Zero-Shot Robotic Manipulation with Pre-Trained Image-Editing Diffusion Models

Anonymous Author(s)

Abstract

If generalist robots are to operate in truly unstructured environments, they need to be able to recognize and reason about novel objects and scenarios. Such objects and scenarios might not be present in the robot's own training data. We propose SuSIE, a method that leverages an image editing diffusion model to act as a high-level planner by proposing intermediate subgoals that a low-level controller attains. Specifically, we fine-tune InstructPix2Pix (Brooks et al., 2023) on robot data such that it outputs a hypothetical future observation given the robot's current observation and a language command. We then use the same robot data to train a low-level goal-conditioned policy to reach a given image observation. We find that when these components are combined, the resulting system exhibits robust generalization capabilities. The high-level planner utilizes its Internet-scale pre-training and visual understanding to guide the low-level goal-conditioned policy, achieving significantly better generalization than conventional language-conditioned policies. We demonstrate that this approach solves real robot control tasks involving novel objects, distractors, and even environments, both in the real world and in simulation. The project website can be found at `http://subgoal-image-editing.github.io`.

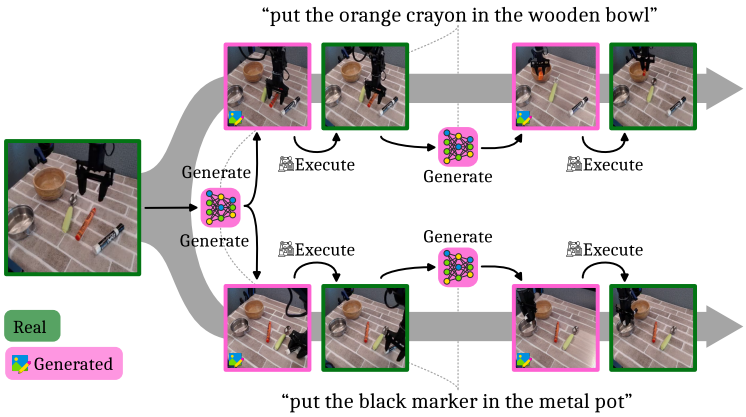


Figure 1: **SuSIE leverages a pre-trained image editing model to generate future image subgoals based on a language commands.** A low-level goal-reaching policy then executes the actions needed to reach each subgoal. Alternating this loop enables us to solve the task.

1 Introduction

A useful generalist robot must be able to — much like a person — recognize and reason about novel objects and scenarios it has never encountered before. For example, if a user instructs the robot to "hand me that jumbo orange crayon," it ought to be able to do so even if it has never interacted

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

with a jumbo orange crayon before. In other words, the robot needs to possess not only the physical capability to manipulate an object of that shape and size but also the semantic understanding to reason about an object outside of its training distribution. As much as robotic manipulation datasets have grown in recent years, it is unlikely that they will ever include every conceivable instance of objects and settings, any more so than the life experiences of a person ever include physical interactions with every type of object. While these datasets contain more than enough examples of manipulating elongated cylindrical objects, they lack the broad semantic knowledge necessary to ground the *particular* objects that robots will undoubtedly encounter during everyday operation.

In this paper, we develop an approach for leveraging a class of pre-trained image-editing models (e.g., InstructPix2Pix (Brooks et al., 2023)) for improving motor control and policy execution. Our key insight is to decompose the robotic control problem into two phases: first, synthesizing a "subgoal" that the robot must reach to complete the user-specified task, and then, attempting to reach this subgoal via a goal-reaching robot controller. The first phase of this recipe incorporates semantic information by fine-tuning an image-editing model on robot data such that, given the robot's current observation and a natural language command, the model generates a hypothetical *future* subgoal that allows the robot to complete the command. We then employ a low-level goal-reaching policy to reach this hypothetical future subgoal.

2 SuSIE: Subgoal Synthesis via Image Editing.

Our goal is to utilize semantic information from the Internet to improve language-guided robot control in novel environments, scenes, and objects. How can we do this when models trained on general-purpose Internet data do not provide guidance in selecting low-level actions? Our key insight is that we can still utilize some sort of a pre-trained model for guiding low-level control if we could decouple the robot control problem into two phases: **(i)** imagining subgoals that would need to be attained to succeed at the task, and **(ii)** learning low-level control policies for reaching these generated subgoals. Our method incorporates semantic information from Internet pre-training in phase (i), by fine-tuning a text-guided image-editing model for subgoal generation. Phase (ii) is accomplished via a goal-conditioned policy trained only on robot data. We describe each of these phases in Section D in Appendix.

2.1 Phase (i): Synthesizing Subgoals From Image Editing Models

The primary component of our method is a generative model that, given a target task specified in natural language, can guide the low-level controller towards a state that it must try to attain in order to solve the task. One way to accomplish this is to train a generative model to produce an immediate next way-point or subgoal frame. We can then incorporate semantic information from the Internet into our algorithm by initializing this generative model with a suitable pre-trained initialization, followed by fine-tuning it on multi-task, diverse robot data.

What is a good pre-trained initialization for initializing this model? Our intuition is that since accomplishing a task is equivalent to "editing" the pixels of an image of the robot workspace under controls prescribed by the language command, a favorable pre-trained initialization is provided by a language-guided image-editing model. We instantiate our approach with Instruct pix2pix (Brooks et al., 2023), though other image editing models could also be used. Formally, this model is given by $p_\theta(\mathbf{s}_{\text{edited}}|\mathbf{s}_{\text{orig}}, l)$. Then, using the dataset $\mathcal{D}$ of robot trajectories, we fine-tune p_θ on tuples containing a pair of images sampled from a trajectory and the corresponding language annotation: $(\mathbf{s}_{\text{orig}} := \mathbf{s}_i^k, \mathbf{s}_{\text{edited}} := \mathbf{s}_j^k, l_k)$, where $\mathbf{s}_j$ is a state that appears after $\mathbf{s}_i$ ($j > i$). During fine-tuning, we run gradient descent on the following objective, starting from $\theta_0 := \theta_{\text{pre-trained}}$:

$$\min_\theta \quad -\mathbb{E}_{(\tau^k, l_k)\sim\mathcal{D};\ \mathbf{s}_i^k\sim\tau^k;\ j\sim q(j|i)} \left[\log p_\theta\left(\mathbf{s}_j^k|\mathbf{s}_i^k, l_k\right)\right]. \tag{1}$$

We need to choose the distribution q over the time-step j given a state $\mathbf{s}_i^k$ for fine-tuning the image-editing model as in Equation 3. Since we model the next subgoal that the low-level controller should attain, and since the depending upon the task, this subgoal could be arbitrarily close to the original state $\mathbf{s}_i$, we require valid tuples $(\mathbf{s}_i, \mathbf{s}_j, l)$ used for fine-tuning p_θ in Equation 3 to have values of j in a bounded interval around i, specifically we choose $j \in [i, i+k]$, where k is a fixed hyperparameter.

2.2 Phase (ii): Reaching Generated Sub-Goals with Goal-Conditioned Policies

In order to utilize the fine-tuned image-editing model to actually control the robot, we further need to train a low-level controller to actually select suitable robot actions. In this section, we present

the design of our low-level controller, followed by a full description our test-time control procedure. Since the image-editing model in SuSIE produces images of future subgoals conditioned on natural language task descriptions, our low-level controller can simply be a language-agnostic goal-reaching policy that aims to reach these generated subgoals.

Training a goal-reaching policy. Our goal-reaching policy is parameterized as $\pi_\phi(\cdot|\mathbf{s}_i, \mathbf{s}_j)$, where $\mathbf{s}_j$ is a future frame that the policy intends to reach, by acting at $\mathbf{s}_i$. At test time, we only need the low-level goal-conditioned policy to be proficient at reaching close-by states that lie within k steps of a given state since the image editing model from phase (i) is also trained to produce subgoals within k steps of any state. To train this policy, we run goal-conditioned behavioral cloning (GCBC) on the robot data, utilized previously in phase (i). In addition, we can also leverage robot data $\mathcal{D}'$ that does not contain language annotations. Formally, our training objective is given by:

$$\max_{\phi} \; \mathbb{E}_{\tau^i \sim \mathcal{D} \cup \mathcal{D}';\, (\mathbf{s}_i^k, \mathbf{a}_i^k) \sim \tau^k;\, j \sim q(j|i)} \left[\log \pi_\phi(\mathbf{a}_i^k | \mathbf{s}_i^k, \mathbf{s}_j^k) \right], \tag{2}$$

where $q(j|i)$ is the distribution over future frames that we previously utilized in Equation 3.

Test-time control with π_ϕ and p_θ. Once both the goal-reaching policy π_ϕ and the image editing subgoal generation model p_θ are trained, we utilize them together to solve new manipulation tasks based on user-specified natural language commands. Given a new scene, $\mathbf{s}_0^{\text{test}}$, and a language task description l^{test}, SuSIE attempts to solve the task by iteratively generating subgoals and commanding the low-level goal-reaching policy with these subgoals. At the start, we sample the first subgoal $\widehat{\mathbf{s}}_+^{\text{test}} \sim p_\theta(\cdot|\mathbf{s}_0^{\text{test}}, l^{\text{test}})$. Once the subgoal is generated, we then roll out the goal-reaching policy π_ϕ, conditioned on $\widehat{\mathbf{s}}_+^{\text{test}}$, for k time-steps, such that each action is chosen according to $\mathbf{a}_j^{\text{test}} \sim \pi_\phi(\cdot|\mathbf{s}_j^{\text{test}}, \widehat{\mathbf{s}}_+^{\text{test}})$. After k time steps, given the current image $\mathbf{s}_k^{\text{test}}$, we refresh the subgoal by sampling from the image-editing model again and repeat the process. Note crucially that this recipe does not require that the subgoal $\mathbf{s}^{\text{test}}$ be attained after k steps, as the generative model effectively "replans" a new subgoal based on the current observation. Overall, at test time, we alternate between obtaining a new subgoal from p_θ and commanding the goal-reaching policy to attain this subgoal, until a maximum number of allowed time steps. Pseudocode is provided in Algorithm 2.

Algorithm 1 SuSIE: Zero-Shot, Test-Time Execution

Require: subgoal model $p_\theta(\mathbf{s}_+|\mathbf{s}_t, l)$, policy $\pi_\phi(\cdot \mid \mathbf{s}_t, \mathbf{s}_+)$, language command l^{test}, max episode length T, goal sampling interval K, initial state $\mathbf{s}_0^{\text{test}}$

$t \leftarrow 0$
while $t \leq T$ **do**
  Sample $\mathbf{s}_+^{\text{test}} \sim p_\theta(\mathbf{s}_+|\mathbf{s}_t^{\text{test}}, l^{\text{test}})$ ▷ Sample a new subgoal every K steps
  for $j = 1$ to k **do**
    Sample $\mathbf{a}_t \sim \pi_\phi\left(\cdot \mid \mathbf{s}_t^{\text{test}}, \mathbf{s}_+^{\text{test}}\right)$ ▷ Predict the action from current state and subgoal
    Execute $\mathbf{a}_t$
    $t \leftarrow t + 1$
  end for
end while

3 Can SuSIE Generate Plausible and Meaningful Subgoals?

We start by presenting qualitative examples of intermediate subgoals generated by the SuSIE image-editing model in Figure 4. Even on previously unseen trajectories and language commands, the model is able to produce visually high-quality and useful subgoals involving the gripper grasping and moving objects. This is nontrivial since it requires the model to have not only the *semantic knowledge* to detect which pixels in the image correspond to a given object, but also an understanding of *dynamics* to predict how to move and rotate the gripper to grasp it.

4 Is the Synthesized Subgoal Useful for Completing New Commands?

Real-world experimental setup and datasets. We conduct our real-robot experiments on a WidowX250 robot platform. Our robot dataset is BridgeData V2 (Walke et al., 2023). The dataset contains over 60k trajectories, 24 environments, 13 skills, and hundreds of objects. Our evaluations present three different scenarios 3, designed specifically to test the ability of various methods at different levels of open-world generalization: **Scene A:** this scene includes an environment and objects

	No. of Instructions Chained				
	1	2	3	4	5
HULC (Mees et al., 2022a)	0.43	0.14	0.04	0.01	0.00
MCIL (Lynch & Sermanet, 2020)	0.20	0.00	0.00	0.00	0.00
MdetrLC (Ge et al., 2023)	0.69	0.38	0.20	0.07	0.04
AugLC (Ge et al., 2023)	0.69	0.43	**0.22**	0.09	0.05
LCBC (Walke et al., 2023)	0.62	0.31	0.14	0.05	0.01
UniPi (Ours) (Du et al., 2023)	0.56	0.16	0.08	0.08	0.04
UniPi (HiP) (Ajay et al., 2023b)	0.08	0.04	0.00	0.00	0.00
SuSIE (Ours)	**0.75**	**0.46**	0.19	**0.11**	**0.07**

Table 1: **Comparison of SuSIE and other prior approaches on CALVIN.** SuSIE is able to chain together more instructions with a higher success rate than all of these prior methods.

	Task	LCBC/MOO	RT-2-X	Ours
Scene A	Eggplant on plate	0.4	0.3	**0.8**
	Carrot on plate	0.3	0.4	**0.7**
	Eggplant in pot	0.4	0.6	**0.7**
	Average	0.37	0.43	**0.73**
Scene B	Bell pepper in pot	0.0	0.0	**0.2**
	Bell pepper in bowl	0.1	0.0	**0.4**
	Average	0.05	0.00	**0.30**
Scene C	Toothpaste in bowl	0.0	**0.5**	**0.5**
	Crayon in bowl	0.0	**0.9**	0.6
	Spoon in bowl	0.3	**0.7**	0.4
	Bowl to top	0.2	**0.9**	0.3
	Average	0.13	**0.75**	0.45

Table 2: **Real-world performance.** SuSIE consistently achieves the best success rates in Scenes A and B, and is able to attain a high absolute success rate of 45% on the most challenging Scene C.

that are well-represented in BridgeData V2; **Scene B:** this scene is situated in an environment with a seen tabletop but a novel background and distractors, where the robot must move a seen object (bell pepper) into a choice of seen container (orange pot) or unseen container (ceramic bowl); and **Scene C:** this scene includes a table texture unlike anything in BridgeData V2 and requires manipulating both seen and unseen objects.

Real-world results. We present performance of real-world evaluations in Table 2. Observe that in Scene A, SuSIE achieves the highest success rate on all three tasks, attaining an average success rate of 73% which improves over RT-2-X (Anonymous) by **69%**. In Scene B, SuSIE again outperforms other prior approaches on the two tasks, successfully grounding both the novel ceramic bowl and the previously seen orange pot. In the most challenging Scene C (unseen domain, unseen objects), SuSIE attains a success rate of 45%, outperforming MOO (Stone et al., 2023) by about **260%**. However, RT-2-X outperforms SuSIE in this scene. We believe that the superior performance of RT-2-X compared to SuSIE in Scene C is because it is a much larger 55B parameter model, initializes from a proprietary VLM, and is trained on much more data — including BridgeData V2, but also a vast quantity of additional tabletop manipulation. These differences in the amount of data and parameters put our method, which only utilizes BridgeData V2, at quite an unfair advantage against RT-2-X. Nevertheless, SuSIE is still able to recognize the novel objects and attain a high absolute success rate of 45%.

Simulation results. We also conducted simulation experiments in CALVIN benchmark (Mees et al., 2022b), where we compared to LCBC, UniPi (Du et al., 2023) and several other state-of-the-art methods, and showed that SuSIE outperforms all prior methods on this benchmark. Further details are described in Section E in Appendix.

5 Does SuSIE Improve Precision and Low-Level Skill Execution?

In this section, we visualize some evaluation rollouts from our experiments in Scene A to understand if SuSIE works merely because it enhances the generalization of the policy to semantic changes in the visual observation or if it actually does improve the precision of the low-level control by commanding meaningful subgoals. Observe in Figure 5 that the RT-2-X policy often produces actions that fail to precisely orient the gripper around the target object or close the gripper early. In contrast, policy executions obtained via SuSIE are more precise, and execute actions that attempt to match the gripper and object positions to the generated subgoal, allowing the policy to succeed at the task. To understand the contribution of the subgoal prediction towards improved precision, we also evaluate an oracle GCBC policy on a subset of tasks. This policy is trained on identical robot data as SuSIE; however, we at test time we command the policy with a real image of the completed task, which our method does not require. Observe that even then this GCBC oracle fails to accomplish the task due to issues with imprecise object localization and untimely gripper closing. Corroborated by numerical results in Table 3, these experiments validate our claim that utilizing subgoal prediction is crucial for enabling precise low-level skill execution and control.

Table 3: **Comparison to GCBC with oracle goals.** Executing generated subgoals improves the performance of GCBC even when the latter is provided with a real goal image.

	Task	GCBC	Ours
Scene A	Eggplant on plate	0.4	**0.8**
	Carrot on plate	0.4	**0.7**
	Eggplant in pot	0.5	**0.7**
CALVIN	8 tasks involving non-prehensile motion	0.16	**0.92**

References

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=sP1fo2K9DFG.

Anurag Ajay, Seungwook Han, Yilun Du, Shaung Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *arXiv preprint arXiv:2309.08587*, 2023b.

Anonymous. Rt-2-x. In *Under review*.

Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2020.

Chethan Bhateja, Derek Guo, Dibya Ghosh, Anika Singh, Manan Tomar, Quan Ho Vuong, Yevgen Chebotar, Sergey Levine, and Aviral Kumar. Robotic offline rl from internet videos via value-function pre-training. 2023. URL https://api.semanticscholar.org/CorpusID:262217278.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023a.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pp. 287–318. PMLR, 2023b.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *arXiv preprint arXiv:2302.00111*, 2023.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

Yuying Ge, Annabella Macaluso, Li Erran Li, Ping Luo, and Xiaolong Wang. Policy adaptation from foundation model feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19059–19069, 2023.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu 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*, pp. 18995–19012, 2022.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022b.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*, 2022a.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022b.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2021.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1780–1790, 2021.

Ivan Kapelyukh, Vitalis Vosylius, and Edward Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. 2023.

Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020.

Donghwan Lee and Niao He. Stochastic primal-dual q-learning. *arXiv preprint arXiv:1810.08298*, 2018.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023.

Zhao Mandi, Homanga Bharadhwaj, Vincent Moens, Shuran Song, Aravind Rajeswaran, and Vikash Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*, 2022.

Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022a.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022b.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pp. 728–755. Springer, 2022.

Vivek Myers, Andre He, Kuan Fang, Homer Walke, Philippe Hansen-Estruch, Ching-An Cheng, Mihai Jalobeanu, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal representations for instruction following: A semi-supervised language interface to control. *arXiv preprint arXiv:2307.00117*, 2023.

Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *arXiv preprint arXiv:1909.05829*, 2019.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhi Gupta. R3m: A universal visual representation for robot manipulation. *ArXiv*, abs/2203.12601, 2022.

Alexander Pashevich, Robin Strudel, Igor Kalevatykh, Ivan Laptev, and Cordelia Schmid. Learning to augment synthetic images for sim2real policy transfer. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2651–2657. IEEE, 2019.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer, December 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rafael Rafailov, Tianhe Yu, A. Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. *Learning for Decision Making and Control (L4DC)*, 2021.

Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pp. 19561–19579. PMLR, 2022.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pp. 894–906. PMLR, 2022.

Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Brianna Zitkovich, Fei Xia, Chelsea Finn, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, pp. 2226–2240. PMLR, 2023.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. Multilingual Universal Sentence Encoder for Semantic Retrieval, July 2019.

Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18456–18466, 2023a.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *arXiv preprint arXiv:2102.08363*, 2021.

Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023b.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

A Introduction

A useful generalist robot must be able to — much like a person — recognize and reason about novel objects and scenarios it has never encountered before. For example, if a user instructs the robot to "hand me that jumbo orange crayon," it ought to be able to do so even if it has never interacted with a jumbo orange crayon before. In other words, the robot needs to possess not only the physical capability to manipulate an object of that shape and size but also the semantic understanding to reason about an object outside of its training distribution. As much as robotic manipulation datasets have grown in recent years, it is unlikely that they will ever include every conceivable instance of objects and settings, any more so than the life experiences of a person ever include physical interactions with every type of object. While these datasets contain more than enough examples of manipulating elongated cylindrical objects, they lack the broad semantic knowledge necessary to ground the *particular* objects that robots will undoubtedly encounter during everyday operation.

How can we imbue this semantic knowledge into language-guided robotic control? One approach to do this would be to utilize pre-trained models trained on vision and language to initialize different components in the robotic learning pipeline. Recent efforts attempt to do this, for example, by initializing robotic policies with pre-trained vision-language encoders (Brohan et al., 2023a) or utilizing pre-trained models for generating semantic scene augmentation (Chen et al., 2023; Yu et al., 2023b). While these methods bring semantic knowledge into robot learning, it remains unclear if these approaches realize the full potential of Internet pre-training in improving low-level motor control and policy execution, or whether they simply improve visual generalization of the policy.

In this paper, we develop an approach for leveraging a class of pre-trained image-editing models (e.g., InstructPix2Pix (Brooks et al., 2023)) for improving motor control and policy execution. Our key insight is to decompose the robotic control problem into two phases: first, synthesizing a "subgoal" that the robot must reach to complete the user-specified task, and then, attempting to reach this subgoal via a goal-reaching robot controller. The first phase of this recipe incorporates semantic information by fine-tuning an image-editing model on robot data such that, given the robot's current observation and a natural language command, the model generates a hypothetical *future* subgoal that allows the robot to complete the command. We then employ a low-level goal-reaching policy to reach this hypothetical future subgoal. Crucially, our image-editing model does not need to understand *how* to achieve this future subgoal, and on the other hand, the policy only needs to infer visuo-motor relationships to determine the correct actuation and does not require an understanding of the semantics. Furthermore, such subgoals can significantly simplify the task by inferring likely poses for the arm or intermediate sub-steps, such as grasping an object when the command requires repositioning it to a new location (see Figure 2). In fact, we observe in our experiments that while existing approaches often fail due to imprecise understanding of obstacles or object orientations, following the generated subgoals enables our method to perform well in such scenarios.

The main contribution of our work is **SU**bgoal **S**ynthesis via **I**mage **E**diting (**SuSIE**), a simple and scalable method for incorporating semantic information in pre-trained models to improve robotic control. The pre-trained image editing model is used with minimal modification, requiring only fine-tuning on robot data. The low-level goal-conditioned policy is trained with standard supervised learning, and faces the comparatively easier problem of reaching nearby image subgoals; this typically only requires attending to a single object or the arm position, ignoring most parts of the scene. Together, we find that this approach solves real robot control tasks involving novel objects, novel distractors, and even novel scenes, all of which are not observed at all in the robot training data.

B Related Work

Incorporating semantic information from vision-language pre-trained models. Prior works that incorporate semantic information from vision-language pre-trained models into robot learning can be classified into two categories. The first category aims to improve visual scene understanding in robot policies with semantic information from VLMs. For instance, GenAug (Chen et al., 2023), ROSIE (Yu et al., 2023b), DALL-E-Bot (Kapelyukh et al., 2023), and CACTI (Mandi et al., 2022) use text-to-image generative models to produce semantic augmentations of a given scene with novel objects and arrangements and train the robot policy on the augmented data to enable it to perform well in a similar scene. MOO Stone et al. (2023) utilizes a pre-trained object detector to extract bounding boxes that guide the robot policy towards the object of interest. Other works directly train language and image-conditioned policies (Brohan et al., 2022, 2023a; Shridhar et al., 2022), by

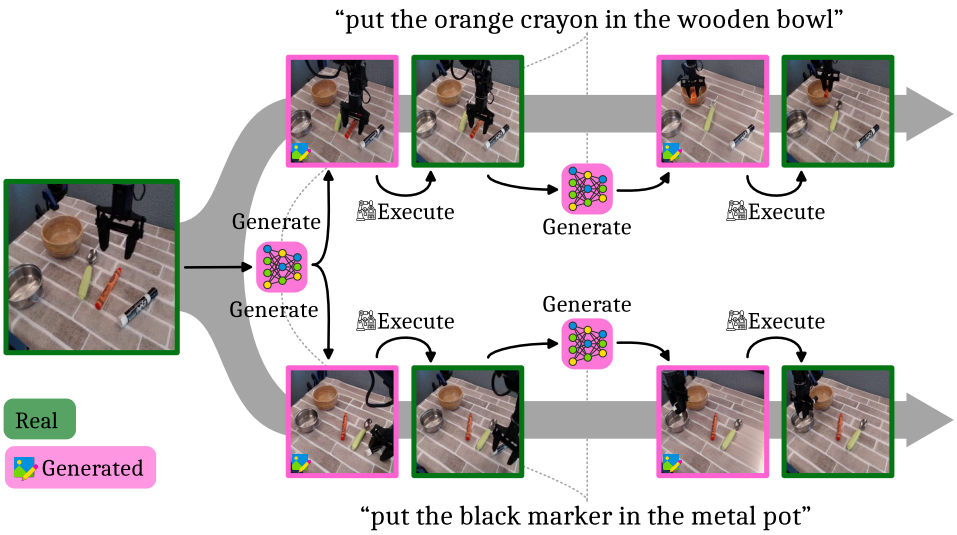


Figure 2: **SuSIE leverages a pre-trained image editing model to generate future image subgoals based on a language commands.** A low-level goal-reaching policy then executes the actions needed to reach each subgoal. Alternating this loop enables us to solve the task.

utilizing frozen or fine-tuned off-the-shelf VLMs (Driess et al., 2023; Radford et al., 2021) on robot data to produce action sequences (Brohan et al., 2023a).

While these approaches do utilize pre-trained models, we find in our experiments, that pre-training using VLMs (e.g., Brohan et al. (2023a)) does not necessarily enhance low-level motor control, in the sense that learned policies often localize the object or move the gripper imprecisely (see Figure 5). On the other hand, our approach is able to incorporate benefits of pre-training in synthesizing subgoals that carefully steer the motion of the low goal-conditioned policy, improving its precision. Also, while our approach can be directly applied in *unstructured* open-world settings, applying GenAug (Chen et al., 2023), MOO (Stone et al., 2023), and ROSIE (Yu et al., 2023b) requires additional information about the scene, such as clean object bounding boxes or 3D object meshes. This significantly restricts their applicability to scenarios where this additional information is not available: for example, GenAug is not applicable in our real-world experiments since 3D object meshes for new target objects are not available. Distinct from our approach for utilizing generative models, other works design representation learning objectives for vision-language pre-training for control (Nair et al., 2022; Ma et al., 2023; Karamcheti et al., 2023; Bhateja et al., 2023), but these methods still need to utilize limited amounts of data from the target task to learn a policy.

The second category of approaches also incorporates semantic information from pre-trained models for planning. Most approaches in this category use pre-trained models to imagine visual (Du et al., 2023; Ajay et al., 2023b) or textual plans (Brohan et al., 2023b; Huang et al., 2022a,b; Liang et al., 2023), which then inform a low-level robot control policy. Low-level policies conditioned on text suffer from a grounding problem, which our approach circumvents entirely since the low-level control policy only observes image-based plans. Perhaps the most related are UniPi (Du et al., 2023) and HiP (Ajay et al., 2023b), which train video models to generate a sequence of frames achieving the target task, and then extract robot actions from an inverse dynamics model. Our approach does *not* attempt to generate full videos (i.e., *all* frames in a rollout), but only the next waypoint that a low-level policy must achieve to solve the commanded task. While this difference might appear small, it has major implications: modeling an entire video puts a very high burden on the generative model, requiring the frames to obey strict physical consistency. Unfortunately, we find that current video models often produce temporally inconsistent frames ("hallucinations"), which only confuse the low-level controller, inhibiting it from completing the task. Indeed, the control evaluations in such prior works often focus on simpler simulated environments. Our method provides more freedom to the low-level controller to handle the physical aspects of the task over a longer time interval while providing higher-level guidance at a level that is suitable to the diffusion model's ability to preserve physical plausibility. In our experiments, we find that our method significantly improves over a reimplementation of UniPi (Du et al., 2023).

Classical model-based RL and planning with no pre-training. The idea behind our approach is also related to several methods in the deep RL literature that do not use pre-trained models and generally do not study language-guided control. For instance, (Hafner et al., 2019; Lee et al., 2020; Yu et al., 2021; Wu et al., 2023; Rafailov et al., 2021; Hafner et al., 2023) train action-conditioned dynamics models and run RL in the model. While our approach also models multi-step dynamics, our model is not conditioned on an action input. Removing the dependency on an action input enables us to de-couple the fine-tuning of the (large) image-editing model from the policy entirely, improving simplicity and time efficiency. APV (Seo et al., 2022) trains an action agnostic dynamics model from videos but fine-tunes it in a loop with the policy with actions, and hence, does not enjoy the above benefits. Finally, these model-based RL methods do not exhibit zero-shot generalization abilities to new tasks, which is an important capability that our method enjoys. Our approach is also related to several video prediction methods (Ebert et al., 2018; Lee & He, 2018; Babaeizadeh et al., 2020; Villegas et al., 2019) but utilizes a better neural network architecture (i.e., diffusion models instead of LSTMs and CNNs). Most related is to our method is hierarchical visual foresight (HVF) (Nair & Finn, 2019): while HVF utilizes MPC to find an action, our approach simply utilizes a goal-reaching policy thereby eliminating the cost of running MPC with large dynamics models.

Our approach is also related to several prior works that utilize generative models for planning in a single-task setting, with no pre-training. Trajectory transformer (TT) (Janner et al., 2021), decision transformer (DT) (Chen et al., 2021), and their extensions condition the policy on the target return or goal. While diffusion-based variants of these methods (Janner et al., 2022; Ajay et al., 2023a) use diffusion models to model long-term rollout distributions over states, actions, and rewards, they still require training data from the target task to learn a policy, unlike our zero-shot planning approach.

C Preliminaries and Problem Statement

We consider the problem setting of language-conditioned robotic control. Specifically, we want a robot to accomplish the task described by a novel language command. We study this problem in the context of learning from a dataset $\mathcal{D}$ of language-labeled robot trajectories, and optionally, an additional dataset, $\mathcal{D}'$ of robot data, which is not annotated any task labels (e.g., play data). Formally, $\mathcal{D} = \{(\tau^1, l_1), (\tau^2, l_2), \cdots, (\tau^N, l_N)\}$, where each rollout τ^i consists of a sequence of scenes (or states) $\mathbf{s}_k^i \in \mathcal{S}$ and actions $\mathbf{a}_k^i \in \mathcal{A}$ that were executed while collecting this data, i.e., $\tau^i = (\mathbf{s}_0^i, \mathbf{a}_0^i, \cdots, \mathbf{s}_k^i, \mathbf{a}_k^i, \cdots)$, following the standard assumptions of a Markov decision process. l_i is a natural language command describing the task accomplished in the trajectory. $\mathcal{D}'$ is organized similarly to $\mathcal{D}$, but does not contain any language annotations l_i. At test time, given a new scene $\mathbf{s}_0^{\text{test}}$ and a new natural language description l^{test} of a task, we evaluate a method in terms of its success rate at accomplishing this task starting from this scene, $\mathbf{s}_0^{\text{test}}$.

D SuSIE: Subgoal Synthesis via Image Editing

Our goal is to utilize semantic information from the Internet to improve language-guided robot control in novel environments, scenes, and objects. How can we do this when models trained on general-purpose Internet data do not provide guidance in selecting low-level actions? Our key insight is that we can still utilize some sort of a pre-trained model for guiding low-level control if we could decouple the robot control problem into two phases: **(i)** imagining subgoals that would need to be attained to succeed at the task, and **(ii)** learning low-level control policies for reaching these generated subgoals. Our method incorporates semantic information from Internet pre-training in phase (i), by fine-tuning a text-guided image-editing model for subgoal generation. Phase (ii) is accomplished via a goal-conditioned policy trained only on robot data. We describe each of these phases below and then summarize the resulting robot controller.

D.1 Phase (i): Synthesizing Subgoals From Image Editing Models

The primary component of our method is a generative model that, given a target task specified in natural language, can guide the low-level controller towards a state that it must try to attain in order to solve the task. One way to accomplish this is to train a generative model to produce an immediate next way-point or subgoal frame. We can then incorporate semantic information from the Internet into our algorithm by initializing this generative model with a suitable pre-trained initialization, followed by fine-tuning it on multi-task, diverse robot data.

What is a good pre-trained initialization for initializing this model? Our intuition is that since accomplishing a task is equivalent to "editing" the pixels of an image of the robot workspace under controls prescribed by the language command, a favorable pre-trained initialization is provided by a language-guided image-editing model. We instantiate our approach with Instruct pix2pix (Brooks et al., 2023), though other image editing models could also be used. Formally, this model is given by $p_\theta(\mathbf{s}_{\text{edited}}|\mathbf{s}_{\text{orig}}, l)$. Then, using the dataset $\mathcal{D}$ of robot trajectories, we fine-tune p_θ on tuples containing a pair of images sampled from a trajectory and the corresponding language annotation: $(\mathbf{s}_{\text{orig}} := \mathbf{s}_i^k, \mathbf{s}_{\text{edited}} := \mathbf{s}_j^k, l_k)$, where $\mathbf{s}_j$ is a state that appears after $\mathbf{s}_i$ ($j > i$). During fine-tuning, we run gradient descent on the following objective, starting from $\theta_0 := \theta_{\text{pre-trained}}$:

$$\min_\theta \quad -\mathbb{E}_{(\tau^k, l_k)\sim\mathcal{D};\ \mathbf{s}_i^k\sim\tau^k;\ j\sim q(j|i)} \left[\log p_\theta\left(\mathbf{s}_j^k|\mathbf{s}_i^k, l_k\right)\right]. \tag{3}$$

We need to choose the distribution q over the time-step j given a state $\mathbf{s}_i^k$ for fine-tuning the image-editing model as in Equation 3. Since we model the next subgoal that the low-level controller should attain, and since the depending upon the task, this subgoal could be arbitrarily close to the original state $\mathbf{s}_i$, we require valid tuples $(\mathbf{s}_i, \mathbf{s}_j, l)$ used for fine-tuning p_θ in Equation 3 to have values of j in a bounded interval around i, specifically we choose $j \in [i, i+k]$, where k is a fixed hyperparameter.

D.2 Phase (ii): Reaching Generated Sub-Goals with Goal-Conditioned Policies

In order to utilize the fine-tuned image-editing model to actually control the robot, we further need to train a low-level controller to actually select suitable robot actions. In this section, we present the design of our low-level controller, followed by a full description our test-time control procedure. Since the image-editing model in SuSIE produces images of future subgoals conditioned on natural language task descriptions, our low-level controller can simply be a language-agnostic goal-reaching policy that aims to reach these generated subgoals.

Training a goal-reaching policy. Our goal-reaching policy is parameterized as $\pi_\phi(\cdot|\mathbf{s}_i, \mathbf{s}_j)$, where $\mathbf{s}_j$ is a future frame that the policy intends to reach, by acting at $\mathbf{s}_i$. At test time, we only need the low-level goal-conditioned policy to be proficient at reaching close-by states that lie within k steps of a given state since the image editing model from phase (i) is also trained to produce subgoals within k steps of any state. To train this policy, we run goal-conditioned behavioral cloning (GCBC) on the robot data, utilized previously in phase (i). In addition, we can also leverage robot data $\mathcal{D}'$ that does not contain language annotations. Formally, our training objective is given by:

$$\max_\phi \quad \mathbb{E}_{\tau^i\sim\mathcal{D}\cup\mathcal{D}';\ (\mathbf{s}_i^k, \mathbf{a}_i^k)\sim\tau^k;\ j\sim q(j|i)} \left[\log \pi_\phi(\mathbf{a}_i^k|\mathbf{s}_i^k, \mathbf{s}_j^k)\right], \tag{4}$$

where $q(j|i)$ is the distribution over future frames that we previously utilized in Equation 3.

Test-time control with π_ϕ and p_θ. Once both the goal-reaching policy π_ϕ and the image editing subgoal generation model p_θ are trained, we utilize them together to solve new manipulation tasks based on user-specified natural language commands. Given a new scene, $\mathbf{s}_0^{\text{test}}$, and a language task description l^{test}, SuSIE attempts to solve the task by iteratively generating subgoals and commanding the low-level goal-reaching policy with these subgoals. At the start, we sample the first subgoal $\hat{\mathbf{s}}_+^{\text{test}} \sim p_\theta(\cdot|\mathbf{s}_0^{\text{test}}, l^{\text{test}})$. Once the subgoal is generated, we then roll out the goal-reaching policy π_ϕ, conditioned on $\hat{\mathbf{s}}_+^{\text{test}}$, for k time-steps, such that each action is chosen according to $\mathbf{a}_j^{\text{test}} \sim \pi_\phi(\cdot|\mathbf{s}_j^{\text{test}}, \hat{\mathbf{s}}_+^{\text{test}})$. After k time steps, given the current image $\mathbf{s}_k^{\text{test}}$, we refresh the subgoal by sampling from the image-editing model again and repeat the process. Note crucially that this recipe does not require that the subgoal $\mathbf{s}^{\text{test}}$ be attained after k steps, as the generative model effectively "replans" a new subgoal based on the current observation. Overall, at test time, we alternate between obtaining a new subgoal from p_θ and commanding the goal-reaching policy to attain this subgoal, until a maximum number of allowed time steps. Pseudocode is provided in Algorithm 2.

D.3 Implementation Details

In Phase (i), we utilize the pre-trained initialization from the InstructPix2Pix model (Brooks et al., 2023), trained to perform language-guided image editing and fine-tune it on our robot dataset. Since the InstructPix2Pix model utilizes a UNet-based diffusion model architecture, we implement Equation 3 using a variational lower bound objective, following the standard recipe for training diffusion models. Our image-editing diffusion model operates on images of size 256×256. The language instructions are encoded with a frozen CLIP encoder (Radford et al., 2021). To ensure that this model

Algorithm 2 SuSIE: Zero-Shot, Test-Time Execution

Require: subgoal model $p_\theta(\mathbf{s}_+|\mathbf{s}_t, l)$, policy $\pi_\phi(\cdot \mid \mathbf{s}_t, \mathbf{s}_+)$, language command l^{test}, max episode length T, goal sampling interval K, initial state $\mathbf{s}_0^{\text{test}}$

$t \leftarrow 0$
while $t \leq T$ **do**
  Sample $\mathbf{s}_+^{\text{test}} \sim p_\theta(\mathbf{s}_+|\mathbf{s}_t^{\text{test}}, l^{\text{test}})$ ▷ Sample a new subgoal every K steps
  for $j = 1$ to k **do**
    Sample $\mathbf{a}_t \sim \pi_\phi\left(\cdot \mid \mathbf{s}_t^{\text{test}}, \mathbf{s}_+^{\text{test}}\right)$ ▷ Predict the action from current state and subgoal
    Execute $\mathbf{a}_t$
    $t \leftarrow t + 1$
  end for
end while

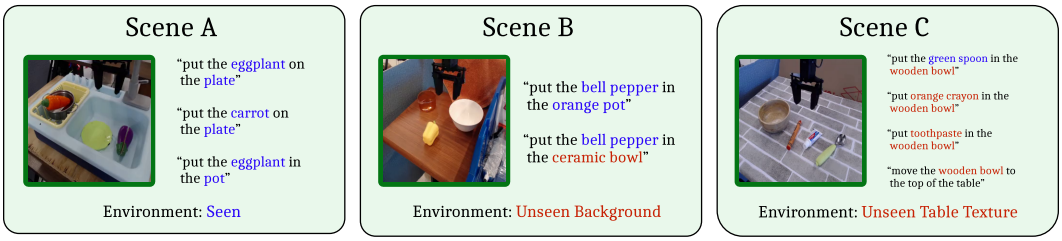


Figure 3: **Real-world experimental setup.** We evaluate our method in 3 real-world scenes. The scenes become progressively more difficult from left to right, due to both an increasing visual departure from the robot training data and an increasingly confounding mixture of both seen and unseen objects.

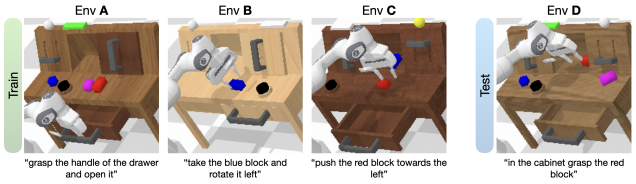
pays attention to the input state and the language command it is conditioned on, we apply classifier-free guidance (Ho & Salimans, 2022) separately to both the language and the image, similarly to InstructPix2Pix. To obtain a robust goal-reaching policy in Phase (ii), we follow the implementation details prescribed by Walke et al. (2023). More details about the training hyperparameters and the architecture of this goal-reaching policy are provided in Appendix G.1.1.

E Experimental Evaluation

The goal of our experiments is to evaluate the efficacy of SuSIE at improving generalization and motor control in open-world robotic manipulation tasks. To this end, our experiments aim to study the following questions: **(1)** Can SuSIE generate plausible subgoals for novel tasks, objects and environments, even those that lie outside of the robot training distribution? **(2)** Are the generated subgoals useful for solving a task specified by a novel language command, in zero-shot?, **(3)** Does SuSIE exhibit an elevated level of precision and dexterity compared to other approaches that do not use subgoals?, and **(4)** How crucial is pre-training on Internet data for attaining zero-shot generalization? To answer these questions, our experiments compare SuSIE to several prior methods including state-of-the-art approaches for training language-conditioned policies that leverage pre-trained vision-language models in a variety of ways.

E.1 Experimental Scenarios and Comparisons

Real-world experimental setup and datasets. We conduct our real-robot experiments on a WidowX250 robot platform. Our robot dataset is BridgeData V2 (Walke et al., 2023), a large and diverse dataset of robotic manipulation behaviors designed for evaluating open-vocabulary instructions. The dataset contains over 60k trajectories, 24 environments, 13 skills, and hundreds of objects. Our evaluations present three different scenarios 3, designed specifically to test the ability of various methods at different levels of open-world generalization: **Scene A:** this scene includes an environment and objects that are well-represented in BridgeData V2; **Scene B:** this scene is situated in an environment with a seen tabletop but a novel background and distractors, where the robot must move a seen object (bell pepper) into a choice of seen container (orange pot) or unseen container (ceramic bowl); and **Scene C:** this scene includes a table texture unlike anything in BridgeData V2 and requires manipulating both seen and unseen objects. We expect Scene C to be the hardest since the robot needs to carefully ground the language command to identify the correct object while resisting its affinity for an object that is well-represented in the data (the spoon).



Simulation tasks. We run our simulation experiments in CALVIN (Mees et al., 2022b), a benchmark for long horizon, language-conditioned manipulation. CALVIN consists of four simulated environments, A, B, C, D, and each environment comes with a dataset of human-collected play trajectories. Approximately 35% of these rollouts are annotated with language. Each environment consists of a Franka Emika Panda robot arm positioned next to a desk with various manipulatable objects, including a drawer, sliding cabinet, light switch, and various colored blocks. Environments are differentiated by the positions of these objects and their textures. With this benchmark, we study the most challenging zero-shot multi-environment scenario: training on A, B, and C, and testing on D. We follow the evaluation protocol from Mees et al. (2022b). During evaluation, a policy given a fixed number of timesteps (default 360) to complete a chain of five language instructions.

Comparisons. Our experiments cover methods that utilize pre-trained models of vision and language in language-guided robot control in a variety of ways. While there are several prior methods that tackle language-based robotic control as we discuss in Section B, in our experiments, we choose to compare to a representative subset of these prior methods to maximally cover the possible set of comparisons. We compare to **(a)** RT-2 (Brohan et al., 2023a) which is one of the most recent works utilizing a pre-trained VLM for initializing the robot policy (specifically, RT-2-X (Anonymous), which was also trained and evaluated on BridgeData V2), generalizing prior work (Shridhar et al., 2022); **(b)** MOO (Stone et al., 2023), which utilizes pre-trained object detectors to obtain bounding box information for the policy and then trains a language-conditioned behavioral cloning policy (denoted as "LCBC/MOO"); and **(c)** UniPi (Du et al., 2023), which trains an entire language-conditioned video prediction model starting from a pre-trained video initialization. Since the original UniPi model utilized proprietary pre-trained initializations that are not available publically, we re-implemented this method using our own video model following the guidelines in Du et al. (2023), but were unable to obtain high-quality generations. In our simulation experiments though, we also evaluate another reimplementation of UniPi, using a video diffusion model trained by concurrent work (Ajay et al., 2023b). We present details for MOO and UniPi baselines in Appendix G.2. Finally, we remark that while we did try to apply GenAug (Chen et al., 2023) as a representative semantic augmentation approach in our real-world experiments, we were not able to obtain 3D mesh predictions for objects in Bridgedata V2, needed for this approach.

We also compare to language-conditioned behavioral cloning ("LCBC") (Walke et al., 2023), trained to produce actions conditioned on an embedding of the natural language task description (Walke et al., 2023); and an oracle goal-conditioned behavioral cloning ("GCBC oracle") approach for tasks that require manipulating objects previously seen in the robot data. We observed that in Scene A, simple LCBC outperforms MOO. However, in Scenes B and C, which include tasks with unseen objects, MOO is crucial for achieving non-zero success. Hence, we report LCBC in Scene A and MOO in Scenes B and C. In simulation, we also compare to additional methods previously studied on the CALVIN benchmark. These include methods that explicitly tackle long-horizon language-based control on CALVIN such as multi-context imitation (MCIL) (Lynch & Sermanet, 2020), hierarchical universal language-conditioned policy (HULC) (Mees et al., 2022a), and improved variants of HULC (Ge et al., 2023). We also compare to other state-of-the-art methods from Ge et al. (2023) that employ an identical training and evaluation protocol as our experiments, namely MdetrLC (Kamath et al., 2021), and AugLC (Pashevich et al., 2019).

E.2 Can SuSIE Generate Plausible and Meaningful Subgoals?

To answer question (1), we start by presenting qualitative examples of intermediate subgoals generated by the SuSIE image-editing model in Figure 4. Even on previously unseen trajectories and language commands, the model is able to produce visually high-quality and useful subgoals involving the gripper grasping and moving objects. This is nontrivial since it requires the model to have not only the *semantic knowledge* to detect which pixels in the image correspond to a given object, but also an understanding of *dynamics* to predict how to move and rotate the gripper to grasp it.

E.3 Is the Synthesized Subgoal Useful for Completing New Commands?

Simulation results. We present performance for SuSIE and other comparisons in Table 1, in terms of success rates (out of 1.0) for completing each language instruction in the chain. Observe that

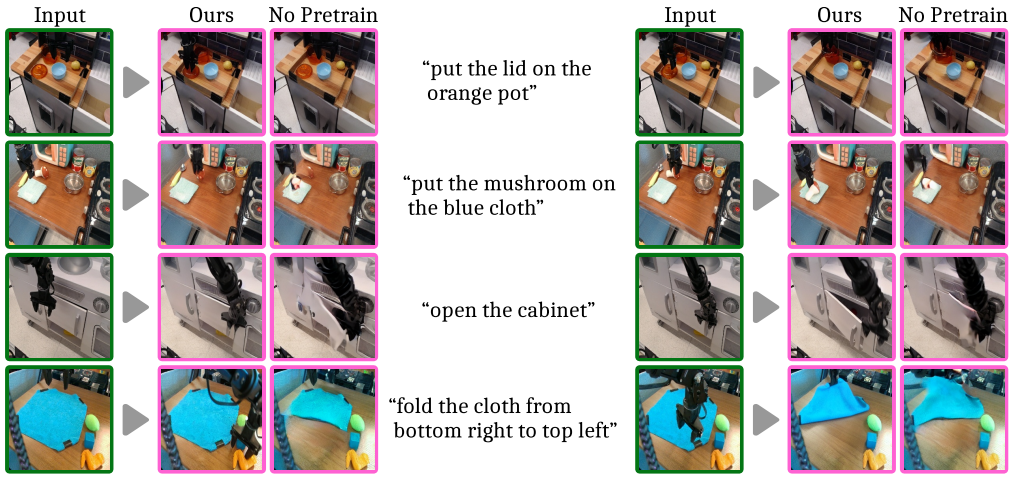


Figure 4: **Examples of subgoals synthesizd by SuSIE.** A Comparison between the pre-trained diffusion model initialized from InstructPix2Pix **(Ours)** and random initialization on BridgeData. Each row is a trajectory from a holdout in-distribution validation set, where the objects and environments are all seen but the particular trajectory and language label are not. The fine-tuned model consistently generates better subgoals.

SuSIE is able to complete instructions with a significantly higher success rate than LCBC, outperforming prior methods on this benchmark, including both the reimplementations of the closest prior approach, UniPi. Concretely, we observe more than about 20% improvement in the success rates for completing the first and second language tasks in the chain, and approximately 10% improvement for the remaining tasks. This indicates that SuSIE is able to produce useful subgoals that enable the low-level policy to accomplish tasks in this novel environment.

Real-world results. We present performance of real-world evaluations in Table 2. Observe that in Scene A, SuSIE achieves the highest success rate on all three tasks, attaining an average success rate of 73% which improves over RT-2-X by **69%**. In Scene B, SuSIE again outperforms other prior approaches on the two tasks, successfully grounding both the novel ceramic bowl and the previously seen orange pot. In the most challenging Scene C (unseen domain, unseen objects), SuSIE attains a success rate of 45%, outperforming MOO by about **260%**. However, RT-2-X outperforms SuSIE in this scene. We believe that the superior performance of RT-2-X compared to SuSIE in Scene C is because it is a much larger 55B parameter model, initializes from a proprietary VLM, and is trained on much more data — including BridgeData V2, but also a vast quantity of additional tabletop manipulation. These differences in the amount of data and parameters put our method, which only utilizes BridgeData V2, at quite an unfair advantage against RT-2-X. Nevertheless, SuSIE is still able to recognize the novel objects and attain a high absolute success rate of 45%.

E.4 Does SuSIE Improve Precision and Low-Level Skill Execution?

Our real-world and simulated results clearly demonstrate the efficacy of SuSIE in executing novel language commands in a variety of scenarios. In this section, we visualize some evaluation rollouts from our experiments in Scene A to understand if SuSIE works merely because it enhances the generalization of the policy to semantic changes in the visual observation or if it actually does improve the precision of the low-level control by commanding meaningful subgoals. Observe in Figure 5 that the RT-2-X policy often produces actions that fail to precisely orient the gripper around the target object or close the gripper early. In contrast, policy executions obtained via SuSIE are more precise, and execute actions that attempt to match the gripper and object positions to the generated subgoal, allowing the policy to succeed at the task.

To understand the contribution of the subgoal prediction towards improved precision, we also evaluate an oracle GCBC policy on a subset of tasks. This policy is trained on identical robot data as SuSIE; however, we at test time we command the policy with a real image of the completed task, which our method does not require. Observe that even then this GCBC oracle fails to accomplish the task due to issues with imprecise object localization and untimely gripper closing. Corroborated by numerical results in Table 3, these experiments validate our claim that utilizing subgoal prediction is crucial for enabling precise low-level skill execution and control.

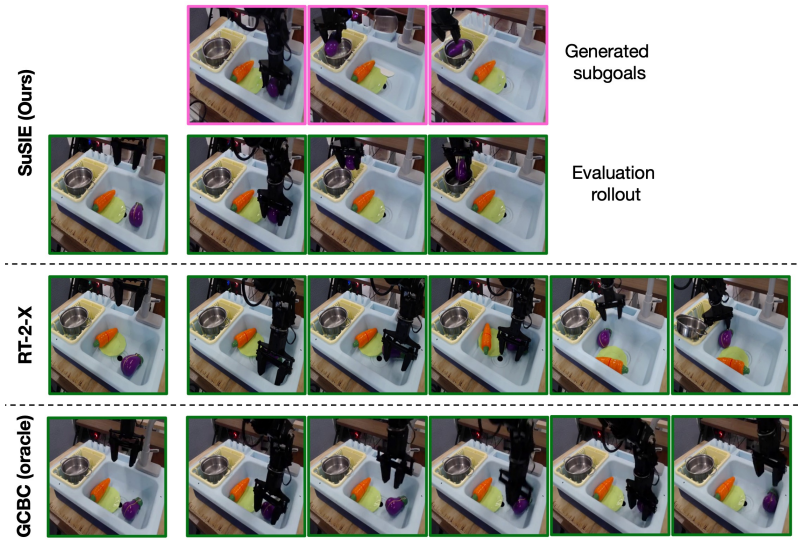


Figure 5: **Visualizing rollouts from SuSIE, RT-2-X, and oracle GCBC.** While RT-2-X and oracle GCBC often fail to precisely localize or grasp the object, generated subgoals from the image editing model in SuSIE guide the low-level controller precisely, improving low-level skill execution with novel language commands.

E.5 Is Pre-Training on Internet Data Crucial for Zero-Shot Generalization?

Finally, we conduct an experiment to understand if pre-training is crucial for generating meaningful subgoals. We train a second image editing model without InstructPix2Pix initialization, but using the same UNet architecture, image autoencoder, and text encoder as InstructPix2Pix. Observe in Figure 4 that the pre-trained model consistently generates superior subgoals.

F Discussion and Future Work

We presented a method for robotic control from language instructions that uses pre-training to generate subgoals to guide low-level goal-conditioned policy, which is unaware of language. The subgoals are generated by an image-editing diffusion model fine-tuned on robot data. This system improves both zero-shot generalization to new objects, and the precision of the overall policy, because the subgoal model incorporates semantic benefits from pre-training and commands the low-level policy to reach more meaningful subgoals. Our experiments show that SuSIE improves over prior techniques on the CALVIN benchmark and attains good performance in three different scenes for a real-world manipulation task, outperforming language-conditioned behavioral cloning, and often outperforming the state-of-the-art, instruction-following approach, RT-2-X, that is trained on more than an order of magnitude more robot data.

Our method is simple and provides good performance, but it does have limitations that suggest promising directions for future work. For instance, the diffusion model and the low-level policy are trained separately indicating that the diffusion model itself is also unaware of the capabilities of the low-level policy — it is trained on the same dataset, but assumes that anything that is reachable in the dataset can also be reached by the policy. We hypothesize that performance can be improved by making the diffusion model aware of the low-level policy's capabilities. More broadly, we found the performance of our method to often be bottlenecked by the performance of the low-level policy, suggesting that addressing either of these limitations might lead to a more performant method for importing Internet-scale knowledge into robotic manipulation.

G Further Details of Implementation

We provide implementation details for SuSIE and the baselines.

G.1 SuSIE implementation details

G.1.1 Goal-reaching policy

We use a diffusion model for our goal-reaching policy since recent work has shown that diffusion-based policies can better capture multi-modality in robot data (Chi et al., 2023; Hansen-Estruch et al., 2023), leading to improved performance across a variety of tasks. In our implementation (which follows Walke et al. (2023)), the observation and goal image are stacked channel-wise before being passed into a ResNet-50 image encoder. This image encoding is used to condition a diffusion process that models the action distribution. We use the DDPM (Denoising Diffusion Probabilistic Models) objective as introduced by Ho et al. (2020). The diffusion process uses an MLP with 3 256-unit layers and residual connections. Following Chi et al. (2023), rather than predicting a single action, we predict a sequence of k actions to encourage temporal consistency. We use an action sequence length of $k = 4$. We use the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 3e-4 and a linear warmup schedule with 2000 steps. We augment the observation and goal with random crops, random resizing, and color jitter. During training, the goal associated with an observation is selected by uniformly sampling an observation from a window of future timesteps in the trajectory. Specifically, we sample a goal from 0-20 steps in the future.

At test time, we have several options for how to predict and execute action sequences. Chi et al. (2023) use receding horizon control, sampling k-length action sequences and only executing some of the actions before sampling a new sequence. This strategy can make the policy more reactive. However we found that the robot behavior was quite jerky as the policy switched between different modes in the action distribution with each sample. Instead, we use a temporal ensembling strategy similar to Zhao et al. (2023). We predict a new k-length action sequence at each timestep and execute a weighted average of the last k predictions.

G.2 Baseline implementation details

G.2.1 Language-conditioned behavior cloning (LCBC)

We use the language-conditioned behavior cloning method from Walke et al. (2023) and Myers et al. (2023). The instruction is encoded using the MUSE sentence embedding Yang et al. (2019), then the image observation is encoded using a ResNet-50 with FiLM conditioning on the language encoding Perez et al. (2017). The output is passed into a fully connected policy network with 3 256-unit layers to produce the action. We use the Adam optimizer Kingma & Ba (2015) with a learning rate of 3e-4 and a linear warmup schedule with 2000 steps. We augment the observation and goal with random crops, random resizing, and color jitter.

G.2.2 UniPi

UniPi (Du et al., 2023) trains a video diffusion model, $p_\theta\left(\tau|\mathbf{s}_0, l\right))$ to generate a sequence of frames given a language command and an initial frame. The original paper employs the model architecture from Imagen Video (Ho et al., 2022a,b). To achieve higher resolution and longer videos for their real-world results, the authors leverage a 1.7B 3D U-Net and four pre-trained super-resolution models from Imagen Video, with 1.7B, 1.7B, 1.4B, and 1.2B parameters, respectively. Since the original models and codes are not publicly available, we tried to replicate their approach in two different ways.

UniPi (ours). We implemented a 3D U-Net video diffusion model, following Ho et al. (2022b,a), combining UniPi's first-frame conditioning. Due to limited computes, we did not train spatial/temporal super-resolution models; instead, we trained a 3D U-Net-based diffusion model to directly generate images with a resolution of 128×128. The model includes 4 residual blocks, with (input channels, output channels) as follows: $(64, 64)$, $(64, 128)$, $(128, 256)$, and $(256, 640)$. The model is trained to produce the trajectory with a fixed horizon of 10 frames $\tau_t = \{s_t, s_{t+1}, \ldots, s_{t+9}\}$ conditioned on the current frame s_t and language command. We used a frozen pre-trained CLIP (Radford et al., 2021) encoder to obtain the language embeddings.

UniPi (HIP, Ajay et al. (2023b)) For the second approach, we followed the UniPi replication in Ajay et al. (2023b). We trained a latent video diffusion model from PVDM (Yu et al., 2023a), building upon the codebase `https://github.com/sihyun-yu/PVDM` where we added first

frame conditioning. We first trained the video autoencoder to project video of size $16 \times 128 \times 128$ into latent representation, followed by training a PVDM-L model that uses a 2D U-Net architecture. We used a Flan-T5- Base (Chung et al., 2022) encoder to obtain the language embeddings.

Data and training details. To incorporate knowledge from internet data into video models, we utilize Ego4D (Grauman et al., 2022), a large-scale human egocentric video dataset with language annotations. For UniPi (ours), we first pre-trained the video model on Ego4D for 270K steps, and fine-tuned it on the robotics dataset, CALVIN for the simulation and BridgeData v2 for the real world, for additional 200K steps. We use a batch size of 4 during the training. For UniPi (HIP), we jointly trained a single model on all Ego4D, BridgeData v2, and CALVIN dataset at the same time. The autoencoder was trained for 85K steps, and the PVDM-L model was trained for 200K steps. We use a batch size of 8 during the training.

Inverse model and test time control. To extract actions from generated videos, we trained an inverse dynamics model $\pi_\phi\left(\cdot \mid \mathbf{s}_t, \mathbf{s}_{t+1}\right)$ to predict the action from two adjacent frames. We employed the same architecture as our GCBC policy described in Section D.2 and set the goal horizon k to 1. During test time, given the current observation s_t and the language command l, we synthesize H image frames from the video model and apply the inverse dynamics model to obtain the corresponding $H-1$ actions. The predicted actions are executed, and we generate a new video from s_{t+H-1} and repeat the process until it reaches the maximum episode step.

Generated videos. While the quality of the video model trained on the simulation dataset is good enough for solving the tasks on the CALVIN benchmark as shown in Table 1, we found that it is nontrivial to obtain a high-quality generation for the real-world dataset. Additionally, sampling the video model of UniPi to rollout a real robot is extremely time-consuming. Therefore, we evaluated UniPi only in simulations.

G.2.3 MOO

MOO (Stone et al., 2023) utilizes a mask to represent the target objects and incorporates it as an additional channel in the observation. Specifically, they train a language-conditioned policy that takes a 4-channel image and a language command as inputs. To acquire the mask for target objects, the Owl-ViT (Minderer et al., 2022) detector is employed. This detector is an open-vocabulary object detection model, pre-trained on internet-scale datasets, and it is used to extract the bounding boxes of the objects of interest from the image. For tasks like "move X to Y," MOO calculates the bounding box for X, representing the object of interest, and Y, indicating the target place. A mask is then created where the pixel at the center of the predicted bounding box is assigned a value of 1.0 for X and 0.5 for Y.

Extracting object entities from BridgeData V2 language annnotaions. In order to obtain the mask, it is necessary to extract the entities corresponding to the object of interest, denoted as X, and the target place, Y, from the language command. In MOO's original paper, the authors assume that the language in their dataset is structured in a way that facilitates the easy separation of X and Y. Specifically, they employ a dataset that exclusively consists of language annotations such as "pick X," "move X near Y," "knock X over," "place X upright," and "place X into Y."

Given that the language annotations in BridgeData v2 are diverse and unstructured, it is challenging to naively extract X and Y. We utilized the the API of OpenAI's `gpt-3.5-turbo-instruct` model to extract the object of interest and the target place (if any) from the language annotations, and input them into Owl-ViT to create masks. We then train a mask conditioned LCBC policy using the same architecture as described in Section G.2.1. Following the original work, we removed X and Y from the prompt and replaced the word X with "object of interest" and the word Y with "target place". For example, given a language prompt "put the eggplant in the pot", we use a modified prompt "put object of interest in target place" as the input to the policy during both training and test time.

Test time. During test time, we use oracle masks annotated from the initial camera observations of each test trial. To enable this, we build a simple interface on the robot machine, allowing the tester to create the masks by clicking on the initial camera image at the beginning of each test trial.