GHIL-Glue: Hierarchical Control with Filtered Subgoal Images

Anonymous Author(s)
Affiliation
Address
email

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

Keywords: Hierarchical Imitation Learning, Image Generation, Video Prediction

1 Introduction

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

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

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

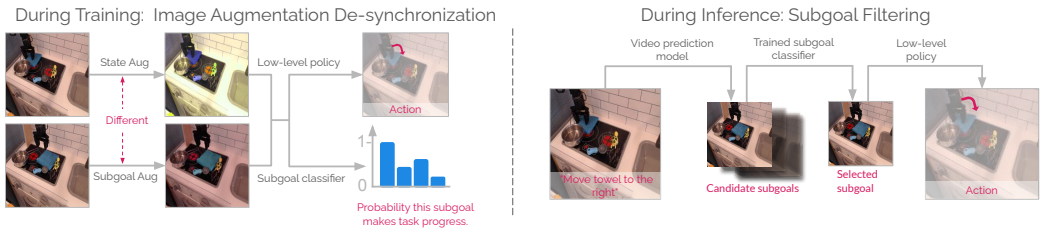


Figure 1: GHIL-Glue. We consider language-conditioned image and video prediction models that can generate multiple subgoals. GHIL-Glue has two components: augmentation de-synchronization (top) and subgoal filtering (bottom). Subgoal filtering: We train a classifier to identify which subgoal is most likely 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).

To address these issues, we propose Generative Hierarchical Imitation Learning-Glue (GHIL-Glue) (fig. 1), a method to *robustly* "Glue" together image or video generative models to a low-level robotic control policy. **First**, we filter out "off-task" subgoals that are physically inconsistent with the commanded language instruction. We do this by training a subgoal classifier to predict the likelihood of the transition between the current state and a given subgoal resulting in progress towards completing the provided language instruction. We then sample a number of candidate subgoals from the generative model and choose the subgoal with the highest classifier ranking. **Second**, we identify a simple yet non-obvious data augmentation practice to robustify the low-level policy and subgoal classifier to visual artifacts in the generated subgoals. While image augmentations are ubiquitous in robot learning methods, our key finding is that the standard way of applying image augmentations does not make low-level policies robust to visual artifacts in generated subgoal images.

Experiments on the CALVIN [10] simulation benchmark and four language-conditioned tasks on the Bridge V2 physical robot platform [11] suggest that GHIL-Glue improves upon prior SOTA methods for zero-shot generalization while adding minimal additional algorithmic complexity.

2 GHIL-Glue

2.1 Subgoal Filtering

The image and video generative models we consider are first pre-trained on general Internet-scale image and video data, and then fine-tuned on a modest amount of robot data. A common failure mode we observe across different models is that, over the course of executing a task, the model begins to go "off-task," meaning that it starts generating subgoals that are consistent with the current image observation but that do not progress towards completing the language instruction l. We hypothesize that this is due to the distribution shift between the Internet image and video pre-training data and the robot data they are fine-tuned on.

To address this challenge, we train a subgoal classifier $f_\theta(s, g, l)$ on a language-conditioned dataset of trajectories $\mathcal{D}_l$ that predicts the probability that the transition between the current image observation s and the next subgoal g makes progress towards completing language instruction l. During training, we sample positive examples of state-goal transitions for l from the set of trajectories that successfully complete the instruction. We construct negative examples in the following three ways: **1) Wrong Instruction:** (s, g, l') where l' is sampled from a different transition than s and g, **2) Wrong Goal Image:** (s, g', l) where g' is sampled from a different transition than s and l, and **3) Reverse Direction**: (g, s, l), where the order of the current image observation and the subgoal image have been switched. We refer to this dataset of negative examples constructed from $\mathcal{D}_l$ as $\mathcal{D}_l^-$. We then train the subgoal classifier by minimizing the binary cross entropy loss between the positive examples and the constructed negative examples:

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

2.2 Image Augmentation De-Synchronization

For both the low-level goal-conditioned policy and the subgoal classifier, each training sample includes two images: the current state s and the corresponding goal g. Applying image augmentation procedures such as random cropping or color jitter during training is a standard approach in image-based robot learning methods [12] to improve the robustness of learned models to distribution shifts between their training and evaluation domains. Standard practice is to sample augmentation parameters $\hat{\phi}$ and apply them to all images in a given training sample [4, 13], which corresponds to applying the same $\hat{\phi}$ to both s and g. In a non-hierarchical policy setting, this makes sense, because at inference time s and g will both be sampled from the camera observations of the current environment instantiation. However, when using an image or video prediction model for subgoal generation, at inference time the low-level policy and subgoal classifier will see states from the camera observations, but the goals will be generated by the image or video prediction model. There will often be differences in the visual artifacts between a camera observation s and the corresponding generated subgoal image g, such as differences in color, contrast, blurriness, and the shapes of objects, which can degrade the performance of low-level policies and subgoal classifiers.

To encourage robustness to this distribution shift, we sample separate augmentation parameters for s and g, denoted by $\hat{\phi}_s$ and $\hat{\phi}_g$ (i.e., we de-synchronize the image augmentations applied to s and g). Concretely, for each s and g pair sampled during training, a different random crop, brightness, contrast, saturation, and hue shift are applied to s than are applied to g. This forces the low-level policy and the subgoal classifier to be robust to differences in visual artifacts between s and g.

3 Experiments

3.1 Experimental Domains

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

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

3.2 Comparison Algorithms

To evaluate GHIL-Glue's performance, we study the impact of applying it to two SOTA hierarchical imitation learning algorithms: SuSIE [4] and UniPi [5]. We use either 4 or 8 candidate subgoals for subgoal filtering (see appendix E for details). To evaluate the importance of hierarchy more generally, we also compare GHIL-Glue to a flat language-conditioned diffusion policy (LCBC Diffusion Policy). Finally, we consider ablations where we separately study the impact of each of our proposed

contributions: subgoal filtering (section 2.1) and de-synchronizing augmentations (section 2.2). For physical experiments, we additionally consider a comparison to OpenVLA [15], which is trained on the Open X-Embodiment dataset [2] (which includes the Bridge V2 dataset).

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.

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

Physical Experiments: We present results (table 2) comparing GHIL-Glue (SuSIE) to OpenVLA and SuSIE across four environments on the Bridge V2 robot platform that require interacting with a number of objects on a cluttered table (fig. 2). GHIL-Glue applied to SuSIE outperforms SuSIE across all tasks and outperforms OpenVLA, a 7-billion parameter SOTA VLA, on 3 out of 4 tasks. Significantly, the baseline SuSIE implementation does not outperform OpenVLA on a single task, whereas **GHIL-Glue (SuSIE) outperforms OpenVLA on 3 out of 4 tasks**, demonstrating that hierarchical goal conditioned architectures with well-tuned interfaces between the high and low-level policies can outperform SOTA VLA methods on zero-shot generalization tasks. See appendix D for qualitative examples of success and failure cases of GHIL-Glue in physical experiments.

4 Conclusion

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

References

[1] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning (CoRL)*, 2019.

[2] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. 2024.

[3] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[4] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[5] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[6] I. Kapelyukh, V. Vosylius, and E. Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters*, 2023.

[7] Y. Du, M. Yang, P. Florence, F. Xia, A. Wahid, B. Ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023.

[8] A. Ajay, S. Han, Y. Du, S. Li, A. Gupta, T. Jaakkola, J. Tenenbaum, L. Kaelbling, A. Srivastava, and P. Agrawal. Compositional foundation models for hierarchical planning. *Advances in Neural Information Processing Systems*, 36, 2024.

[9] J. Gao, K. Hu, G. Xu, and H. Xu. Can pre-trained text-to-image models generate visual goals for reinforcement learning? *Advances in Neural Information Processing Systems*, 36, 2024.

[10] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. In *IEEE Robotics and Automation Letters (RAL)*, 2021.

[11] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[12] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *International Conference on Intelligent Robots and Systems*, 2017.

[13] C. Zheng, B. Eysenbach, H. Walke, P. Yin, K. Fang, R. Salakhutdinov, and S. Levine. Stabilizing contrastive rl: Techniques for offline goal reaching. *arXiv preprint arXiv:2306.03346*, 2023.

[14] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, and et al. The" something something" video database for learning and evaluating visual common sense. In *IEEE international conference on computer vision (ICCV)*, 2017.

[15] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[16] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[17] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

A Experimental Domains

We study the degree to which GHIL-Glue improves existing hierarchical imitation learning algorithms across a number of tasks in simulation and physical experiments that assess zero-shot generalization. We evaluate our method on the CALVIN [10] simulation benchmark and the Bridge V2 [11] physical experiment setup with a WidowX250 robot. The experimental domains are visualized in fig. 2.

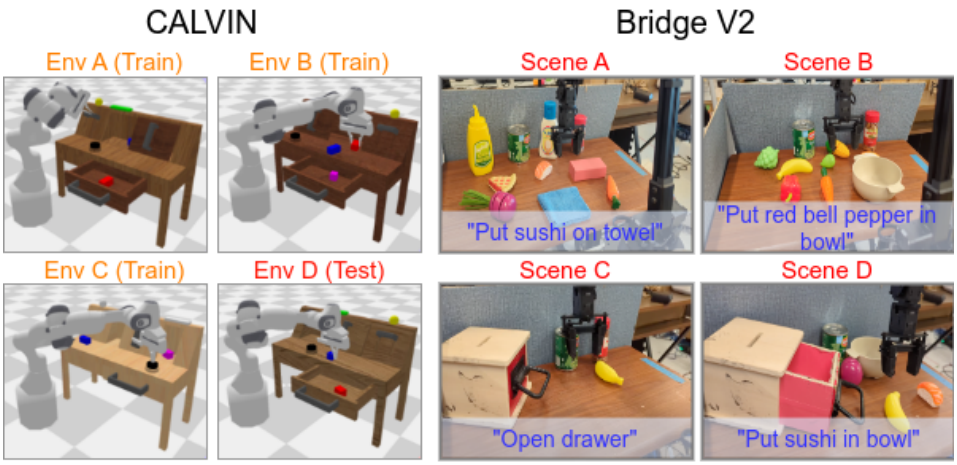


Figure 2: Experimental Domains. Simulation Environments (Left): Train/test environments in the CALVIN simulation benchmark. The environments each have different table textures, furniture positions, and initial configurations of the colored blocks. Each environment contains 34 tasks, each with an associated language instruction. To test zero-shot generalization, environment D is held out for evaluation. Physical Environments (Right): 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.

B Classifier Training

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

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

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

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

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

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

Network architecture and training hyperparameters: The classifier network architecture consists of a ResNet-34 encoder from [11], followed by a two-layer MLP with layers of dimension 256.

Separate encoders are used to encode the image observations and the goal images (parameters are not shared between the two). Both of these encoders use FiLM conditioning [16] after each residual block to condition on the language instruction. Classifier networks are trained using a learning rate of 3×10^{-4} and a batch size of 256 for $100,000$ gradient steps. A dropout rate of 0.1 is used.

C Image Augmentations

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

1. Random Resized Crop:
   - scale: $(0.8, 1.0)$
   - ratio:$(0.9, 1.1)$
2. Random Brightness Shift:
   - shift ratio: 0.2
3. Random Contrast:
   - Contrast range: $(0.8, 1.2)$
4. Random Saturation:
   - Saturation range: $(0.8, 1.2)$
5. Random Hue:
   - shift ratio: 0.1

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.

D Qualitative Analysis

D.1 Effect of subgoal filtering

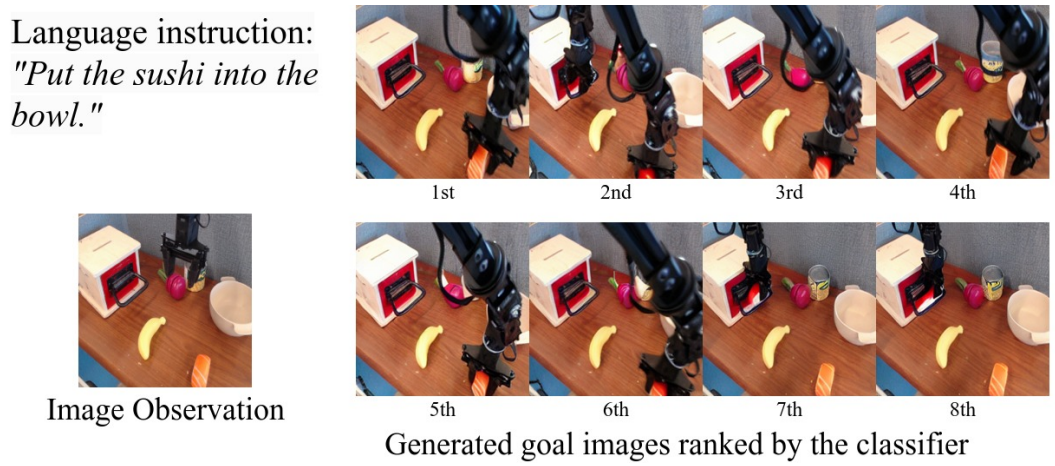
Although we use classifier-free guidance (CfG) [17] on the image or video generative model with respect to the language-prompt at inference in our experiments, we find that producing "off-task" subgoals is still a common failure mode that is not solved by increasing the guidance parameter alone. In fig. 4, we visualize how subgoal filtering can prevent "off-task" subgoals generated by the image or video model from being passed to the low-level control 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.

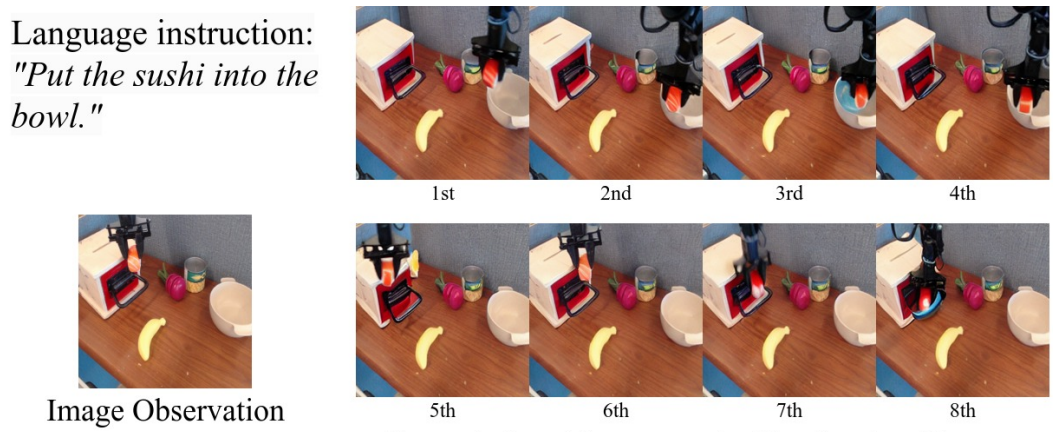
D.2 Classifier rankings

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

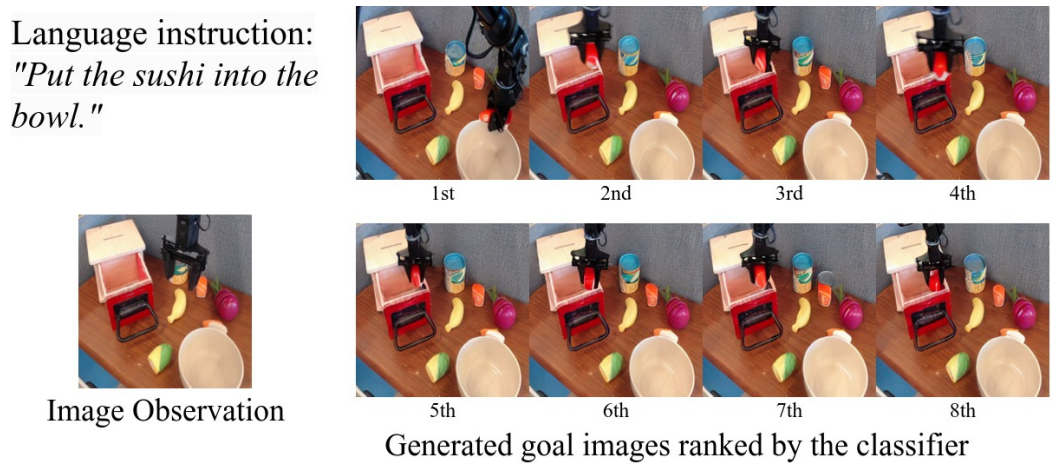
Figure 5: Classifier ranking examples Examples of the classifier network rankings on 8 generated candidate subgoals given an observation from Scene D of the physical experiments and a language instruction. Note that during GHIL-Glue inference, only the first-ranked subgoal is passed to the low-level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



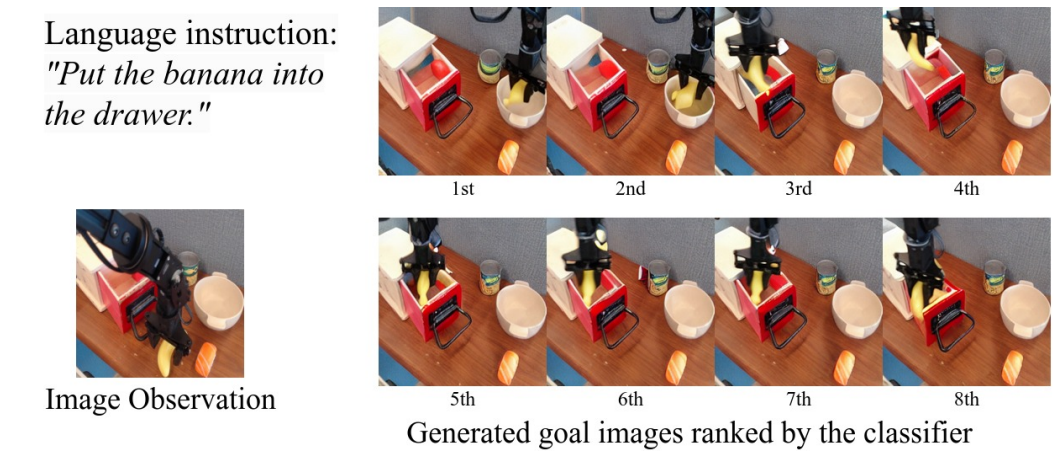
(a) Correct Example of Classifier Filtering The classifier correctly ranks the subgoal images where the robot is grasping the sushi higher than the subgoal images where the robot is grasping the drawer handle.



(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.



(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.

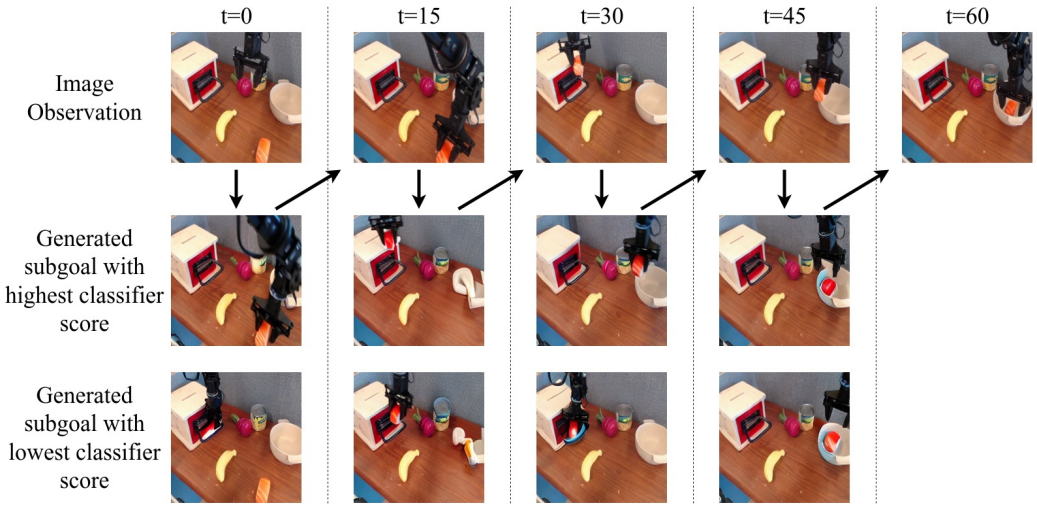


(d) The classifier incorrectly ranks the subgoal images higher where the robot is placing the banana into the bowl than it ranks the subgoal images where the robot is placing the banana into the drawer. This could be due to there being a strong bias for placing objects in bowls in the Bridge V2 training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

D.3 Trajectory Visualizations

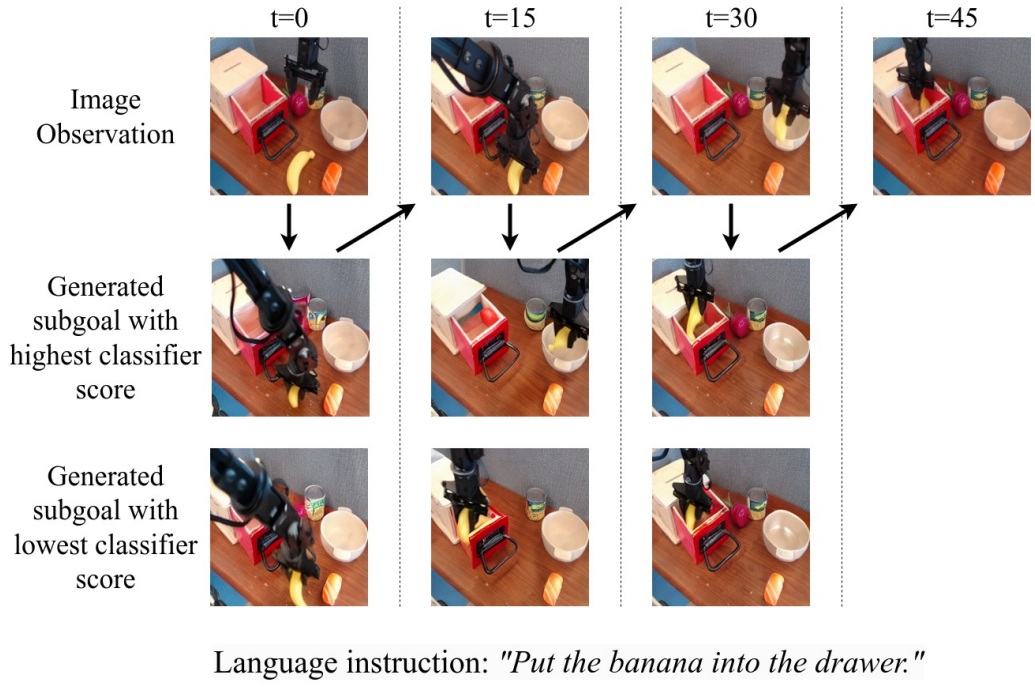
We show examples of rollouts of GHIL-Glue (SuSIE) on our physical experiment set up. These examples showcase when GHIL-Glue successfully filters out off-task subgoal images (Figure 6a), as well as an instance of when GHIL-Glue nearly causes a failure (Figure 6b).

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.



(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.

D.4 Qualitative Analysis of Augmentation De-synchronization

We see that when applying augmentation de-synchronization, the number of failures due to low-level policy errors (missed grasps, dropping held objects, etc.) decreases, indicating that augmentation de-synchronization is important for the low-level policy to be able to correctly interpret and follow the subgoal images generated by the video prediction model. This is particularly important in domains where there is a large visual generalization gap between the training data and the evaluation tasks. For example, in the CALVIN benchmark, the colors and shapes of objects differ between the training and evaluation scenes. This difference causes the subgoals generated by the video prediction model to often contain objects with incorrect shapes and colors (Figure 7). Augmentation de-synchronization seems to be critical to allowing the low-level policy to be robust to these hallucinations and artifacts.

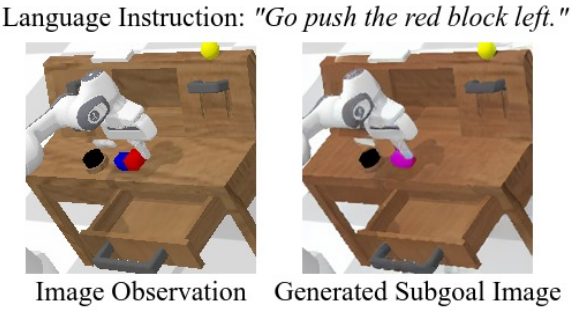


Figure 7: Generated Subgoal Image on CALVIN A subgoal image generated by the SuSIE video model on the unseen environment D of the CALVIN benchmark. The colors and shapes of objects are different in each of the four CALVIN environments, and since the model was not trained on data from environment D, it 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.

E Number of Candidate Subgoals

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