

GHIL-Glue: Hierarchical Control with Filtered Subgoal Images

Anonymous Author(s)

Affiliation

Address

email

1 **Abstract:** Image and video generative models that are pre-trained on Internet-
2 scale data can increase the generalization capacity of robot learning systems.
3 These models can function as high-level planners, generating intermediate sub-
4 goals for low-level goal-conditioned policies to reach. However, the performance
5 of these systems can be bottlenecked by the interface between generative mod-
6 els and low-level controllers. Generative models may predict photorealistic yet
7 physically infeasible frames. Low-level policies may also be sensitive to subtle
8 visual artifacts in generated goal images. This paper addresses these facets
9 of generalization, providing an interface to “glue together” language-conditioned
10 image or video prediction models with low-level goal-conditioned policies. Our
11 method, Generative Hierarchical Imitation Learning-Glue (GHIL-Glue), filters
12 out subgoals that do not lead to task progress and improves the robustness of goal-
13 conditioned policies to generated subgoals with harmful visual artifacts. GHIL-
14 Glue achieves a new state-of-the-art on the CALVIN simulation benchmark for
15 policies using observations from a single RGB camera. GHIL-Glue also outper-
16 forms other generalist robot policies across 3/4 language-conditioned manipula-
17 tion tasks testing zero-shot generalization on a physical robot. Additional details
18 are available at <https://generative-hierarchical-glue.github.io>.

19 **Keywords:** Hierarchical Imitation Learning, Image Generation, Video Prediction

20 1 Introduction

21 As Internet-scale foundation models achieve success in computer vision and natural language pro-
22 cessing, a central question arises for robot learning: how can Internet-scale models enable embodied
23 behavior generalization? While one approach is to collect increasingly large action-labeled robot
24 manipulation training datasets [1, 2, 3], video datasets (without actions) from the Internet are vastly
25 larger. However, while videos may be useful for inferring the steps in a task, such as how the objects
26 should be moved, or which parts of an object to manipulate (e.g., grabbing a cup by the handle), they
27 are less useful for learning details about low-level control. For example, it is difficult to infer the
28 actions for controlling a robot’s fingers from videos of humans performing manipulation tasks. One
29 promising solution to this challenge is to employ a hierarchical approach [4, 5]: infer high-level sub-
30 goal images using models trained on Internet-scale videos, and then fill in the fine-grained motions
31 with low-level policies trained on robot data (see appendix A for a discussion of related work).

32 While this general approach has seen success in prior robotic manipulation work [6, 4, 7, 5, 8, 9],
33 the interface between the high-level planner generating subgoals and the low-level policy that must
34 reach these subgoals can be brittle. First, generative models may occasionally sample subgoals that
35 do not progress towards completing a given language instruction. If one such “off-task” subgoal is
36 followed, it can have a compounding errors effect, leading to subsequent subgoals being increasingly
37 “off-task.” Second, even if the generated subgoals lead to task progress, they can contain subtle
38 visual artifacts that degrade the performance of a naively trained low-level policy.



Figure 1: GHIL-Glue. We consider language-conditioned image and video prediction models that can generate multiple subgoals. GHIL-Glue has two components: augmentation de-synchronization (top) and subgoal filtering (bottom). **Subgoal filtering:** We train a classifier to identify which subgoal is most likely to progress towards completing the language instruction. This subgoal and the image observation are then passed to the low-level policy to choose a robot action. **Augmentation de-synchronization:** The distribution shift between subgoals sampled from the robot dataset during training and those sampled from the generative model during inference can degrade low-level policy and subgoal classifier performance. To robustify the low-level policy and subgoal classifier to artifacts in generated subgoals, we explicitly de-synchronize the image-augmentations applied to the current state (State Aug) and the sampled goal (Subgoal Aug).

39 To address these issues, we propose Generative Hierarchical Imitation Learning-Glue (GHIL-Glue)
40 (fig. 1), a method to *robustly* “Glue” together image or video generative models to a low-level robotic
41 control policy. **First**, we filter out “off-task” subgoals that are physically inconsistent with the com-
42 manded language instruction. We do this by training a subgoal classifier to predict the likelihood
43 of the transition between the current state and a given subgoal resulting in progress towards com-
44 pleting the provided language instruction. We then sample a number of candidate subgoals from the
45 generative model and choose the subgoal with the highest classifier ranking. **Second**, we identify
46 a simple yet non-obvious data augmentation practice to robustify the low-level policy and subgoal
47 classifier to visual artifacts in the generated subgoals. While image augmentations are ubiquitous in
48 robot learning methods, our key finding is that the standard way of applying image augmentations
49 does not make low-level policies robust to visual artifacts in generated subgoal images. Experiments
50 on the CALVIN [10] simulation benchmark and four language-conditioned tasks on the Bridge V2
51 physical robot platform [11] suggest that GHIL-Glue improves upon prior SOTA methods for zero-
52 shot generalization while adding minimal additional algorithmic complexity.

53 2 GHIL-Glue

54 2.1 Subgoal Filtering

55 The image and video generative models we consider are first pre-trained on general Internet-scale
56 image and video data, and then fine-tuned on a modest amount of robot data (see appendix B for
57 a detailed description of the problem setting we consider). A common failure mode we observe
58 across different models is that, while executing a task, the model begins to go “off-task,” generating
59 subgoals that are consistent with the current image observation but that do not progress towards com-
60 pleting the language instruction l . We hypothesize that this is due to the distribution shift between
61 the Internet image and video pre-training data and the robot data they are fine-tuned on.

62 To address this challenge, we train a subgoal classifier $f_\theta(s, g, l)$ on a language-conditioned dataset
63 of trajectories \mathcal{D}_l that predicts the probability that the transition between the current image ob-
64 servation s and the next subgoal g makes progress towards completing language instruction l .
65 During training, we sample positive examples of state-goal transitions for l from the set of tra-
66 jectories that successfully complete the instruction. We construct negative examples in the fol-
67 lowing three ways: **1) Wrong Instruction:** (s, g, l') where l' is sampled from a different transi-
68 tion than s and g , **2) Wrong Goal Image:** (s, g', l) where g' is sampled from a different transi-
69 tion than s and l , and **3) Reverse Direction:** (g, s, l) , where the order of the current image
70 observation and the subgoal image have been switched. We refer to this dataset of negative ex-
71 amples constructed from \mathcal{D}_l as \mathcal{D}_l^- . We then train the subgoal classifier by minimizing the bi-
72 nary cross entropy loss between the positive examples and the constructed negative examples:

73 $\mathcal{J}(\theta) = \mathbb{E}_{(s,g,l) \sim \mathcal{D}_l} [\log(f_\theta(s, g, l))] + \mathbb{E}_{(s^-, g^-, l^-) \sim \mathcal{D}_l^-} [\log(1 - f_\theta(s^-, g^-, l^-))]$. At inference,
74 given a set of K subgoals predicted by the image or video model, GHIL-Glue selects the subgoal
75 with the highest classifier ranking to the low-level policy for conditioning.

76 2.2 Image Augmentation De-Synchronization

77 For both the low-level goal-conditioned policy and the subgoal classifier, each training sample in-
78 cludes two images: the current state s and the corresponding goal g . Applying image augmentation
79 procedures during training is a standard approach in image-based robot learning methods [12] to
80 improve the robustness of learned models to distribution shifts between their training and evaluation
81 domains. Standard practice is to sample augmentation parameters $\hat{\phi}$ and apply them to all images
82 in a given training sample [4, 13], which corresponds to applying the same $\hat{\phi}$ to both s and g . In
83 a non-hierarchical policy setting, this makes sense, because at inference time s and g will both be
84 sampled from the camera observations of the current environment instantiation. However, when
85 using an image or video prediction model for subgoal generation, at inference time the observations
86 will come from the environment, but the goals will be generated by the image or video prediction
87 model. There will often be differences in the visual artifacts between a camera observation s and the
88 corresponding generated subgoal image g , such as differences in color, contrast, blurriness, and the
89 shapes of objects, which can degrade the performance of low-level policies and subgoal classifiers.
90 To encourage robustness to this distribution shift, we sample separate augmentation parameters for
91 s and g , denoted by $\hat{\phi}_s$ and $\hat{\phi}_g$ (i.e., we de-synchronize the image augmentations applied to s and
92 g). Concretely, for each s and g pair sampled during training, a different random crop, brightness,
93 contrast, saturation, and hue shift are applied to s than are applied to g . This forces the low-level
94 policy and the subgoal classifier to be robust to differences in visual artifacts between s and g . See
95 appendix C for additional discussion of image augmentation de-synchronization.

96 3 Experiments

97 3.1 Experimental Domains

98 **Simulation Experiment Setup:** Simulation experiments are performed in the CALVIN [10] bench-
99 mark, which focuses on long-horizon language-conditioned robot manipulation. We follow the same
100 protocol as in [4], and train on data from three environments (A, B, and C) and test policies on a
101 fully unseen environment (D). The held-out environment (D) contains unseen desk and object colors,
102 positions, and shapes. See appendix D for a visualization of the CALVIN environment.

103 **Physical Experiment Setup:** Physical experiments are performed with the Bridge V2 [11] ex-
104 periment setup with a WidowX250 robot. We use the same datasets as in [4] for training both the
105 high-level image prediction model and the low-level goal-conditioned policy. The Bridge V2 dataset
106 contains 45K language-annotated trajectories, which are used for the language-labeled robot dataset
107 $\mathcal{D}_{l,a}$. The remaining 15K trajectories are used for the action-only dataset \mathcal{D}_a . As in [4], we use a
108 filtered version of the Something-Something V2 dataset [14] with the same filtering scheme as in [4]
109 (resulting in 75K video clips) as our video-only dataset \mathcal{D}_l . We test our policies on four tasks on four
110 different cluttered table top scenes (fig. 2) on the Bridge V2 physical robot platform. These environ-
111 ments require generalizing to novel scenes, with novel objects, and with novel language commands
112 that are not seen in the Bridge V2 dataset. See appendix D for visualizations of the evaluation set-up.

113 3.2 Comparison Algorithms

114 We study the impact of applying GHIL-Glue to two SOTA hierarchical imitation learning algo-
115 rithms: SuSIE [4] and UniPi [5]. We use either 4 or 8 candidate subgoals for subgoal filtering
116 (see appendix J for details). We also compare GHIL-Glue to a flat language-conditioned diffu-
117 sion policy (LCBC Diffusion Policy). Finally, we consider ablations where we separately study the
118 impact of each of our proposed contributions: subgoal filtering (section 2.1) and de-synchronizing

119 augmentations (section 2.2). For physical experiments, we additionally consider a comparison to
 120 OpenVLA [15], which is trained on the Open X-Embodiment dataset [2] (which includes the Bridge
 121 V2 dataset). See appendix E for a detailed description of each of these algorithms.

122 3.3 Experimental Results

Method	Tasks completed in a row					
	1	2	3	4	5	Avg. Len.
LCBC Diffusion Policy	68.5%	43.0%	22.5%	11.0%	6.8%	1.52
SuSIE [4]	89.8%	75.0%	57.5%	41.8%	29.8%	2.94
GHIL-Glue (SuSIE) - Aug De-sync Only	95.2%	84.0%	69.5%	56.0%	46.2%	3.51
GHIL-Glue (SuSIE) - Subgoal Filtering Only	88.5%	75.5%	56.2%	43.0%	32.5%	2.96
GHIL-Glue (SuSIE)	95.2%	88.5%	73.2%	62.5%	49.8%	3.69
UniPi [5]	56.8%	28.3%	12.0%	3.5%	1.5%	1.02
GHIL-Glue (UniPi) - Aug De-sync Only	60.2%	29.5%	12.5%	5.5%	1.8%	1.1
GHIL-Glue (UniPi) - Subgoal Filtering Only	69.5%	40.0%	15.8%	6.5%	4.2%	1.36
GHIL-Glue (UniPi)	75.2%	44.8%	19.7%	11.2%	5.5%	1.56

Table 1: CALVIN: Simulation Results. Success rates on the validation tasks from the held-out D environment of the CALVIN zero-shot generalization challenge averaged across 4 random seeds. Applying GHIL-Glue to SuSIE and UniPi significantly improves performance over their respective base methods. GHIL-Glue (SuSIE) significantly outperforms all other methods, achieving a new state-of-the-art on the CALVIN benchmark for policies using observations from a single RGB camera.

	Task	OpenVLA [15]	SuSIE [4]	GHIL-Glue (SuSIE)
Scene A	Put Sushi On Towel	22/30	19/30	28/30
Scene B	Put Red Bell Pepper in Bowl	14/30	12/30	16/30
Scene C	Open Drawer	23/30	19/30	22/30
Scene D	Put Sushi in Bowl	15/30	15/30	18/30

Table 2: Bridge V2 Physical Experiments Results. Success rates across four tasks on four physical robot scenes (pictured in fig. 2) that test zero-shot generalization to novel objects, novel language commands, and novel scene configurations. GHIL-Glue applied to SuSIE outperforms SuSIE across all tasks and outperforms OpenVLA on 3 out of 4 tasks.

123 **Simulation Experiments:** We present results on the CALVIN benchmark in table 1. Applying
 124 GHIL-Glue yields significant improvements for SuSIE and UniPi, increasing the average successful
 125 task sequence length from **2.94** to **3.69** for SuSIE and from **1.02** to **1.56** for UniPi. **GHIL-Glue**
 126 (**SuSIE**) achieves a new SOTA on CALVIN for policies that use single RGB camera observations.
 127 See appendix F for additional discussion of results.

128 **Physical Experiments:** We present results (table 2) comparing GHIL-Glue (SuSIE) to OpenVLA
 129 and SuSIE across four environments on the Bridge V2 robot platform that require interacting with
 130 a number of objects on a cluttered table (fig. 2). GHIL-Glue applied to SuSIE outperforms SuSIE
 131 across all tasks and outperforms OpenVLA, a 7-billion parameter SOTA VLA, on 3 out of 4 tasks.
 132 Significantly, the baseline SuSIE implementation does not outperform OpenVLA on a single task,
 133 whereas **GHIL-Glue (SuSIE) outperforms OpenVLA on 3 out of 4 tasks**, demonstrating that hi-
 134 erarchical goal conditioned architectures with well-tuned interfaces between the high and low-level
 135 policies can outperform SOTA VLA methods on zero-shot generalization tasks. See appendix F for
 136 additional discussion of results and appendix I for qualitative analysis of success and failure cases.

137 4 Conclusion

138 We present GHIL-Glue, a method for better aligning image and video prediction models and low-
 139 level control policies for hierarchical imitation learning. Our key insight is that while image and
 140 video foundation models can generate highly realistic subgoals for goal-conditioned policy learn-
 141 ing, when generalizing to novel environments, the generated images are prone to containing visual
 142 artifacts and can be inconsistent with the task the robot is commanded to perform. GHIL-Glue pro-
 143 vides two simple ideas to address these challenges, significantly improving zero-shot generalization
 144 performance over prior work in the CALVIN simulation benchmark and in physical experiments.

145 **References**

- 146 [1] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and
147 C. Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning (CoRL)*,
148 2019.
- 149 [2] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee,
150 A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan,
151 A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi,
152 A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid,
153 B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn,
154 C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu,
155 D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalash-
156 nikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp,
157 G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn,
158 G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Fu-
159 ruta, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra,
160 J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu,
161 J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério,
162 J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao,
163 K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund,
164 K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana,
165 K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto,
166 L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel,
167 M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang,
168 M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess,
169 N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiuallah, O. Mees, O. Kroemer,
170 O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano,
171 P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi,
172 R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Men-
173 donca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore,
174 S. Bahl, S. Dass, S. Sonawani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist,
175 S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park,
176 S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu,
177 T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Vanhoucke,
178 W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu,
179 X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu,
180 Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu,
181 Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang,
182 Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. 2024.
- 183 [3] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany,
184 M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma,
185 P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park,
186 I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mer-
187 cat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe,
188 T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen,
189 T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson,
190 C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen,
191 A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang,
192 P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Ja-
193 yaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu,
194 M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot
195 manipulation dataset. 2024.

- 196 [4] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine. Zero-
197 shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint*
198 *arXiv:2310.10639*, 2023.
- 199 [5] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel.
200 Learning universal policies via text-guided video generation. *Advances in Neural Information*
201 *Processing Systems*, 36, 2024.
- 202 [6] I. Kapelyukh, V. Vosylius, and E. Johns. Dall-e-bot: Introducing web-scale diffusion models
203 to robotics. *IEEE Robotics and Automation Letters*, 2023.
- 204 [7] Y. Du, M. Yang, P. Florence, F. Xia, A. Wahid, B. Ichter, P. Sermanet, T. Yu, P. Abbeel, J. B.
205 Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023.
- 206 [8] A. Ajay, S. Han, Y. Du, S. Li, A. Gupta, T. Jaakkola, J. Tenenbaum, L. Kaelbling, A. Srivastava,
207 and P. Agrawal. Compositional foundation models for hierarchical planning. *Advances in*
208 *Neural Information Processing Systems*, 36, 2024.
- 209 [9] J. Gao, K. Hu, G. Xu, and H. Xu. Can pre-trained text-to-image models generate visual goals
210 for reinforcement learning? *Advances in Neural Information Processing Systems*, 36, 2024.
- 211 [10] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-
212 conditioned policy learning for long-horizon robot manipulation tasks. In *IEEE Robotics and*
213 *Automation Letters (RAL)*, 2021.
- 214 [11] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch,
215 Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset
216 for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.
- 217 [12] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for
218 transferring deep neural networks from simulation to the real world. *International Conference*
219 *on Intelligent Robots and Systems*, 2017.
- 220 [13] C. Zheng, B. Eysenbach, H. Walke, P. Yin, K. Fang, R. Salakhutdinov, and S. Levine. Stabi-
221 lizing contrastive rl: Techniques for offline goal reaching. *arXiv preprint arXiv:2306.03346*,
222 2023.
- 223 [14] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fru-
224 end, P. Yianilos, M. Mueller-Freitag, and et al. The “something something” video database for
225 learning and evaluating visual common sense. In *IEEE international conference on computer*
226 *vision (ICCV)*, 2017.
- 227 [15] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster,
228 G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine,
229 P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint*
230 *arXiv:2406.09246*, 2024.
- 231 [16] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learn-
232 ing using nonequilibrium thermodynamics. In *International conference on machine learning*,
233 pages 2256–2265. PMLR, 2015.
- 234 [17] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural*
235 *information processing systems*, 33:6840–6851, 2020.
- 236 [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polo-
237 sukhin. Attention is all you need. *Advances in neural information processing systems*, 30,
238 2017.

- 239 [19] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.
- 240 [20] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy:
241 Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.
- 242 [21] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess,
243 A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to
244 robotic control. *arXiv preprint arXiv:2307.15818*, 2023.
- 245 [22] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna,
246 C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh,
247 C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of
248 Robotics: Science and Systems*, Delft, Netherlands, 2024.
- 249 [23] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine. Scaling cross-embodied learning:
250 One policy for manipulation, navigation, locomotion and aviation. In *Conference on Robot
251 Learning*, 2024.
- 252 [24] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine. Robotic control via
253 embodied chain-of-thought reasoning. In *Conference on Robot Learning*, 2024.
- 254 [25] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar. Cacti:
255 A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint
256 arXiv:2212.05711*, 2022.
- 257 [26] Z. Chen, S. Kiami, A. Gupta, and V. Kumar. Genaug: Retargeting behaviors to unseen situa-
258 tions via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.
- 259 [27] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta,
260 B. Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint
261 arXiv:2302.11550*, 2023.
- 262 [28] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani,
263 B. Zitkovich, F. Xia, et al. Open-world object manipulation using pre-trained vision-language
264 models. *arXiv preprint arXiv:2303.00905*, 2023.
- 265 [29] A. Peng, I. Sucholutsky, B. Z. Li, T. R. Sumers, T. L. Griffiths, J. Andreas, and J. A. Shah.
266 Learning with language-guided state abstractions. *arXiv preprint arXiv:2402.18759*, 2024.
- 267 [30] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Ex-
268 tracting actionable knowledge for embodied agents. In *International Conference on Machine
269 Learning*, pages 9118–9147. PMLR, 2022.
- 270 [31] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch,
271 Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language
272 models. *arXiv preprint arXiv:2207.05608*, 2022.
- 273 [32] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang,
274 R. Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In
275 *Conference on robot learning*, pages 287–318. PMLR, 2023.
- 276 [33] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg. Text2motion: From natural language
277 instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365, 2023.
- 278 [34] Z. Wang, S. Cai, G. Chen, A. Liu, X. Ma, and Y. Liang. Describe, explain, plan and select:
279 Interactive planning with large language models enables open-world multi-task agents. *arXiv
280 preprint arXiv:2302.01560*, 2023.

- 283 [35] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.
- 284
- 285 [36] S. K. S. Ghasemipour, D. Schuurmans, and S. S. Gu. Emaq: Expected-max q-learning operator for simple yet effective offline and online rl. In *International Conference on Machine Learning*, pages 3682–3691. PMLR, 2021.
- 286
- 287
- 288 [37] H. Chen, C. Lu, C. Ying, H. Su, and J. Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. *arXiv preprint arXiv:2209.14548*, 2022.
- 289
- 290 [38] P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.
- 291
- 292 [39] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- 293
- 294
- 295 [40] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.
- 296
- 297 [41] A. Hosseini, X. Yuan, N. Malkin, A. Courville, A. Sordoni, and R. Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*, 2024.
- 298
- 299 [42] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton. Structdiffusion: Language-guided creation of physically-valid structures using unseen objects. *arXiv preprint arXiv:2211.04604*, 2022.
- 300
- 301
- 302 [43] W. Huang, F. Xia, D. Shah, D. Driess, A. Zeng, Y. Lu, P. Florence, I. Mordatch, S. Levine, K. Hausman, et al. Grounded decoding: Guiding text generation with grounded models for robot control. *arXiv preprint arXiv:2303.00855*, 2023.
- 303
- 304
- 305 [44] A. Z. Ren, J. Clark, A. Dixit, M. Itkina, A. Majumdar, and D. Sadigh. Explore until confident: Efficient exploration for embodied question answering. In *Robotics Science and Systems (RSS)*, 2024.
- 306
- 307
- 308 [45] Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.
- 309
- 310 [46] S. Nair, E. Mitchell, K. Chen, B. Ichter, S. Savarese, and C. Finn. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. *Conference on Robot Learning (CoRL)*, 2021.
- 311
- 312
- 313 [47] L. P. Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–8. Citeseer, 1993.
- 314
- 315 [48] T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.
- 316
- 317 [49] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.
- 318
- 319 [50] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:25–55, 2020.
- 320
- 321 [51] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.
- 322
- 323
- 324 [52] O. Mees, L. Hermann, and W. Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):11205–11212, 2022.
- 325
- 326

- 327 [53] O. Mees, J. Borja-Diaz, and W. Burgard. Grounding language with visual affordances over
328 unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Au-*
329 *tomation (ICRA)*, London, UK, 2023.
- 330 [54] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data.
331 *arXiv preprint arXiv:2005.07648*, 2020.
- 332 [55] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox. Iris: Implicit
333 reinforcement without interaction at scale for learning control from offline robot manipulation
334 data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages
335 4414–4420. IEEE, 2020.
- 336 [56] S. Park, D. Ghosh, B. Eysenbach, and S. Levine. Hqrl: Offline goal-conditioned rl with latent
337 states as actions. *Advances in Neural Information Processing Systems*, 36, 2024.
- 338 [57] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for tem-
339 poral abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.
- 340 [58] P.-L. Bacon, J. Harb, and D. Precup. The option-critic architecture. In *Proceedings of the AAAI*
341 *conference on artificial intelligence*, volume 31, 2017.
- 342 [59] J. Schmidhuber. Learning to generate sub-goals for action sequences. In *Artificial neural*
343 *networks*, pages 967–972, 1991.
- 344 [60] P. Dayan and G. E. Hinton. Feudal reinforcement learning. *Advances in neural information*
345 *processing systems*, 5, 1992.
- 346 [61] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum. Hierarchical deep reinforce-
347 *ment learning: Integrating temporal abstraction and intrinsic motivation*. *Advances in neural*
348 *information processing systems*, 29, 2016.
- 349 [62] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and
350 K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International*
351 *conference on machine learning*, pages 3540–3549. PMLR, 2017.
- 352 [63] A. Levy, G. Konidaris, R. Platt, and K. Saenko. Learning multi-level hierarchies with hindsight.
353 *arXiv preprint arXiv:1712.00948*, 2017.
- 354 [64] O. Nachum, S. S. Gu, H. Lee, and S. Levine. Data-efficient hierarchical reinforcement learning.
355 *Advances in neural information processing systems*, 31, 2018.
- 356 [65] O. Nachum, S. Gu, H. Lee, and S. Levine. Near-optimal representation learning for hierarchical
357 reinforcement learning. *arXiv preprint arXiv:1810.01257*, 2018.
- 358 [66] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving
359 long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*,
360 2019.
- 361 [67] A. Ajay, A. Kumar, P. Agrawal, S. Levine, and O. Nachum. Opal: Offline primitive discovery
362 for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.
- 363 [68] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning
364 latent plans from play. In *Conference on Robot Learning (CoRL)*, pages 1113–1132. PMLR,
365 2020.
- 366 [69] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard. Latent plans for task-
367 agnostic offline reinforcement learning. In *Conference on Robot Learning*, pages 1838–1849.
368 PMLR, 2023.

- 369 [70] T. Zhang, S. Guo, T. Tan, X. Hu, and F. Chen. Generating adjacency-constrained subgoals in
370 hierarchical reinforcement learning. *Advances in neural information processing systems*, 33:
371 21579–21590, 2020.
- 372 [71] K. Pertsch, Y. Lee, and J. Lim. Accelerating reinforcement learning with learned skill priors.
373 In *Conference on robot learning*, pages 188–204. PMLR, 2021.
- 374 [72] E. Chane-Sane, C. Schmid, and I. Laptev. Goal-conditioned reinforcement learning with imagined
375 subgoals. In *International Conference on Machine Learning*, pages 1430–1440. PMLR,
376 2021.
- 377 [73] N. Savinov, A. Dosovitskiy, and V. Koltun. Semi-parametric topological memory for navigation.
378 *arXiv preprint arXiv:1803.00653*, 2018.
- 379 [74] B. Eysenbach, R. R. Salakhutdinov, and S. Levine. Search on the replay buffer: Bridging
380 planning and reinforcement learning. *Advances in neural information processing systems*, 32,
381 2019.
- 382 [75] S. Nair and C. Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via
383 visual subgoal generation. *arXiv preprint arXiv:1909.05829*, 2019.
- 384 [76] S. Nasiriany, V. Pong, S. Lin, and S. Levine. Planning with goal-conditioned policies. *Ad-*
385 *vances in Neural Information Processing Systems*, 32, 2019.
- 386 [77] Z. Huang, F. Liu, and H. Su. Mapping state space using landmarks for universal goal reaching.
387 *Advances in Neural Information Processing Systems*, 32, 2019.
- 388 [78] C. Hoang, S. Sohn, J. Choi, W. Carvalho, and H. Lee. Successor feature landmarks for long-
389 horizon goal-conditioned reinforcement learning. *Advances in neural information processing*
390 *systems*, 34:26963–26975, 2021.
- 391 [79] J. Kim, Y. Seo, and J. Shin. Landmark-guided subgoal generation in hierarchical reinforcement
392 learning. *Advances in neural information processing systems*, 34:28336–28349, 2021.
- 393 [80] L. Zhang, G. Yang, and B. C. Stadie. World model as a graph: Learning latent landmarks
394 for planning. In *International conference on machine learning*, pages 12611–12620. PMLR,
395 2021.
- 396 [81] D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, and S. Levine. Rapid exploration for open-
397 world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021.
- 398 [82] K. Fang, P. Yin, A. Nair, and S. Levine. Planning to practice: Efficient online fine-tuning by
399 composing goals in latent space. In *2022 IEEE/RSJ International Conference on Intelligent*
400 *Robots and Systems (IROS)*, pages 4076–4083. IEEE, 2022.
- 401 [83] J. Li, C. Tang, M. Tomizuka, and W. Zhan. Hierarchical planning through goal-conditioned
402 offline reinforcement learning. *IEEE Robotics and Automation Letters*, 7(4):10216–10223,
403 2022.
- 404 [84] J. Kim, Y. Seo, S. Ahn, K. Son, and J. Shin. Imitating graph-based planning with goal-
405 conditioned policies. *arXiv preprint arXiv:2303.11166*, 2023.
- 406 [85] K. Fang, P. Yin, A. Nair, H. R. Walke, G. Yan, and S. Levine. Generalization with lossy
407 affordances: Leveraging broad offline data for learning visuomotor tasks. In *Conference on*
408 *Robot Learning*, pages 106–117. PMLR, 2023.
- 409 [86] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing
410 instructions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

- 411 [87] J. Xing, M. Xia, Y. Zhang, H. Chen, W. Yu, H. Liu, X. Wang, T.-T. Wong, and Y. Shan.
412 Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint*
413 *arXiv:2310.12190*, 2023.
- 414 [88] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*,
415 2022.
- 416 [89] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based gen-
417 erative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*,
418 2020.
- 419 [90] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural*
420 *information processing systems*, 34:8780–8794, 2021.
- 421 [91] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image syn-
422 thesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer*
423 *vision and pattern recognition*, pages 10684–10695, 2022.
- 424 [92] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with
425 a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*,
426 volume 32, 2018.

427 **A Related work**

428 **Generative Models for Robotic Control:** Prior works have explored diverse ways to leverage
429 generative models, such as diffusion models [16, 17] and Transformers [18], for robotic control.
430 They have employed highly expressive generative models, potentially pre-trained on Internet-scale
431 data, for low-level control [19, 20, 21, 22, 23, 24], data augmentation [25, 26, 27], object detec-
432 tion [28, 29], semantic planning [30, 31, 32, 33, 34], and visual planning [6, 4, 7, 5, 8, 9]. Among
433 them, our work is most related to prior works that employ image or video prediction models to gen-
434 erate intermediate subgoal images for the given language task [6, 4, 7, 5, 8, 9]. These works use
435 diffusion models to convert language instructions into visual subgoal plans, which are then fed into
436 low-level subgoal-conditioned policies to produce actions. While sensible, this configuration leads
437 to failures due to the misalignment of the generative models and the low-level policies that control
438 the robot behavior, as shown in our experiments (section 3).

439 **Rejection Sampling:** One of our key ideas in this paper is based on rejection sampling, where we
440 sample multiple subgoal proposals from an image or video prediction model and pick the best one
441 based on a learned subgoal classifier. The idea of test-time rejection sampling has been widely used
442 in diverse areas of machine learning, such as filtering-based action selection in offline reinforcement
443 learning (RL) [35, 36, 37, 38], response verification in natural language processing [39, 40, 41], and
444 planning and exploration in robotics [42, 32, 33, 43, 44]. Previous works in robotics have proposed
445 several ways to filter out infeasible plans generated by pre-trained foundation models [42, 32, 33, 43,
446 45]. Unlike these works, we focus on filtering visual subgoals instead of language plans [32, 43, 45],
447 and do not involve any planning procedures [33] or structural knowledge [42]. While the subgoal
448 classifier we train resembles the classifier from [46], our classifier differs in two key ways. First, we
449 use our classifier to filter out “off-task” subgoals, whereas the classifier in [46] is used as a reward
450 function for training downstream policies. Second, the classifier from [46] is conditioned on the initial
451 state s_0 and the current state s , whereas our classifier is conditioned on the current state s and
452 a generated subgoal g .

453 **Goal-Conditioned Policy Learning:** Our method is broadly related to goal-conditioned policy
454 learning [47, 48, 49], language-conditioned policy learning [50, 51, 52, 53, 54], and hierarchical
455 control [4, 5, 55, 56, 57, 58]. Most prior works in hierarchical policy learning either train a high-
456 level policy from scratch that produces subgoals or latent skills [59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
457 69, 70, 71, 72, 56] or employ subgoal planning [73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 69, 85].
458 Unlike these works, we do not train a high-level subgoal prediction model from scratch nor involve
459 a potentially complex planning procedure. Instead, we sample multiple potential subgoals from a
460 pre-trained (or potentially fine-tuned) image or video prediction model and pick the best one based
461 on a trained subgoal classifier. Among hierarchical policy methods, perhaps the closest work to
462 ours is IRIS [55], which trains a conditional variational autoencoder to generate subgoal proposals
463 and selects the best subgoal that maximizes the task value function. While conceptually similar,
464 our method differs from IRIS in that we do not assume access to a reward function in order to train
465 a value function. Our classifier is trained on trajectories consisting only of images and language
466 descriptions.

467 **Diffusion Model Guidance:** The generative models we consider in our paper [86, 87] are diffusion-
468 based models trained using classifier-free guidance (CfG) [88]. Although we use a large value for
469 the language-prompt guidance parameter at inference in our experiments, we find that producing
470 “off-task” subgoals is still a common failure mode that is not solved by increasing this parameter
471 alone.

472 Classifier guidance [16, 89, 90] is also a plausible alternative to rejection sampling, but there are
473 some practical challenges in training a subgoal classifier for this purpose. First, the diffusion models
474 we consider use latent diffusion [91], and therefore would require training the subgoal classifier to
475 operate in the latent space of the diffusion model. Second, the subgoal classifier would need to be
476 trained on noised data in order to guide the diffusion denoising process of the generative model.
477 Nevertheless, classifier guidance is a potentially appealing direction for future work.

478 **B Preliminaries**

479 We consider the same problem setting as [4], where the goal is for a robot to perform a task de-
480 scribed by some previously unseen language command l . To do this, we consider the same three
481 dataset categories as in [4]: (1) language-labeled video clips \mathcal{D}_l which contain no robot actions; (2)
482 language-labeled robot data $\mathcal{D}_{l,a}$ that includes both language labels and robot actions; (3) unlabeled
483 robot data that only includes actions \mathcal{D}_a . The dataset $\mathcal{D}_{l,a}$ consists of a set of trajectory and task
484 language pairs, $\{(\tau^n, l^n)\}_{n=1}^N$, and a trajectory contains a sequence of state, $s_t^n \in \mathcal{S}$, and action,
485 $a_t^n \in \mathcal{A}$, pairs, $\tau^n = (s_0^n, a_0^n, s_1^n, a_1^n, \dots)$. Given these datasets, we assume access to two learned
486 modules:

- 487 1. **a subgoal generation module** from which we can sample multiple possible future sub-
488 goals. This can be trained on \mathcal{D}_l and $\mathcal{D}_{l,a}$.
- 489 2. **a low-level goal-reaching policy** that chooses actions to reach generated subgoals. This
490 can be trained on \mathcal{D}_a and/or $\mathcal{D}_{l,a}$.

491 Our contribution is a set of approaches to robustify the interface between these two modules.

492 While GHIL-Glue can be applied to any hierarchical imitation learning method consisting of the
493 two components mentioned above, in this work we apply GHIL-Glue to two specific algorithms: (1)
494 UniPi [5], in which a high-level model generates a subgoal video, and a low-level inverse-dynamics
495 model predicts the actions needed to “connect” the images in the video, and (2) SuSIE [4], in which
496 a high-level model generates a subgoal image by “editing” the current image observation, and a
497 goal-conditioned policy predicts actions to achieve the subgoal image. We define subgoals, $g \in \mathcal{G}$,
498 as video or image samples from the high-level models used in these algorithms.

499 **C Additional Discussion of Image Augmentation De-Synchronization**

500 Generated subgoals can contain visual artifacts that degrade the performance of both the low-level
501 control policy and the subgoal classifier. This performance degradation results from the distribution
502 shift between the subgoal images seen by the policy during training, which come from the robot
503 dataset, and the subgoal images seen during inference, which come from the generative model.
504 Ideally, the low-level policy and subgoal classifier would be trained on the same distribution of
505 *generated* subgoal images that they will see at inference time. However, due to the high degree of
506 variance in sampling images from a generative model, there is not a clear way to obtain generated
507 subgoal images that match the actual future states reached in trajectories in the training data. To
508 address this issue, we identify a simple yet non-obvious data augmentation practice to train the low-
509 level policy and subgoal classifier on goals from the robot dataset while also robustifying them to
510 visual artifacts in generated subgoals.

511 Applying image augmentation procedures such as random cropping or color jitter during training is a
512 standard approach in image-based robot learning methods [12] to improve the robustness of learned
513 models to distribution shifts between their training and evaluation domains. More formally, let ϕ
514 be the set of image augmentation parameters to be randomly sampled from space Φ , $p_\Phi(\cdot)$ be some
515 probability distribution over Φ , and let $\hat{\phi} \sim p_\Phi(\cdot)$ be some realization of augmentations sampled
516 from $p_\Phi(\cdot)$. Typically, for each training sample, a different value $\hat{\phi}$ is applied during training to
517 make a model robust to any augmentation in the space Φ .

518 For both the low-level goal-conditioned policy and the subgoal classifier, each training sample in-
519 cludes two images: the current state s and the corresponding goal g . Standard practice is to sample
520 augmentation parameters $\hat{\phi}$ and apply them to all images in a given training sample [4, 13], which
521 corresponds to applying the same $\hat{\phi}$ to both s and g . In a non-hierarchical policy setting, this makes
522 sense, because at inference time s and g will both be sampled from the camera observations of the
523 current environment instantiation. However, when using an image or video prediction model for
524 subgoal generation, at inference time the low-level policy and subgoal classifier will see states from

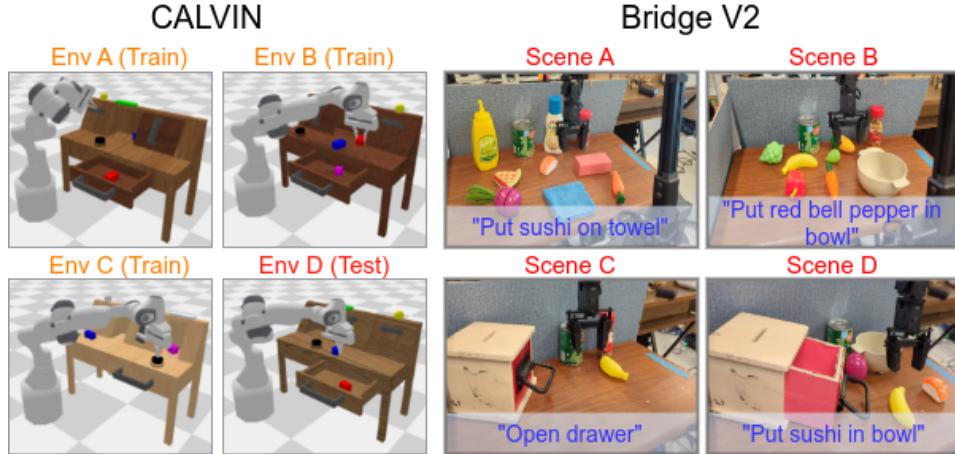
525 the camera observations, but the goals will be generated by the image or video prediction model.
526 There will often be differences in the visual artifacts between a camera observation s and the cor-
527 responding generated subgoal image g , such as differences in color, contrast, blurriness, and the
528 shapes of objects, which can degrade the performance of low-level policies and subgoal classifiers.

529 To encourage robustness to this distribution shift, we sample separate augmentation parameters for
530 s and g , denoted by $\hat{\phi}_s$ and $\hat{\phi}_g$ (i.e., we de-synchronize the image augmentations applied to s and
531 g). Random cropping, brightness shifts, contrast shifts, saturation shifts, and hue shifts comprise
532 our space of augmentations. Concretely, for each s and g pair sampled during training, a different
533 random crop, brightness, contrast, saturation, and hue shift are applied to s than are applied to g .
534 This forces the low-level policy and the subgoal classifier to learn to make accurate predictions on
535 (s, g) pairs that have differences in visual artifacts.

536 While image augmentations are ubiquitous in robot learning methods, our experiments show that
537 the standard way of applying image augmentations for goal-conditioned policies and classifiers is
538 deficient for the hierarchical policy methods that we consider. We also note that augmentation de-
539 synchronization is applied not only to the policy, but also to the subgoal classifier (section 2.1),
540 which has a significant impact on overall performance (section 3).

541 D Experimental Domains

542 We study the degree to which GHIL-Glue improves existing hierarchical imitation learning algo-
543 rithms across a number of tasks in simulation and physical experiments that assess zero-shot gen-
544 eralization. We evaluate our method on the CALVIN [10] simulation benchmark and the Bridge
545 V2 [11] physical experiment setup with a WidowX250 robot. The experimental domains are visual-
546 ized in fig. 2.



547 **Figure 2: Experimental Domains.** Simulation Environments (Left): Train/test environments in the CALVIN
548 simulation benchmark. The environments each have different table textures, furniture positions, and initial con-
549 figurations of the colored blocks. Each environment contains 34 tasks, each with an associated language instruc-
550 tion. To test zero-shot generalization, environment D is held out for evaluation. Physical Environments (Right):
551 We consider four test scenes in the Bridge V2 robot platform with four total language instructions. To test zero-
shot generalization, these test scenes contain novel objects, language commands, and object configurations not
seen in the training data.

547 E Comparison Algorithms

548 A detailed description of the comparison algorithms referenced in section 3.2 is provided below:

- 549 1. **LCBC Diffusion Policy:** Low-level language-conditioned behavior cloning diffusion pol-
550 icy [20] trained only on robot trajectories with language annotations. We use the same
551 implementation as in [4].

- 552 2. **OpenVLA [21]:** A SOTA language-conditioned vision-language-action model (VLA)
 553 trained on the Open X-Embodiment dataset [2] (which includes the entirety of the Bridge
 554 V2 dataset).
- 555 3. **SuSIE [4]:** A method which fine-tunes InstructPix2Pix [86], an image-editing diffusion
 556 model, to generate subgoal images given the current image observation. Low-level control
 557 is performed using a goal-conditioned policy. For SuSIE and all methods that build on it,
 558 we predict subgoals 20 steps in the future as in the original paper.
- 559 4. **UniPi [5]:** A method which fine-tunes a language-conditioned video prediction model
 560 on robot data and then uses an inverse dynamics model for low-level goal reaching. For
 561 UniPi and all methods that build on it, we predict video sequences of 16 frames. As the
 562 original UniPi model is not publicly available, we re-implement UniPi by fine-tuning the
 563 video model from [87].
- 564 5. **GHIL-Glue (SuSIE / UniPi):** GHIL-Glue applied on top of either SuSIE or UniPi. For
 565 all experiments we implement the subgoal filtering step by sampling four to eight subgoals
 566 from the high-level video prediction model and selecting amongst them. We directly filter
 567 the subgoal images generated by the SuSIE model. We filter the video sequences generated
 568 by the UniPi model based on the final frame of each sequence.
- 569 6. **GHIL-Glue (SuSIE / UniPi) - Subgoal Filtering Only:** GHIL-Glue applied to SuSIE or
 570 UniPi using subgoal filtering but without augmentation de-synchronization.
- 571 7. **GHIL-Glue (SuSIE / UniPi) - Aug De-sync Only:** GHIL-Glue applied to SuSIE or UniPi
 572 using augmentation de-synchronization but without subgoal filtering.

573 F Discussion of Results

574 **Simulation Experiments:** We present results on the CALVIN benchmark in table 1. Applying
 575 GHIL-Glue yields significant performance increases for SuSIE and UniPi, increasing the average
 576 successful task sequence length from **2.94** to **3.69** for SuSIE and from **1.02** to **1.56** for UniPi. **GHIL-**
 577 **Glue (SuSIE) achieves a new SOTA on CALVIN for policies that use observations from a single**

578 **RGB camera.** The two components of GHIL-Glue (subgoal filtering and image augmentation
 579 de-synchronization) improve performance when applied individually, but, when applied together, these
 580 components build on each other, leading to a performance increase greater than the sum of the
 581 individual benefits. Specifically, for SuSIE, image augmentation de-synchronization and subgoal
 582 filtering individually yield increases in sequence length of 0.56 and 0.02 respectively, whereas when
 583 applied together they yield an increase of 0.75. Similarly, for UniPi, the individual improvements
 584 yield increases in sequence length of 0.08 and 0.34 respectively, compared to an increase of 0.54
 585 when applied together.

586 When applied alone, image augmentation de-synchronization increases the average successful task
 587 sequence length from 2.94 to 3.51 for SuSIE and from 1.02 to 1.1 for UniPi. We hypothesize
 588 that augmentation de-synchronization improves performance a large amount with SuSIE because
 589 its low-level policy is conditioned on a camera observation image s from the environment and a
 590 subgoal image g generated by the image model. When generalizing to the held-out test environment
 591 D, the SuSIE image model generates subgoal images with visual discrepancies from the camera
 592 observation images. In contrast, the UniPi video model predicts a sequence of frames as opposed to a
 593 single subgoal image. The UniPi low-level policy functions as an inverse dynamics model, choosing
 594 actions to link between the frames of the generated subgoal video, and is therefore conditioned on
 595 an s and g that both come from the predicted subgoal video.

596 When applied alone, subgoal filtering has a small effect on SuSIE, while on UniPi it increases the
 597 average successful task sequence length from 1.02 to 1.36. This suggests that unless the SuSIE
 598 low-level policy is made robust to visual artifacts in generated subgoals, simply selecting the most
 599 task relevant subgoal is insufficient to improve performance. As discussed previously, the SuSIE

600 low-level policy is more sensitive to visual artifacts in generated subgoals than is the UniPi inverse
601 dynamics model.

602 **Physical Experiments:** We present results (table 2) comparing GHIL-Glue (SuSIE) to OpenVLA
603 and SuSIE across four environments on the Bridge V2 robot platform that require interacting with
604 a number of objects on a cluttered table (fig. 2). These environments require generalizing to novel
605 scenes, with novel objects, and with novel language commands that are not seen in the Bridge V2
606 dataset. GHIL-Glue applied to SuSIE outperforms SuSIE across all tasks and outperforms Open-
607 VLA, a 7-billion parameter SOTA VLA, on 3 out of 4 tasks. Significantly, the baseline SuSIE
608 implementation does not outperform OpenVLA on a single task, whereas GHIL-Glue (SuSIE) out-
609 performs OpenVLA on 3 out of 4 tasks, demonstrating that hierarchical goal conditioned architec-
610 tures with well-tuned interfaces between the high and low-level policies can outperform SOTA VLA
611 methods on zero-shot generalization tasks.

612 G Classifier Training

613 **Training objective:** The classifier is trained using binary cross-entropy loss:

$$\mathcal{J}(\theta) = \mathbb{E}_{(s,g,l) \sim D_l} [\log(f_\theta(s, g, l))] + \mathbb{E}_{(s',g',l') \sim N(D_l)} [1 - \log(f_\theta(s', g', l'))],$$

614 where D_l is the language-annotated dataset that consists of trajectory and language task pairs, and
615 N is a function for generating negative examples from the dataset. Given a dataset D_l , N generates
616 negatives from D_l in the following ways:

- 617 1. **Wrong Instruction:** (s, g, l') where l' is sampled from a different transition than s and g .
- 618 2. **Wrong Goal Image:** (s, g', l) where g' is sampled from a different transition than s and l .
- 619 3. **Reverse Direction:** (g, s, l) , where the order of the current image observation and the
620 subgoal image have been switched.

621 Across all our experiments, we sample 50% of each training batch to be positive examples and
622 50% of each training batch to be negative examples. Of the negative examples, 40% are “wrong
623 instruction”, 40% are “reverse direction”, and 20% are “wrong goal image”.

624 **Goal sampling:** In a given training tuple (s_t, g, l) , g is sampled by taking the goal image from the
625 s_{t+k} , where k is a uniformly sampled integer from 16 to 24.

626 **Network architecture and training hyperparameters:** The classifier network architecture consists
627 of a ResNet-34 encoder from [11], followed by a two-layer MLP with layers of dimension 256.
628 Separate encoders are used to encode the image observations and the goal images (parameters are
629 not shared between the two). Both of these encoders use FiLM conditioning [92] after each residual
630 block to condition on the language instruction. Classifier networks are trained using a learning rate
631 of 3×10^{-4} and a batch size of 256 for 100,000 gradient steps. A dropout rate of 0.1 is used.

632 H Image Augmentations

633 During training of low-level policy networks and classifier networks, we apply the following aug-
634 mentations to the image observations and the goal images, in the following order:

- 635 1. Random Resized Crop:
 - 636 • scale: (0.8, 1.0)
 - 637 • ratio: (0.9, 1.1)
- 638 2. Random Brightness Shift:
 - 639 • shift ratio: 0.2

- 640 3. Random Contrast:
 641 • Contrast range: (0.8, 1.2)
 642 4. Random Saturation:
 643 • Saturation range: (0.8, 1.2)
 644 5. Random Hue:
 645 • shift ratio: 0.1

646 Figure 3 visualizes examples from the Bridge dataset before and after augmentations are applied:

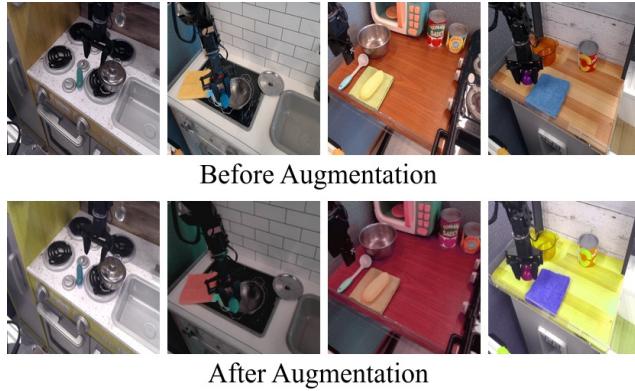


Figure 3: Image augmentation examples Examples of images from the Bridge dataset before and after having the image augmentations applied to them that are used during policy and classifier training.

647 I Qualitative Analysis

648 I.1 Effect of subgoal filtering

649 Although we use classifier-free guidance (CfG) [88] on the image or video generative model with
 650 respect to the language-prompt at inference in our experiments, we find that producing “off-task”
 651 subgoals is still a common failure mode that is not solved by increasing the guidance parameter
 652 alone. In fig. 4, we visualize how subgoal filtering can prevent “off-task” subgoals generated by the
 653 image or video model from being passed to the low-level control policy.

654 I.2 Classifier rankings

655 We show examples of how the classifier network ranks generated goal images on tasks from Scene D
 656 of our physical experimental domain. Figures 5a, 5b, 5c show examples of the classifier correctly
 657 ranking the generated goal images (highly ranked images correspond to making progress towards
 658 correctly completing the language instruction), while fig. 5d shows an example of the classifier
 659 erroneously giving high rankings to goal images that do not make progress towards completing the
 660 language instruction. Note that while the classifier scores can be close across various goal images,
 661 so long as the relative ranking of the generated goal images is correct, then incorrect subgoal images
 662 will be rejected and correct subgoal images will be passed to the low-level policy.

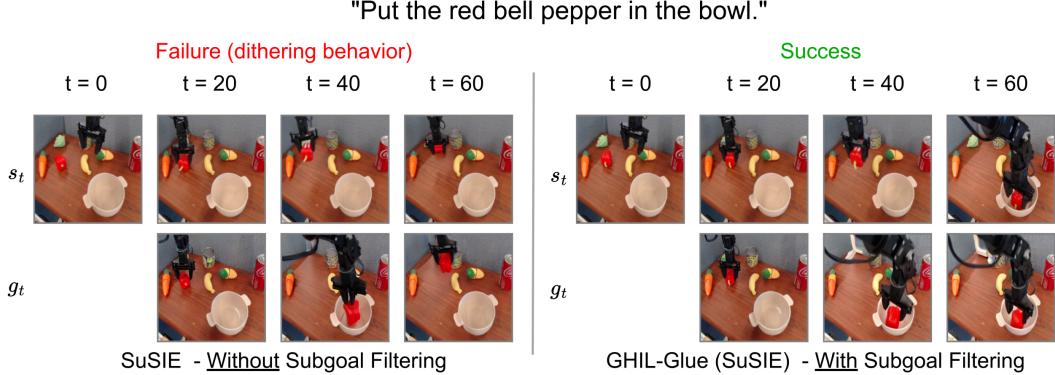


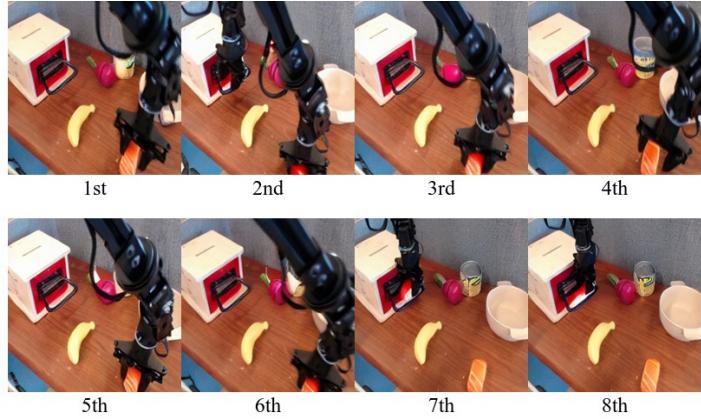
Figure 4: GHIL-Glue Subgoal Filtering. We visualize policy rollouts of SuSIE without subgoal filtering vs. GHIL-Glue SuSIE with subgoal filtering. We show the states reached every 20 timesteps (top row) and the corresponding predicted subgoals (bottom row). Without subgoal filtering, the subgoal at $t = 60$ is not consistent with making progress towards placing the pepper in the bowl, causing the robot to dither and drop the pepper. When subgoal filtering is used, the selected subgoals make iterative progress towards a successful task completion.

Figure 5: Classifier ranking examples Examples of the classifier network rankings on 8 generated candidate subgoals given an observation from Scene D of the physical experiments and a language instruction. Note that during GHIL-Glue inference, only the first-ranked subgoal is passed to the low-level policy.

Language instruction:
"Put the sushi into the bowl."



Image Observation



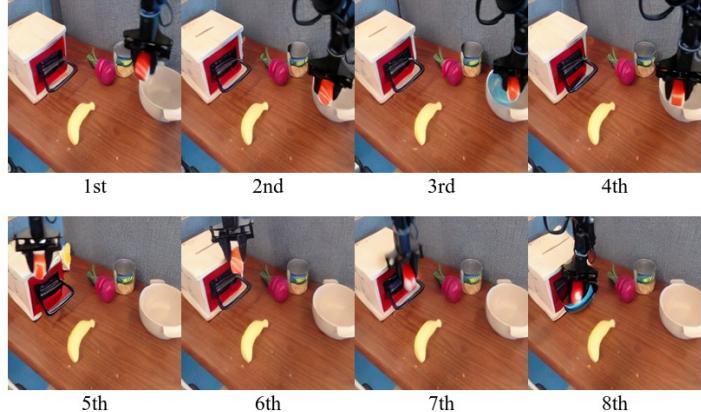
Generated goal images ranked by the classifier

(a) Correct Example of Classifier Filtering The classifier correctly ranks the subgoal images where the robot is grasping the sushi higher than the subgoal images where the robot is grasping the drawer handle.

Language instruction:
"Put the sushi into the bowl."



Image Observation



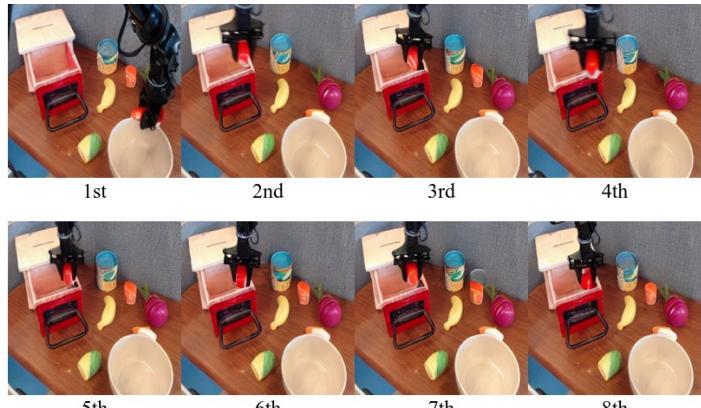
Generated goal images ranked by the classifier

(b) Correct Example of Classifier Filtering The classifier correctly ranks the subgoal images where the robot moves to place the grasped sushi into the bowl higher than the subgoal images where the robot moves its gripper towards the drawer handle. It ranks the subgoal image with the hallucinated blue bowl-like artifact last.

Language instruction:
"Put the sushi into the bowl."



Image Observation



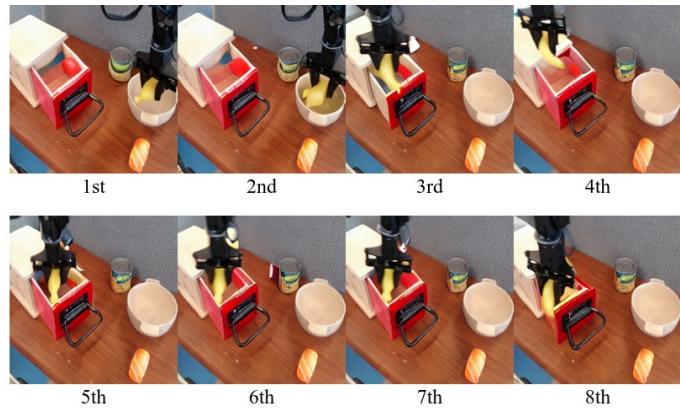
Generated goal images ranked by the classifier

(c) Correct Example of Classifier Filtering The classifier correctly ranks the subgoal image highest that shows the robot completing the correct task – only a single generated subgoal image shows the robot placing the sushi into the bowl, while all other generated subgoal images show the robot placing the sushi into the drawer.

Language instruction:
"Put the banana into the drawer."



Image Observation



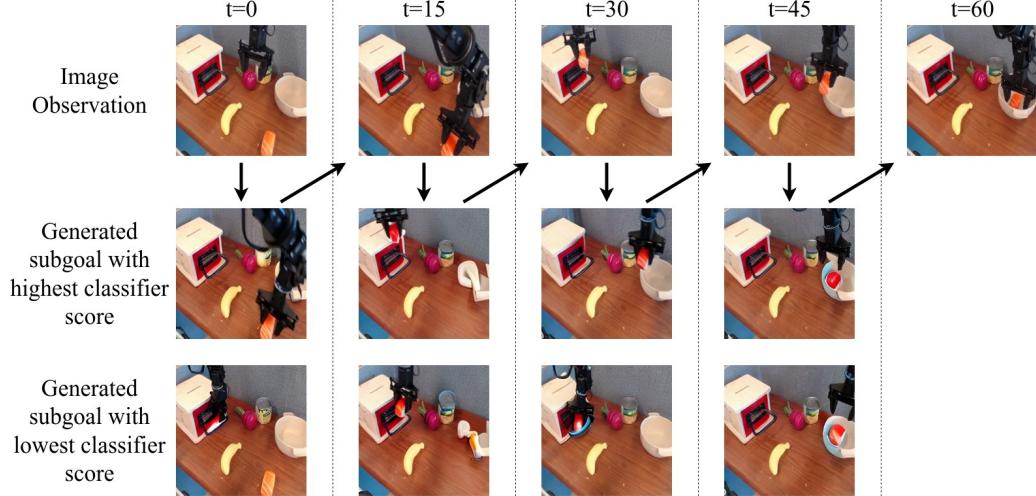
Generated goal images ranked by the classifier

(d) The classifier incorrectly ranks the subgoal images higher where the robot is placing the banana into the bowl than it ranks the subgoal images where the robot is placing the banana into the drawer. This could be due to there being a strong bias for placing objects in bowls in the Bridge V2 training data.

663 **I.3 Trajectory Visualizations**

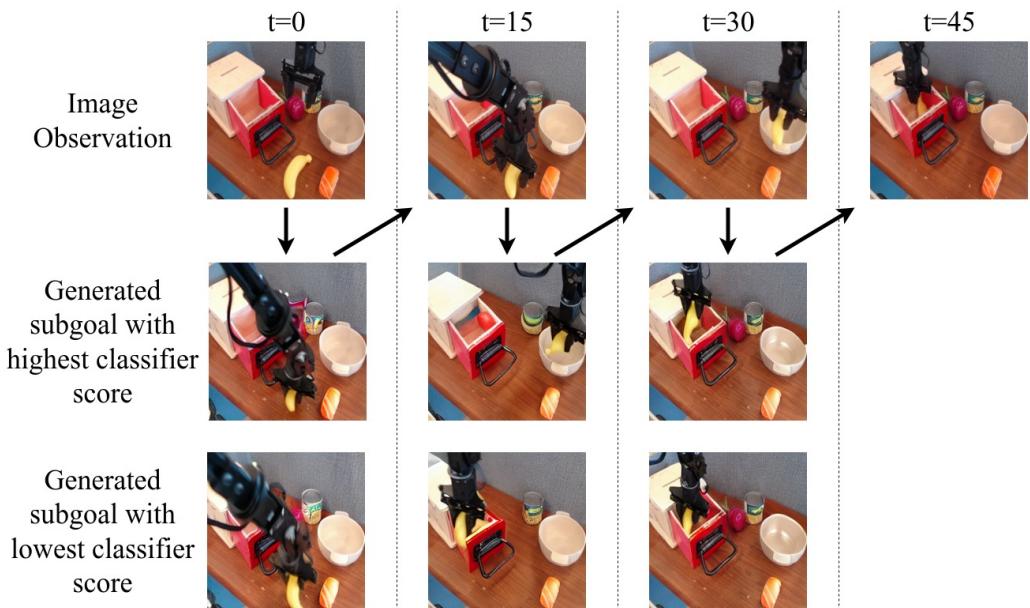
664 We show examples of rollouts of GHIL-Glue (SuSIE) on our physical experiment set up. These
 665 examples showcase when GHIL-Glue successfully filters out off-task subgoal images (Figure 6a),
 666 as well as an instance of when GHIL-Glue nearly causes a failure (Figure 6b).

Figure 6: GHIL-Glue (SuSIE) Trajectory Visualization Visualization of a rollout of GHIL-Glue (SuSIE) on Scene D in the physical experiments set up. The top row shows the current image observation at every timestep at which the video prediction model is queried. The second and third rows show the highest and lowest ranked generated subgoal images out of the 8 generated subgoal images, as ranked by the classifier. Note that during GHIL-Glue inference, only the first-ranked subgoal is passed to the low-level policy.



Language instruction: "Put the sushi into the bowl."

(a) **"Put the sushi into the bowl."** This rollout shows two examples of the classifier filtering preventing the policy from going off-task: at $t = 0$, the lowest ranked generated subgoal shows the gripper grasping the drawer handle instead of moving to grasp the sushi; at $t = 30$, the lowest ranked generated subgoal shows the gripper moving towards the drawer handle instead of towards placing the sushi into the bowl. Note the hallucinated objects and artifacts visible in the goal images at $t = 15, 30, 45$. Augmentation de-synchronization helps to make the low-level policy and classifier robust to hallucinated artifacts such as these.



Language instruction: "*Put the banana into the drawer.*"

(b) “Put the banana into the drawer.” In this rollout, classifier filtering fails and causes a near-miss. At $t = 15$, the classifier ranks a subgoal image highest that shows the robot placing the banana into the bowl instead of the drawer. However, at $t = 30$, when the robot reaches the state specified by this subgoal image, the subsequent generated subgoals all show the robot correctly placing the banana into the drawer. Although, as in this example, the classifier network can occasionally rank incorrect subgoal images higher than correct subgoal images, such errors occur infrequently as GHIL-Glue (SuSIE/UniPi) outperforms base-SuSIE/UniPi across all of our physical and simulated experiments.

667 **I.4 Qualitative Analysis of Augmentation De-synchronization**

668 We see that when applying aug-
 669 mentation de-synchronization, the
 670 number of failures due to low-
 671 level policy errors (missed grasps,
 672 dropping held objects, etc.) de-
 673 creases, indicating that augmentation
 674 de-synchronization is important for
 675 the low-level policy to be able to cor-
 676 rectly interpret and follow the sub-
 677 goal images generated by the video
 678 prediction model. This is particularly
 679 important in domains where there is
 680 a large visual generalization gap be-
 681 tween the training data and the eval-
 682 uation tasks. For example, in the
 683 CALVIN benchmark, the colors and
 684 shapes of objects differ between the
 685 training and evaluation scenes. This
 686 difference causes the subgoals gener-
 687 ated by the video prediction model to
 688 often contain objects with incorrect shapes and colors (Figure 7). Augmentation de-synchronization
 689 seems to be critical to allowing the low-level policy to be robust to these hallucinations and artifacts.

Language Instruction: "Go push the red block left."

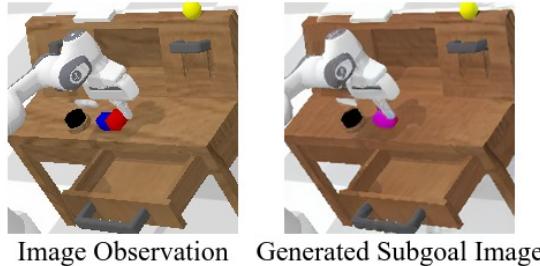


Figure 7: Generated Subgoal Image on CALVIN A subgoal image generated by the SuSIE video model on the unseen environment D of the CALVIN benchmark. The colors and shapes of objects are different in each of the four CALVIN environments, and since the model was not trained on data from environment D, it often generates images with incorrect shapes and colors. Augmentation de-synchronization is important for the low-level policy and classifier to be able to handle these mismatches between image observations and corresponding generated subgoal images.

690 **J Number of Candidate Subgoals**

691 We conduct an ablation over the number of candidate subgoals used for subgoal filtering in GHIL-
 692 Glue (SuSIE) in the CALVIN benchmark. We find that GHIL-Glue (SuSIE) achieves similar per-
 693 formance whether 4, 8, or 16 candidate subgoals are used. In our main results (section 3.3), we
 694 report the performance of GHIL-Glue (SuSIE) on the CALVIN benchmark when using 8 candidate
 695 subgoals for filtering. For GHIL-Glue (UniPi) on the CALVIN benchmark, we use 4 candidate sub-
 696 goals for filtering, due to the increased computation burden of generating video subgoals with the
 697 UniPi video model vs. generating image subgoals with the SuSIE image model. In our physical
 698 experiments, we run GHIL-Glue (SuSIE) using 4 candidate subgoals for filtering.

Method	Tasks completed in a row					
	1	2	3	4	5	Avg. Len.
GHIL-Glue (SuSIE) - 4 samples	95.2%	86.0%	71.2%	60.5%	50.0%	3.63
GHIL-Glue (SuSIE) - 8 samples	95.2%	88.5%	73.2%	62.5%	49.8%	3.69
GHIL-Glue (SuSIE) - 16 samples	95.0%	86.5%	72.8%	60.8%	48.0%	3.63

Table 3: Effect of Number of Candidate Goal Images Sampled in GHIL-Glue (SuSIE) Success rates on the validation tasks from environment D of the CALVIN Challenge when using GHIL-Glue (SuSIE) when using 4, 8, or 16 candidate goal images with classifier filtering. Results are averaged across 4 random seeds. Results are similar across all numbers of samples, with 8 samples performing the best by a slight margin.