Just Add Force for Delicate Robot Policies

Anonymous Author(s)
Affiliation
Address
email

Abstract: Robot trajectories used for learning end-to-end robot policies typically contain end-effector and gripper position, workspace images, and language. Policies learned from such trajectories are unsuitable for delicate grasping, which require tightly coupled and precise gripper force and gripper position. We collect and make publically available 130 trajectories with force feedback of successful grasps on 30 unique objects. Our current-based method for sensing force, albeit noisy, is gripper-agnostic and requires no additional hardware. We train and evaluate two diffusion policies: one with (forceful) the collected force feedback and one without (position-only). We find that forceful policies are superior to position-only policies for delicate grasping and are able to generalize to unseen delicate objects, while reducing grasp policy latency by near 4x, relative to LLM-based methods. With our promising results on limited data, we hope to signal to others to consider investing in collecting force and other such tactile information in new datasets, enabling more robust, contact-rich manipulation in future robot foundation models. Our data, code, models, and videos are viewable at https://justaddforce.github.io/.

1 Introduction

Robot foundation models [1, 2, 3, 4, 5, 6, 7] leverage large-scale datasets spanning diverse objects, scenes, and embodiments to produce generalizable, cross-platform robot policies. The utilized data adheres to limited modalities: vision, language, and robot action–most typically, workspace camera view, text annotation of a given task, end-effector pose, and binary (open or closed) gripper position [2]. The latter, binary gripper position, especially without force feedback, precludes robot foundation models from successfully grasping many delicate objects such as soft produce, brittle dried goods, paper containers, and other such fragile and deformable items. In this paper, we propose a modification to this archetypal structure: continuous, rather than binary, gripper positions and corresponding grasp force feedback.

We contribute 1) a novel dataset of 130 trajectories with continuous gripper position and force feedback, spanning 30 unique objects ranging in deformability, volume, and mass (from 5g to 500g) and 2) train diffusion policies [8] with and without force feedback, showing that force enables delicate grasping performant with state-of-the-art LLM-based methods at a near 4x reduced latency with promise for generalizability at greater data scale.

Our position is that force, a strong supervisory signal of contact and grasp-success, along with continuous gripper position, rather than binary open or closed states, should be included in future datasets used in the training of robot foundation models. Our current-draw-based force sensing method is gripper-agnostic and requires no special hardware (“skin” or otherwise). While noisier and less accurate than bespoke solutions, policies trained on our data are capable of delicate grasps. Improved resolution and frequency of force and other tactile signals likely would further improve grasp fidelity and robustness.

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

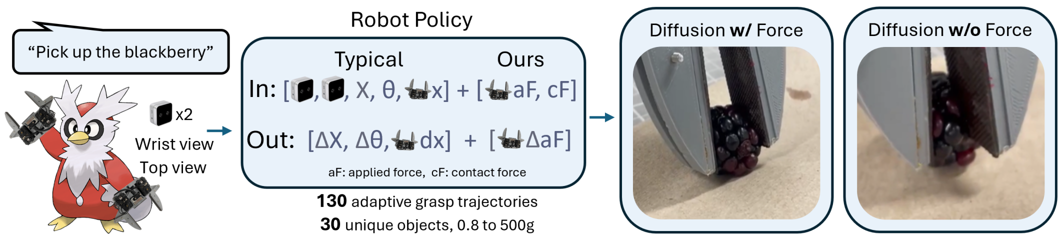


Figure 1: We leverage LLM-directed expert demonstrations [9] of delicate objects to generate a dataset of 130 successful grasps of 30 different objects spanning a variety of physical properties. Our trajectories, unlike other datasets used in end-to-end learning [2, 5], contain observed gripper applied and contact force and the action of increased gripper applied force. We train diffusion policies [8] on the dataset with and without force data and observe that forceful policies can, despite limited data, replicate trained behavior and generalize to unseen delicate objects at 4x reduced latency relative to LLM-policies, and position-only policies cannot.

2 Related Work

Large-scale robotic datasets [2, 5] have enabled the emergence of generalist, end-to-end robot foundation models [1, 2, 3, 4, 6, 7] which typically append a behavior cloning architecture [10, 11, 12, 8] to generate robot policies from a larger representation space. However, these robot foundation models are pre-trained on limited modalities: vision, language, and robot joint and/or end effector data.

There is a growing field exploring new modalities for end-to-end robot policy models, primarily in audio and tactile sensing [13, 14, 15]. Such policies offer novel advantages in contact-rich manipulation and manipulation in visually occluded scenes but require new complexities, namely: custom and/or nontrivially emulated hardware and increased model complexity in processing and incorporation of high-dimension input data. In comparison, manipulator applied force and contact normal force can be approximated as 1-dimensional. And while traditional force sensing is costly relative to audio and touch and thus unused in end-to-end learning, we leverage current draw as a gripper-agnostic force measurement, without additional sensing hardware, using a MAGPIE gripper [9, 16] which interfaces with its motor control board to more easily provide this information.

In this work we examine grasping of delicate and deformable objects, which has primarily been done via adaptive grasping methods with traditional closed-loop control or LLM-based robot control: [9, 17, 18, 19, 20]. Traditional controllers are not as generalizable as methods leveraging large amounts of data [9], such as LLM-based methods, which in turn are high latency and computationally expensive. Utilizing force feedback from expert demonstrations of adaptive grasping in training or fine-tuning of robot foundation models may yield both lower latency and high generalizability.

3 Methods

We introduce a dataset of 130 successful adaptive grasp trajectories across 30 unique objects spanning two orders of magnitude in mass (1g to 500g) and variable deformability (additional dataset detail and download link in A.1 and A.3). Data is collected at 5 Hz from a MAGPIE gripper [16] on a UR5 robot arm with a wrist-mounted Realsense D405 camera and a Realsense D435 camera overlooking a square, 55cm table. The user also provides a task instruction. The robot is positioned arbitrarily above and in-front of the target object, and the target object is placed arbitrarily on the table. We make our dataset publically available in an RLDS format [21] compatible with Open-X and Droid datasets, Octo models, and other foundation models trained on RLDS format data.

To collect expert demonstrations, we employ DeliGrasp [9], which navigates to the object and queries the LLM with the user-provided object description and uses LLM-estimated object mass, friction coefficient, and spring constants as parameters in a proportional controller which increases applied force and gripper closure until a measured contact force [17, 18]. We command applied force by incrementing motor torque limit on a Dynamixel motor (an equivalent actuator-agnostic approach would be to increase supply current), and we measure contact force from increased current draw.

We "distill" these expert demonstrations [22] by training four diffusion policies [5, 8] on this data, with and without (position-only) force, and with the entire trajectory or the grasp only (GO) (training details in A.2). Initial testing showed that full trajectory policies did not learn meaningful robot motion, potentially due to the low amounts of data and each (robot start, target object) position pair being unique. Henceforth, we refer only to the policies trained on grasp-only data. By default, position-only policies apply a constant 2N and forceful policies begin at the lowest setting, 0.15N.

In our experiments we localize the object and position the robot at a viable grasp position using [9] and deploy and evaluate the policies only during the stationary grasp portion of a trajectory. We manually qualify deformation failures on a per-object common-sense basis (object crushed, cracked, etc...) and check for slip by raising the robot gripper directly vertically by 10cm. As the average adaptive grasp in the dataset completes in under 10 steps, for one "grasp" we rollout the policy for 15 steps at 4Hz (3.75s per grasp vs 14.11s for an LLM-based grasp [9], a 3.76x reduction).

4 Experiments

We conduct 10 trials of grasps on 10 different objects: four objects seen in the training set (empty paper cup, raspberry, tomato, paper cup filled with water) but assessed to be difficult objects and six unseen objects (blackberry, egg, empty metal can, empty soft-shelled taco, pepper, potato chip). We compare between two models: 1) position-only policies (PO) with the canonical gripper position input and output and image & task instruction inputs, and 2) forceful policies with applied force and contact force as additional inputs and applied force as an additional output.

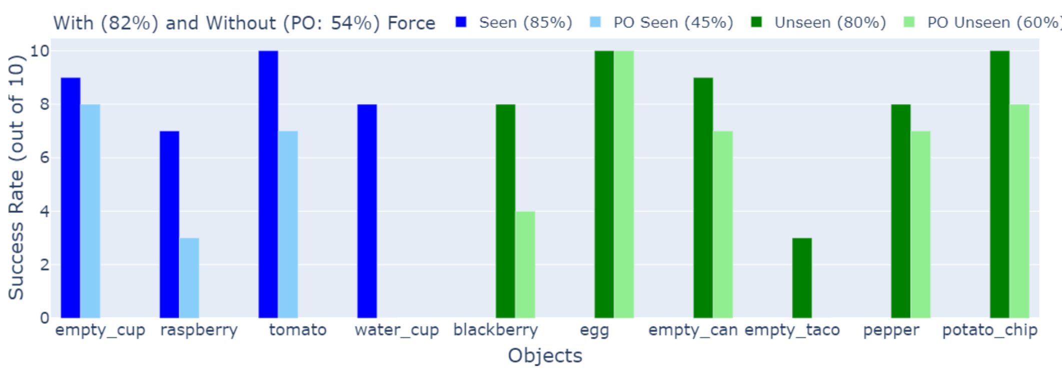


Figure 2: We conduct a series of 10 trials for a selection of 10 objects; four seen in training, six unseen. Forceful policies (82%) replicate seen grasps (85%) and generalize to similar but unseen objects (80%). position-only policies (54%) retain a level of performance on seen (45%) and improve on unseen (60%) delicate objects, suggesting that continuous gripper position control alone contributes to successful delicate grasps. We note that position-only policy failures are generally deforming and compress more than forceful policies (see Fig. 3)

Across all objects, we find that forceful policies (82% success) are superior to position-only policies (54% success) (Fig. 2) and that position-only policies compress more than forceful policies (Fig. 3). Position-only policies are still capable, perhaps because they are artifacts of forceful adaptive grasping, just trained without the force feedback, and the control law may be implicitly learned through solely vision, gripper position, and task instruction. Forceful policies generalize to unseen objects (80% success, compared to 85% for seen objects) and withhheld policies improve (60%, up from 45%), potentially due to relatively stiff objects like the egg and potato chip being forgiving for additional compression.

More granularly, we qualify failures as either deformation or slip. While both policies generally perform deformation failures, forceful policies slip (7 occurrences) more than position-only policies (3 occurrences), representing a 28% vs. 6.1% share of respective policy failures. For produce like tomatoes and peppers, position-only policies generate grasps which are individually successful, but we observe that after 10 trials, the produce is noticeably deformed ("mushy") due to repeated greater compression, unlike for forceful policies (Fig. 3). We leave these grasps marked as successes as the produce threshold of desirability is dependent on the end-user.

Additionally, both policies occasionally generate generated grasps end several mm, up to several cm, offset from the object. We note these occurrences as "null grasps," separate from successes or failures. We note that the forceful policies produced null grasps 11.5% of the time (13 occurrences, even across seen and unseen grasps) and position-only policies produced null grasps 20% of the time (25 occurrences with 6 on the raspberry and 5 on the blackberry). We also observe volatility, though much rarer, in gripper position and force post-contact, resulting in abrupt crushes (notably affecting the average applied force on the raspberry in Fig. 3).

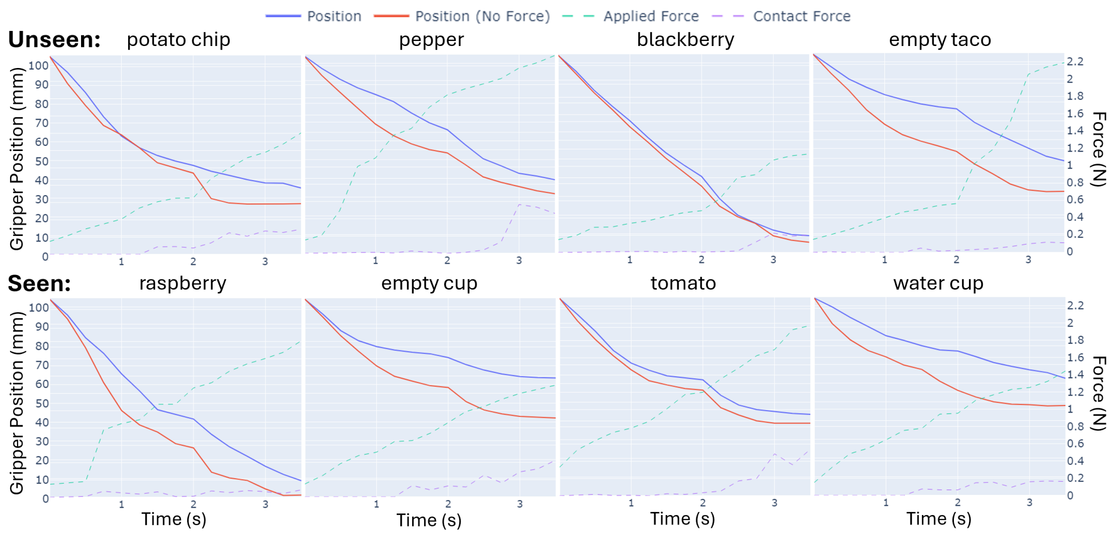


Figure 3: We plot 1) forceful policies gripper position (blue), applied force (green dash), and contact force (purple dash) and 2) position-only policies gripper position (red) against time, with additional plots in A.4. Uniformly, position-only policies close more narrowly than forceful policies, leading to deformation failures, particularly for delicate objects like blackberries and raspberries. Individual position-only policy grasps on produce like tomatoes and peppers are successful, but we observe that after 10 trials, the produce is noticeably deformed due to greater compression, unlike for forceful grasps. On objects like the pepper, empty taco, blackberry, and tomato, applied force flattens as contact force increases.

In Fig. 3, we depict per-object grasp trajectories and forces and observe that position-only policies uniformly compress more than forceful policies. position-only policies are initially more aggressive in closing the gripper and often continue aggressive closure past contact, resulting in deformation failures. Forceful policies flatten applied force as contact force increases for some objects (pepper, empty taco, blackberry, tomato), showing vestiges of the proportional control law used in expert demonstrations, however, policies still apply more force than is typically needed and have not fully learned the control characteristics. Additionally, while objects span a large range of gripper position (5 to 65mm), final applied force lies in a smaller range (1.1N to 2.3N).

5 Conclusion

We add force observations and actions to the common data structure of imagery, task instruction, robot pose, gripper position used in training end-to-end robot policy models in a dataset of 130 grasps across 30 objects. We train a diffusion policy trained on force feedback which outperforms a policy trained without force on delicate objects and generalizes to unseen objects, indicating that force may be a worthwhile inclusion in future data collection endeavors.

Limitations and Future Work: As the second derivative of gripper position, force may encode enough information to be all you need for manipulation. Our models are currently only evaluated at rest, and we do not explore adaptive grasping while in motion. Moreover, our evaluated models are simplistic and trained on a toy dataset—future work includes finetuning on foundation models which allow new modalities [3] or collecting diverse, large scale data with force feedback. Adaptive grasping may also benefit from a pretrained LLM backbone to leverage common-sense reasoning about forces. Force also has applications beyond our demonstrated use case of slip/contact sensing and may be used for generating non-prehensile manipulation trajectories.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

[2] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, 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. Schlkopf, 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. Bchler, 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. Bharadhwaj, 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. Vakil, 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. Silvrio, 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. Tulsiani, 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. Kumar, 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. https://arxiv.org/abs/2310.08864, 2023.

[3] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[4] M. J. 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, 2024. URL https://arxiv.org/abs/2406.09246.

[5] 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. Martn-Martn, 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.

[6] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine. Robotic control via embodied chain-of-thought reasoning, 2024. URL https://arxiv.org/abs/2407.08693.

[7] J. Wen, Y. Zhu, J. Li, M. Zhu, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen, Y. Peng, F. Feng, and J. Tang. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation, 2024. URL https://arxiv.org/abs/2409.12514.

[8] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[9] W. Xie, M. Valentini, J. Lavering, and N. Correll. Deligrasp: Inferring object properties with llms for adaptive grasp policies, 2024. URL https://arxiv.org/abs/2403.07832.

[10] N. M. M. Shafiullah, Z. J. Cui, A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone, 2022. URL https://arxiv.org/abs/2206.11251.

[11] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[12] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior generation with latent actions, 2024. URL https://arxiv.org/abs/2403.03181.

[13] Z. Liu, C. Chi, E. Cousineau, N. Kuppuswamy, B. Burchfiel, and S. Song. Maniwav: Learning robot manipulation from in-the-wild audio-visual data, 2024. URL https://arxiv.org/abs/2406.19464.

[14] H. Li, Y. Zhang, J. Zhu, S. Wang, M. A. Lee, H. Xu, E. Adelson, L. Fei-Fei, R. Gao, and J. Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation, 2022. URL https://arxiv.org/abs/2212.03858.

[15] R. Bhirangi, V. Pattabiraman, E. Erciyes, Y. Cao, T. Hellebrekers, and L. Pinto. Anyskin: Plug-and-play skin sensing for robotic touch, 2024. URL https://arxiv.org/abs/2409.08276.

[16] N. Correll, D. Kriegman, S. Otto, and J. Watson. A versatile robotic hand with 3d perception, force sensing for autonomous manipulation. *arXiv:2402.06018*, 2024.

[17] Z. Ding, N. Paperno, K. Prakash, and A. Behal. An adaptive control-based approach for 1-click gripping of novel objects using a robotic manipulator. *IEEE Transactions on Control Systems Technology*, 27(4):1805–1812, 2019. doi:10.1109/TCST.2018.2821651.

[18] K. Sullivan, H. Chizeck, and A. Marburg. Using a rigid gripper on objects of different compliance underwater. In *OCEANS 2022, Hampton Roads*, pages 1–4, 2022. doi:10.1109/OCEANS47191.2022.9977278.

[19] M. Al-Mohammed, R. Adem, and A. Behal. A switched adaptive controller for robotic gripping of novel objects with minimal force. *IEEE Transactions on Control Systems Technology*, 31(1):17–26, 2023. doi:10.1109/TCST.2022.3171655.

[20] Y. Gong, Y. Xing, J. Wu, and Z. Xiong. Tactile-Based Slip Detection Towards Robot Grasping. In H. Yang, H. Liu, J. Zou, Z. Yin, L. Liu, G. Yang, X. Ouyang, and Z. Wang, editors, *Intelligent Robotics and Applications*, pages 93–107, Singapore, 2023. Springer Nature Singapore. ISBN 978-981-9964-95-6.

[21] S. Ramos, S. Girgin, L. Hussenot, D. Vincent, H. Yakubovich, D. Toyama, A. Gergely, P. Stanczyk, R. Marinier, J. Harmsen, O. Pietquin, and N. Momchev. Rlds: an ecosystem to generate, share and use datasets in reinforcement learning, 2021. URL https://arxiv.org/abs/2111.02767.

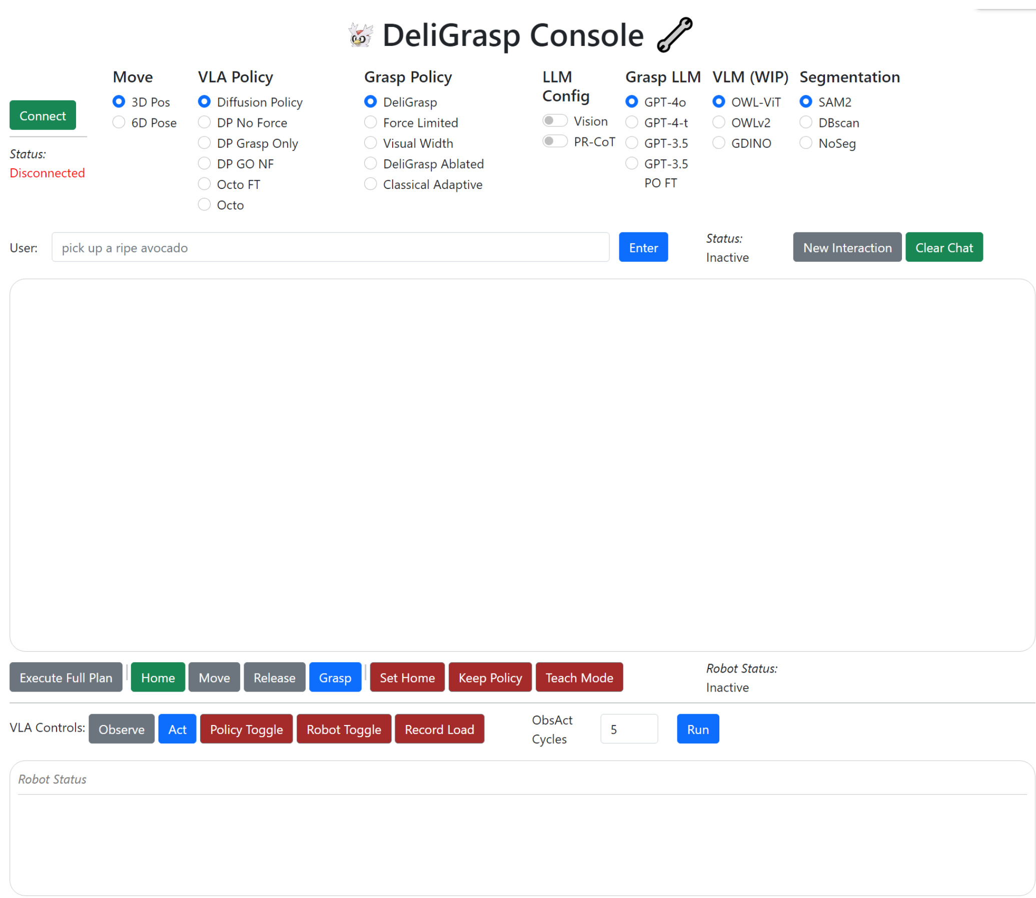
[22] H. Ha, P. Florence, and S. Song. Scaling up and distilling down: Language-guided robot skill acquisition, 2023. URL https://arxiv.org/abs/2307.14535.

A Appendix

A.1 Dataset Details

Objects: orange bottle, peeled garlic clove, stuffed animal, garlic clove, green block, tomato, red screwdriver handle, scallion stalk, small avocado, yellow ducky, water bottle, small black motor, empty paper cup, circuit board, red button, scalion stalk, orange noodle bag, yellow block, strawberry, bottle cap, small suction cup, light green chip, ziptie bag, metal lock, cardboard box, raspberry, large bearing, paper cup with water, small red green apple, paper airplane, green circuit board, plastic bottle, cherry tomato, mushroom, garlic bulb.

For most objects collect 5-7 trajectories, with a few one-offs. We use a webapp console to interoperate between DeliGrasp, robot control, and diffusion policy evaluation.



A.2 Diffusion Policy Training

We train our models using DROID Policy Learning [5], which deviates from the vanilla implementations in three ways: 1) opting out of SparseSoftmax to retrieve regional keypoints, instead keeping the feature channels of the image embedding, 2) adding language conditioning by encoding task instruction and adding it to the observation input, and 3) downsizing the input dimension to a fixed size. We do not alter the DROID hyperparameters except for the following: we train for 3000 steps (30 epochs) and a batch size of 16. Model training is done locally on a 2070 Super, taking approximately 1 hour to train per model. We use T_o, T_a, T_p of 2, 8, and 16, but in evaluation use receding horizon control ($T_a = 1$).

A.3 Data and Model Downloads

1. https://justaddforce.github.io/datasets
2. https://justaddforce.github.io/models

A.4 Additional Unseen Plots