaches. There have been similar findings on IL from large offline datasets [18]. CACTI is inspired by these findings where we collect offline data, and use pre-trained visual representations for multi-task IL on the offline data, but instead of collecting all the data by experts [18] (which is expensive in robotics), we have an efficient data augmentation scheme for multiplying a small set of expert data. In the next paragraphs, we discuss CACTI 's four stages in relation with respective prior works.

Visual policy learning. Learning control policies from visual observations helps amortize the cost of learning representations of recurring objects and scenes [19, 20, 21, 22, 23, 24, 25]. However several prior works have looked at visual policy learning in simple simulated environments like the DM Control Suite [26] that involves stick agents locomoting [27, 28, 22] or in simplified manipulation environments like MetaWorld that involves only a few objects in the scene with a robot arm [29, 30]. Other works have tackled policy learning in much more complex settings like a simulated realistic looking kitchen with several objects, but assume ground-truth simulator state observations instead of visual inputs [31, 32]. In contrast, CACTI (sim) is based on a simulated kitchen similar to [31] but with much more diversity of visual observations and layouts, and incorporates only visual observations as inputs to the multi-task multi-scene agents making it readily amenable for real-world environments where it is not possible to obtain ground-truth states of objects in the scene.

Domain randomization. Domain randomization [33, 34, 35, 36, 37, 38, 39] bridges the reality gap by leveraging rich variations of the simulation environment during training. The hope is that by adding random variability in the simulator, the real data distribution will be within that of the training data. This has been useful in recent advances for visual navigation and manipulation in real-world environments [40]. Inspired by similar ideas, we go beyond simple domain randomization like color jitters, camera motions, texture changes, to more semantic augmentations based on distractor objects, and layout variations, through hindsight relabeling of limited expert demonstrations. We also incorporate a novel image in-painting [41] based data augmentation that lets us add different realistic objects in the scene by running inference through trained generative models [2, 4].

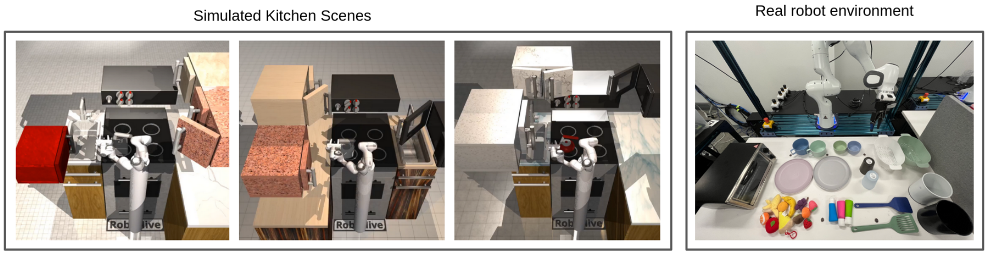


Figure 6: **Environment variations.** Visualization from random scene variations in the simulated kitchen environment (left) and the set of all objects in the real robot environment (right). The scenes in simulation have randomized object layouts, with different colors, textures, and lighting conditions. Both the simulation and the real environment have a Franka Emika Panda arm that is operated through joint position control.

Figure 7: **Real robot tasks.** Illustration of the five real robot tasks, namely: drag mug, close toaster, place can on the plate, place can on the table, put watermelon in strainer. The colored arrows approximate the task trajectories.

Representation learning for control. Recent progress in video prediction and self-supervised learning, such as developing suitable lower bounds to mutual information (MI) based objectives [42, 43, 44, 22, 45, 46, 47, 48, 49], have enabled learning of visual representations that are useful for downstream tasks. Prior work have examined pretraining on large datasets like ImageNet [50] and Ego4D [9], and using the frozen representation for doing downstream robot control [51, 6]. CACTI leverages such frameworks for learning compressed visual representations, both with out of domain internet data of human videos, and with in-domain augmented dataset that is generated as part of the framework.

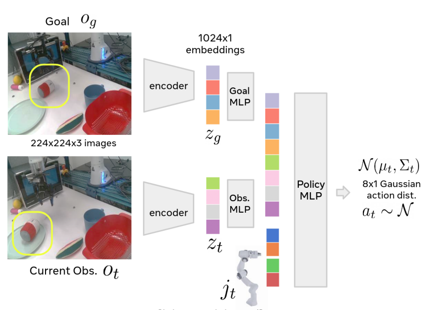


Figure 8: **Policy deployment pipeline.** Schematic of the deployment setup for the final multi-task multi-scene visual imitation policy.