ForceSight: Multi-Task Text-Guided Mobile Manipulation with Visual-Force Goals

Anonymous Author(s)
Affiliation
Address
email

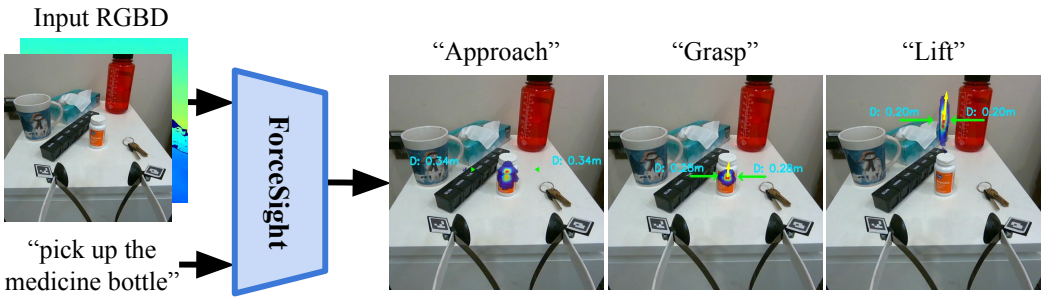

Figure 1: ForceSight is an RGBD-adapted, text-conditioned vision transformer. Given an RGBD image and a text prompt, ForceSight produces visual-force goals for a mobile manipulator. Action primitives, shown above each image, are appended to the text input by a simple low-level controller.

Abstract: Prior work has demonstrated that deep models that output kinematic keyframes enable manipulation by real robots with human-interpretable motion goals. Contact forces are critical to manipulation, yet have typically been relegated to lower-level execution during keyframe-driven manipulation. We present ForceSight, a system for multi-task, text-guided mobile manipulation with a deep model that outputs visual and force goals (visual-force keyframes) suitable for visual-force servoing. Given a single RGBD image and a text prompt as input, ForceSight's deep model outputs a visual-force goal, which can be inferred at a high enough rate to work with a moving camera. We evaluate ForceSight using an eye-in-hand RGBD camera on a mobile manipulator. We show that by explicitly representing net applied force and grip force, ForceSight predicts forces suitable to the task, operates more effectively, and provides human-interpretable force goals. Videos, code, and trained models are available at https://force-sight.github.io/.

Keywords: Transformers, Imitation Learning, Manipulation, Force Sensing, Language Grounding

1 Introduction

Robotic manipulation has significantly benefited from the integration of tactile and force information. These modalities enable direct perception and control of contact with the environment, which can be advantageous. For example, grasping a small object off of a flat smooth surface can benefit from first making fingertip contact with the surface and then sliding the fingertips across the surface to pick up the object. Success depends on fingertip force that is high enough to maintain contact with the surface and grasp the object, but low enough to slide the fingertips. In general, grip force

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

and net applied force provide strong cues that appropriate contact has been achieved for a task. In contrast, kinematic models are not well suited to achieving task-relevant contact and forces.

We present ForceSight, a transformer-based, text-conditioned robotic planner that enables coarse-to-fine visual servoing by proposing tactile and kinematic objectives relevant to a given task, thus enabling the interpretable execution of tasks in novel environments with unseen object instances. ForceSight uses an RGBD-adapted, text-conditioned vision transformer to encode an RGBD image from a gripper-mounted camera, from which visual-force keyframes relevant to the next step of the specified task are predicted. These keyframes consist of 3D fingertip contact locations, grip force, and resultant force applied to the gripper.

We present the following contributions:

- **Force-based planner:** We present ForceSight, a system that infers visual-force goals given an RGBD image from an eye-in-hand camera and a natural language prompt.
- **Real-world tasks:** We show that our method enables a mobile manipulator to learn 10 text-conditioned tasks using a simple low-level controller, generalizing to unseen environments and novel object instances.
- **Novel dataset collection method:** We present a novel method for collecting data for imitation learning, speeding up the process by 5 times in comparison to typical data collection methods.
- **Open source:** We release our code, dataset, and trained models.

2 Related Work

Imitation learning has been widely adopted in the field of robotics, enabling robots to learn from human demonstrations and execute complex tasks. Recent work has explored imitation learning to create robot policies from combined text and image input [1, 2, 3, 4]. Although language and vision are both rich modalities for specifying and executing a task, practical considerations such as occlusion and depth ambiguity limit the performance of such systems in practice. To address these limitations, our method utilizes force information to ground the visual representations that determine a robotic policy. While some methods [2, 3, 5, 6] learn behaviors in a data-efficient manner, and generalize to object pose via clever data augmentation or explicit object representations, they often come at the cost of being restricted to narrow environments with fixed cameras, and do not test with out-of-distribution examples. At the other end of the spectrum, many imitation learning methods prioritize generality, but require much more data to do so [1, 4]. We believe our method lies between these two regimes, achieving generalization across environments and camera poses while only requiring a modest amount of data collection effort.

Several existing works have used kinematic objectives for robotic planning [2, 3, 7, 8, 9]. Despite their effectiveness, these methods come with limitations. These methods are restricted to tabletop environments, and their representation of interactions with the environment is often simplified, e.g. by representing grip as a binary value or by predicting whether a collision will occur rather than directly controlling the nature of contact. Several works have performed robotic planning with the use of contact points [10, 11], but are often restricted to grasp generation. Many real-world tasks involve complex interactions with the environment that can significantly benefit from tactile information. This is particularly true for tasks requiring dexterous manipulation, such as grasping small objects, opening doors and drawers, pressing buttons, and twisting knobs [12, 13]. Contact and force are modalities that retain their significance across different environments and with various objects. This makes them more readily generalizable than other modalities such as vision or joint states.

We present a method for course-to-fine visual-force servoing. The idea of combined vision-force servoing has been explored with both classical and data-driven methods in limited settings [14, 15] such as peg-in-hole and contour following. The conceptualization of course-to-fine visual servoing

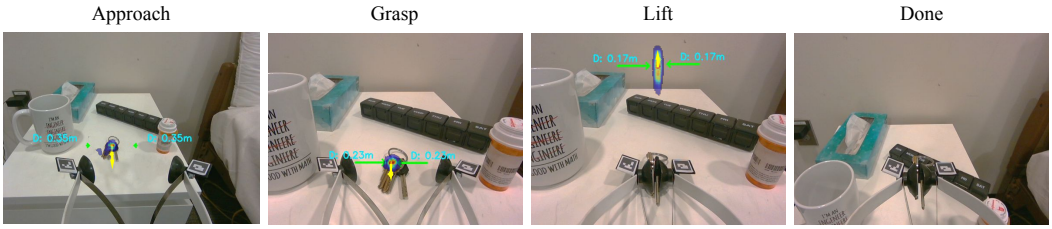

Figure 2: Task sequence for the key grasping task. Our representation of affordances includes future 3D contact points (green arrow tips), future grip force (green arrow magnitudes), and future resultant force (yellow arrow). The heatmap represents a probability distribution describing the future tool center point location. Action primitives, which are appended to the prompt, are shown above each keyframe.

has also been proposed in prior work using data-driven methods, where many initial poses lead to a singular "bottleneck pose" [5, 6, 16]. However, these data collection methods were only demonstrated to work for single-step tasks in tabletop environments with seen object instances. Our method instead chains together visual-force goals at human-annotated keyframes, which are similar to bottleneck poses, to complete more complex tasks. In contrast to methods with similar data collection schemes, ForceSight is capable of composing behaviors to complete multi-step tasks, functions in a variety of environments, and generalizes to objects semantically similar to those it has been trained on.

3 ForceSight: A Force-Based Robotic Planner

We present a system that represents robotic tasks as sequences of visual-force goals, where a goal includes target fingertip contact locations in image space and a target grip force and net force to be applied by the gripper.

3.1 Representing Affordances

We parameterize target contact locations associated with the next keyframe as $C = \{C_{xy}, C_z, W, \theta\}$. $C_{xy} \in \mathbb{Z}^2$ is a probability distribution in pixel space describing the likelihood that the gripper should move to each pixel location. By taking $\arg\max_{x,y} C_{xy}$, we generate a ray in Cartesian space that aligns with the tool center point of the gripper in the next keyframe. C_z signifies a depth estimation along this ray for a pinhole camera model, representing the estimated distance from the camera to the tool center point in the subsequent keyframe. W is the Euclidean distance between the predicted contact locations, and Y is the predicted yaw of the gripper with respect to the camera frame. From these values, we can derive the future 3D contact locations for both fingertips of the gripper.

The force applied by the gripper $F_R \in \mathbb{R}^3$ is the force in the next keyframe, and is measured by a force/torque sensor mounted to the wrist of the robot, transformed into the camera frame. We train a small neural network, parameterized as an MLP, to estimate the grasp force F_G given the gripper motor state and fingertip positions. To provide ground truth for the grip force model, we grasp a force/torque sensor at various grasp widths and record the measured magnitude of the force.

Given an RGBD observation and a text prompt, ForceSight predicts visual-force goals one keyframe into the future. To enhance the functionality of our system, we predict future contact locations within the current camera frame, thus making the network's predictions more amenable to visual servoing [17]. Moreover, this means predictions have the potential to be useful to other robots. For example, another two-fingered robot might achieve success using the fingertip and force goals either directly or with adaptations. The predicted tactile objectives remain consistent in a global frame, irrespective of the camera mounting location (Figure 6). Consequently, a low-level policy can be adapted to visually servo a robot with respect to these camera locations.

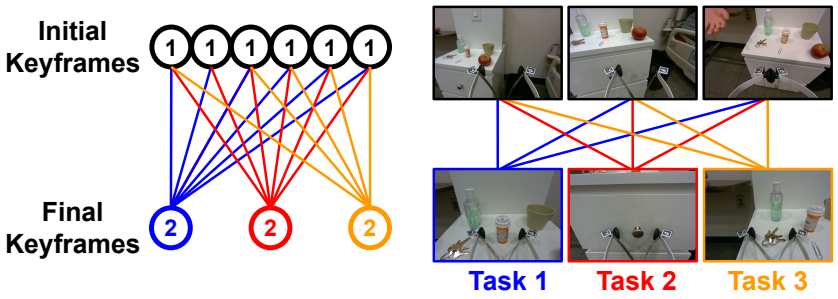

Figure 3: **Left:** During data collection, we partition each task into pairs of hand-specified keyframes. During each data collection session, we pair keyframes associated with several initial gripper poses to a single final keyframe. **Right:** We are also able to pair these initial keyframes with final frames associated with additional tasks.

3.2 Data Collection

We collect a dataset D, with $D_i = \{I, T, C, F_G, F_R\}$, where $I \in \mathbb{R}^{HxWx4}$ is an RGBD image captured by a gripper-mounted camera, T is a text prompt associated with a task along with an action primitive associated with the current timestep, $C \in \mathbb{R}^{2x3}$ is a set of two 3D contact locations in the camera frame associated with the next keyframe, $F_G \in \mathbb{R}$ is a scalar value representing the grasp force at the next keyframe, and $F_R \in \mathbb{R}^3$ is a vector in the camera frame representing the resultant force applied by the gripper at the next keyframe. Grasp force F_G is represented as a scalar value in Newtons, and is measured by the output of a small neural network as described in Section 3. These combined elements constitute the data points in the dataset, providing a comprehensive representation of a robot's interaction with its environment.

We partition each task into a series of keyframe pairs. In order to facilitate a course-to-fine process, keyframes are paired together in a many-to-one fashion, with many keyframes at a given timestep being paired with the same keyframe at the next timestep (Figure 3). This mapping allows our algorithm to be resilient to potentially imprecise predictions, thereby strengthening the robustness of the system.

To further enhance the efficiency of data collection, items relevant to other tasks were deliberately included in many initial frames as distractor objects. This strategy allows us to map identical initial images to a final keyframe for each task relevant to the input image, consequently generating substantially more data points per image in the dataset. Moreover, this has the dual benefit of bolstering the robustness of text conditioning.

Our data collection approach, therefore, not only enables a more streamlined and efficient process, but also ensures that our algorithm is exposed to a diverse set of environments and object configurations. We believe that this data collection methodology was key to achieving the necessary robustness to generalize across a wide range of real-world tasks and environments. Conducted over the course of 30 hours, our data collection process yielded over 26,000 high-quality keyframe pairs, the equivalent of approximately 10,000 distinct task demonstrations. We estimate that this would have taken approximately 160 hours if demonstrations were instead collected sequentially.

3.3 Architecture

Our proposed architecture, ForceSight, leverages a large-scale Vision Transformer [18] (ViT-large, 304M parameters) and a frozen T5 text encoder [19] to output precise visual-force goals (Figure 4). Visual-force goals are composed of fingertip contact locations, grip force, and resultant force, enabling robotic manipulation based on tactile objectives.

To harness the power of readily available internet-scale data, we initialize our vision transformer with weights from pre-training on ImageNet 21k [20]. We introduce an enhanced patch embedding layer that accepts RGBD inputs to this pre-trained network. To accommodate depth alongside RGB channels, we add a fourth input channel to the patch embedding projection, much like the early

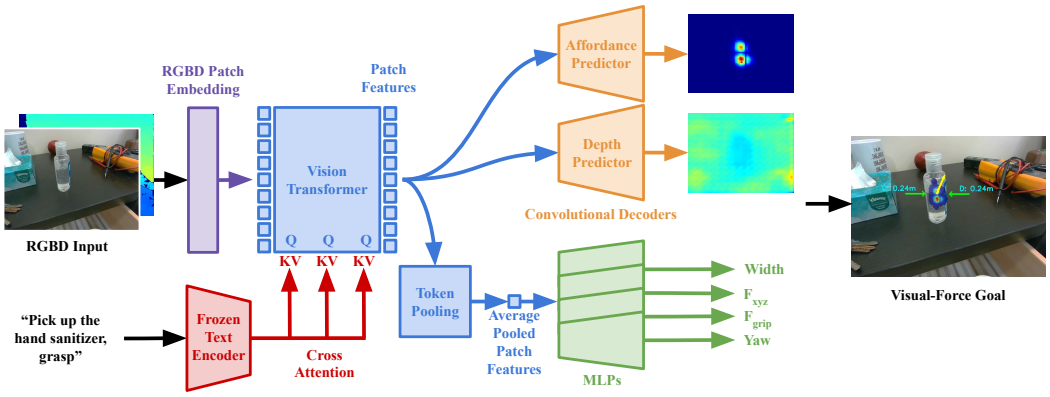

Figure 4: ForceSight is a text-conditioned RGBD vision transformer. An RGBD image is first divided into patches and passed into an RGBD-adapted patch encoder that transforms image patches into visual tokens. These visual tokens are then taken as input into a vision transformer. After every transformer block inside the vision transformer, the visual features are conditioned on text embeddings from a frozen text encoder via cross-attention to produce conditioned patch features. These patch features are passed into two simple convolutional decoders to produce an affordance map and a depth map. The patch features are additionally average pooled and passed into several MLPs in order to predict the grasp width, wrist force, grip force, and yaw.

fusion technique described in [21]. We initialize the weights of this additional channel with the average of the existing RGB channel weights, enabling smooth integration of depth information without forgetting information learned during pre-training.

The text-conditioned component of ForceSight relies on a frozen pre-trained T5 text encoder, which generates text embeddings that offer context for a given task. To effectively utilize this text information, we incorporate a cross-attention mechanism across all layers of the vision transformer. This approach allows the network to draw rich relationships between text prompts and visual features at multiple levels of abstraction. Inspired by developments in text-conditioned image generation [22], this operation applies cross-attention using learned projections of visual features as query vectors and projections of text features as key and value vectors.

The network output includes a 224x224 affordance map and a depth map. The ViT image encoder generates patch features associated with the input image's particular regions, which are transformed into an affordance heatmap and depth map through a convolutional decoder. We apply a weighted cross-entropy loss to the affordance map, translating the task into a pixel-wise classification problem (Equation 4). The depth prediction is supervised via an L_1 loss, masked by the ground truth affordance map, to focus only on the predicted affordance locations (Equation 5). This representation of affordances allows our network to propose multiple hypotheses, exhibiting robustness in complex scenarios where multiple visible objects are semantically relevant to the task.

Finally, the patch features are average-pooled and fed into several multi-layer perceptrons (MLPs). These MLPs estimate vital task execution parameters such as grasping width, yaw, wrist force, and grip force (Equations 6 - 9).

3.4 Action Primitives

The use of Large Language Models (LLMs) has demonstrated its efficacy in addressing sequential long-horizon robotic tasks [23, 24, 25, 26, 27]. In language-based tasks, it is often possible to identify shared subgoals across different tasks. By leveraging positive transfer of action subgoals among tasks, it becomes feasible to generalize primitive actions for robot tasks. To facilitate the transition between subgoals, we incorporate action primitives such as *approach, grasp, lift, and pull*, which are appended to the model prompt input. The inclusion of action primitives in ForceSight has proven effective in facilitating smooth transitions between different subgoals (Table 2).

Figure 5: For evaluating ForceSight and conducting ablations, we select a set of 10 tasks from real-world experiments with the Stretch RE1 robot, encompassing various objects and environments not present in the training set.

4 Experimental Results

We use a Stretch RE1 [28] from Hello Robot to conduct real-world experiments. To evaluate our model, we conduct experiments in held-out environments, including a mock bedroom and a real kitchen. We also exclusively interact with unseen object instances, i.e. objects which are semantically similar to those seen during training but are visually distinct from objects in the training dataset. We use the low-level controller described in Appendix F to command the robot.

4.1 Tasks and Environments

We select a set of 10 household tasks and perform 10 trials on each. For details about task success definitions, see Appendix A.

Task	Environment	Objects	Success Metrics
Pick up the apple	Kitchen	Apple, top drawer	Pick and lift
Pick up the medicine bottle	Bedroom	Medicine bottle, top drawer	Pick and lift
Pick up the keys	Bedroom	Keys, top drawer	Pick and lift
Pick up the paperclip	Bedroom	Paper clip, top drawer	Pick and lift
Pick up the hand sanitizer	Kitchen	Hand sanitizer, counter top	Pick and lift
Pick up the cup	Kitchen	Cup, counter top	Pick and lift
Place object in the trash	Bedroom	Medicine bottle, trash bin	Approach and ungrasp
Place object in the hand	Bedroom	Medicine bottle, real human hand	Place and ungrasp
Turn on the light	Atrium	Light switch	Push the light switch
Open the drawer	Bedroom	Bedside drawer	Pull the drawer

Table 1: Tasks, environments, objects and success metrics

4.2 Test Set Results vs. Baselines

We collect a test set in a mock bedroom environment containing several unseen objects (See Appendix A). The test set comprises a representative set of 60 keyframe pairs from each task. We evaluate performance on the test set by calculating the average contact distance, which is the average L_2 error between the predicted contact locations and the ground truth contact locations for each fingertip.

Representation	Avg. Contact Dist. (m)
Contact point regression	0.429
Contact point classification	0.107
Centroid class. + width + yaw	0.057
Centroid class. + width + yaw + Action Primitive	**0.036**

Table 2: Representation Ablations.

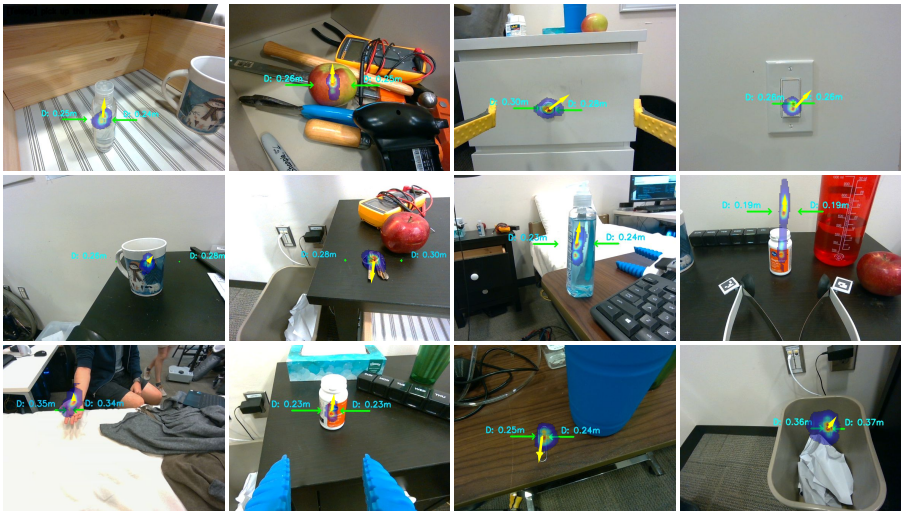

Figure 6: ForceSight is capable of providing accurate estimations of contact location, resultant forces, and grasping forces associated with a task in a variety of environments. Our network additionally generalizes to unseen object instances and does not depend on a specific camera setup.

4.3 Real-World Results

We run the model with a lower 3 Hz frequency for stability, providing kinematic and force objectives for visual-force servoing. Through these experiments, we show that our proposed ForceSight achieves a good success rate given unseen objects and environments.

Task	Task Success	Subgoal 1	Subgoal 2	Subgoal 3
Pick up the apple	100%	100%	100%	100%
Pick up the medicine bottle	70%	100%	90%	70%
Pick up the keys	60%	90%	90%	60%
Pick up the paperclip	60%	80%	80%	60%
Pick up the hand sanitizer	80%	100%	100%	80%
Pick up the cup	80%	100%	100%	80%
Place object in the trash	100%	100%	100%	-
Place object in the hand	100%	100%	100%	-
Turn off the light switch	70%	90%	70%	-
Open the drawer	90%	100%	90%	90%

Table 3: Real-word task success rates with ForceSight, Tasks are composed of multiple action primitive subgoals. Picking task comprises: *approach, grasp, lift*; placing task: *approach, ungrasp*; turn on light switch task: *approach, push*; and open the drawer task: *approach, grasp, pull*.

4.4 Ablations

According to the results presented in Table 4, we demonstrate the significance of force as a modality for successfully executing tactile tasks, such as picking up a paper clip. Our findings indicate that by using the ForceSight planner, which incorporates force-related objective information, the robot achieves improved performance in these tasks. Conversely, when the low-level controller ignores the force objective information, the robot's ability to successfully complete the task is compromised.

Furthermore, we observed that training the model without depth input leads to a tendency for the robot to prematurely grasp the object. This outcome can be attributed to the lack of depth perception, which affects the robot's ability to accurately assess the object's position and make informed grasping decisions. In addition, data augmentation proves to be an effective approach for enhancing performance in diverse environments with unseen objects. In 6, the model also shows generalizability of visual-force goals in unseen instances.

	Task Success	Avg. Contact Dist.	RMSE Resultant Force	RMSE Grip Force
ForceSight (Ours)	**81%**	**0.036**	**0.404**	1.524
w/o forces	50%	-	-	-
w/o depth	20%	0.063	1.493	1.32
w/o pretraining	20%	0.078	0.583	1.576
w/o augmentation	25%	0.049	1.181	1.759
w/o text conditioning	0 %	0.075	0.907	**1.26**

Table 4: Ablations of ForceSight. Each ablation is tested with 2 trials for each of the 10 real-world tasks for a total of 20 trials per ablation. For more details on success rate see Appendix A. The contact distance is in meters and force RMSE is in Newtons.

5 Limitations

While ForceSight exhibits encouraging performance across diverse and challenging tasks, we recognize certain limitations and areas for future improvements in our current methodology.

Our current study focuses on a rather limited set of tasks, predominantly consisting of traditional pick-and-place. Although we demonstrate efficacy in these contexts, it is essential to expand this validation to a more comprehensive suite of tasks in future work. Given the inherent scalability of transformer models and the efficiency of our data collection procedure, we are optimistic that our methodology could be generalized to a much broader range of complex tasks.

One of the recurrent failure modes observed in our ForceSight model pertains to inaccuracies in depth predictions, particularly those further away from the camera (1m). Despite the adaptive capabilities of visual servoing that can account for some discrepancies as errors diminish at closer distances, inaccurate depth prediction remains a primary cause of task execution failure. We plan on exploring additional representations of depth, which may improve the precision of the predictions and overall task execution.

Another limitation of the current ForceSight model is the requirement for targets to be within the camera's field of view, which limits performance on some tasks. This constraint may limit task performance in scenarios such as drawer opening, where predictions are sometimes clipped to the image's edge. Modifications to the current model or the integration of additional sensory inputs may be explored in future works to overcome this constraint.

While our model is adaptable to various robot manipulators and camera setups, we have only carried out comprehensive real-world experimental setups using the RE1 robot. To further evaluate the versatility and robustness of our method, future studies should extend these experiments to various end-effectors and robotic manipulators.

Finally, our keyframe representations do not provide complete information about the gripper's pose, as the gripper's pitch and roll are assumed to be constant values. However, this limitation can be addressed by adding additional heads to the model's output to specify these parameters. We believe that the incorporation of these additional pose parameters will further enhance the performance and generality of the model.

6 Conclusion

We presented ForceSight, a text-conditioned robotic planner that generates tactile and kinematic goals to enable the execution of multiple contact-rich tasks, generalizing to unseen environments and new object instances. We demonstrated the usefulness of ForceSight with 10 robotic tasks, and show that the use of tactile objectives improves performance on these tasks.

References

[1] 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.

[2] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[3] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[4] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[5] E. Valassakis, G. Papagiannis, N. Di Palo, and E. Johns. Demonstrate once, imitate immediately (dome): Learning visual servoing for one-shot imitation learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8614–8621. IEEE, 2022.

[6] E. Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021.

[7] T. D. Kulkarni, A. Gupta, C. Ionescu, S. Borgeaud, M. Reynolds, A. Zisserman, and V. Mnih. Unsupervised learning of object keypoints for perception and control. *Advances in neural information processing systems*, 32, 2019.

[8] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *Robotics Research: The 19th International Symposium ISRR*, pages 132–157. Springer, 2022.

[9] S. James, K. Wada, T. Laidlow, and A. J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.

[10] Z. Xue, J. M. Zoellner, and R. Dillmann. Grasp planning: Find the contact points. In *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 835–840. IEEE, 2007.

[11] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.

[12] P. Grady, J. A. Collins, S. Brahmbhatt, C. D. Twigg, C. Tang, J. Hays, and C. C. Kemp. Visual pressure estimation and control for soft robotic grippers. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3628–3635. IEEE, 2022.

[13] J. A. Collins, C. Houff, P. Grady, and C. C. Kemp. Visual contact pressure estimation for grippers in the wild. *arXiv preprint arXiv:2303.07344*, 2023.

[14] J. Baeten, H. Bruyninckx, and J. De Schutter. Integrated vision/force robotic servoing in the task frame formalism. *The International Journal of Robotics Research*, 22(10-11):941–954, 2003.

[15] K. Almaghout, R. A. Boby, M. Othman, A. Shaarawy, and A. Klimchik. Robotic pick and assembly using deep learning and hybrid vision/force control. In *2021 International Conference" Nonlinearity, Information and Robotics"(NIR)*, pages 1–6. IEEE, 2021.

[16] B.-S. Lu, T.-I. Chen, H.-Y. Lee, and W. H. Hsu. Cfvs: Coarse-to-fine visual servoing for 6-dof object-agnostic peg-in-hole assembly. *arXiv preprint arXiv:2209.08864*, 2022.

[17] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[20] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[21] G. Tziafas and H. Kasaei. Early or late fusion matters: Efficient rgb-d fusion in vision transformers for 3d object recognition.

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

[23] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg. Text2motion: From natural language instructions to feasible plans. *arXiv preprint arXiv:2303.12153*, 2023.

[24] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor. Chatgpt for robotics: Design principles and model abilities. *Microsoft Auton. Syst. Robot. Res*, 2:20, 2023.

[25] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[26] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[27] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[28] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich. The design of stretch: A compact, lightweight mobile manipulator for indoor human environments. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3150–3157. IEEE, 2022.

[29] Intel® Realsense™ D405 – intelrealsense.com. https://www.intelrealsense.com/depth-camera-d405/.

[30] ATI Industrial Automation. F/T Sensor: mini45, 2022. URL https://www.ati-ia.com/products/ft/ft_models.aspx?id=mini45.

[31] M. Liu, M. Zhu, and W. Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

[32] J. A. Collins, P. Grady, and C. C. Kemp. Force/torque sensing for soft grippers using an external camera. *arXiv preprint arXiv:2210.00051*, 2022.

[33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Appendix

A. Task Definition

For tasks that involve picking (e.g. *"pick up the apple"*), the robot should successfully approach, grasp, and lift the target object above the surface for more than 5 seconds.

For placing tasks (e.g. *"place the object in the hand"*), the robot should successfully approach the desired target object, ungrasp and place the held object to the target object. In this case, into a trash can or a static human hand.

For the *"turn off the light switch"* task, the light switch should be completely pressed in to be a success.

Finally, for *"open the drawer"* task, the robot must successfully pull the drawer open, extending to its full range of motion.

Additionally, we label the execution of a task a failure if the trial does not terminate within a 1-minute time window.

B. Data Collection Setup

In order to collect visual data, we mount an Intel® RealSense™ D405 [29] to the robot's gripper to capture the RGBD image I. The fingertips are first located in the image via ArUco tags attached to the gripper, and a transformation is applied to map these to the point at the center of the fingertip surface, which we call the contact locations C. The contact locations are mapped from the future camera frame to the current frame by utilizing the robot's forward kinematics. We use a Stretch mobile manipulator for data collection. Certainly, this can also be substituted with different data collection setup as long as the ground truth future fingertip location can be obtained.

The resultant force applied to the gripper is measured by a wrist-mounted force/torque sensor [30]. This sensor provides accurate readings of the forces exerted by the gripper, enabling our system to predict future forces and enabling the robot to perform tasks requiring force control.

C. Details on Real-world Experiments

	Overall Task Success	**Subgoal 1**	**Subgoal 2**	**Subgoal 3**
ForceSight (Ours)	81/100 (81%)	96/100 (96%)	93/96 (97%)	54/65 (83%)
w/o forces	10/20 (50%)	13/20 (65%)	11/13 (85%)	6/7 (86%)
w/o augmentation	5/20 (25%)	11/20 (55%)	7/11 (64%)	5/7 (71%)
w/o depth	4/20 (20%)	8/20 (40%)	6/8 (75%)	3/3 (100%)
w/o pretraining	4/20 (20%)	5/20 (25%)	4/5 (80%)	3/3 (100%)
w/o text conditioning	0/20 (0%)	9/20 (45%)	0/9 (0%)	-

Table 5: The detailed task success rate for ForceSight ablation on real-world experiments.

From the analysis presented in Table 5, it is evident that ForceSight outperforms other approaches in terms of task success rate. Specifically, when the low-level controller disregards the force goals, both *subgoal 1* and *subgoal 2* exhibit lower success rates. This can be attributed to the robot's inability to accurately approach and grip the target object. For example, the gripper will prematurely grasp the drawer's handle without applying force to the drawer during the approach stage. Additionally, the absence of data augmentation adversely affects the model's ability to perceive new objects, as evidenced by its struggles in recognizing unseen objects such as the green apple and the black drawer. By excluding depth information from the model, the success rate of *subgoal 1* decreases, primarily because this stage ("approach" action) heavily relies on accurate depth perception. Moreover, pretraining proves to be a valuable technique for enabling the robot to learn with minimal data. Interestingly, when text conditioning is removed, the model can successfully detect interesting objects as affordances (9 out of 20 in *subgoal 1*), but fails to determine the appropriate actions to perform with the target object, seen in *subgoal 2*.

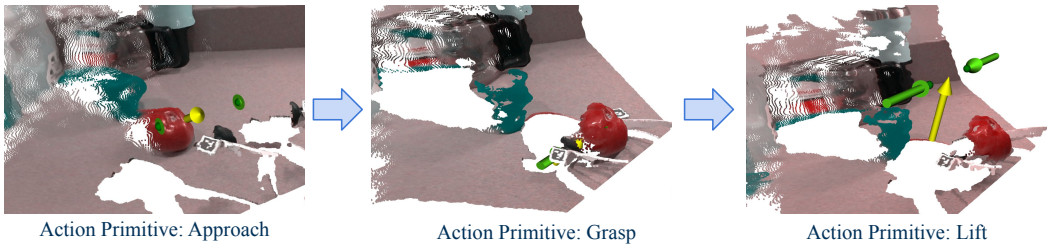

Figure 7: Force objectives prediction in 3D perspective when conducting "pick up the apple" task in a kitchen environment. Yellow arrow represents wrist force and green arrows represent grip force acting on the target object.

Figure 8: **Left:** Objects present in the training set. **Right:** Objects present in the test set and ablation studies.

D. Emergent Properties

After training, we observed multiple surprising behaviors exhibited by ForceSight. Notably, ForceSight demonstrated a surprising flexibility and adaptability in its treatment of action primitives in relation to various objects. For instance, ForceSight managed to apply certain action primitives to objects without any explicit representation of such actions during the training phase. For example, if given the action primitive "grasp" in combination with a prompt concerning a light switch or the action primitive "push" in combination with a prompt concerning a medicine bottle, the model responded with predictions corresponding to the specified actions, despite these specific actions being absent in the training data.

We also observe that ForceSight is agnostic to the gripper it is observing; although during training we only show the network a single gripper, it learns to completely ignore the visual features associated with the gripper, and the predictions were observed to be invariant to the type of gripper the RGBD camera was mounted to, and even works without any gripper present in the field of view.

E. Determining Output Representation

We experimented with three output representations of the contact locations, namely: Regression-based, Pixel-Space Contact points, and Pixel-Space Centroid output representation.

The regression-based representation directly minimizes the L_1 error of the contact point locations. We observed this representation to work well in simplified environments with a single object, but it struggled in situations where there were multiple plausible contact points, especially when two semantically relevant objects were present in the input image. We hypothesize that this is because the L_1 objective encourages the model to find the mean of the target distribution of plausible contact locations. In this situation where this distribution is multimodal, the model produces estimates that average out the different modes of the distribution, resulting in a model that provides suboptimal predictions for all modes.

To tackle this issue of estimating 3D locations in a multimodal distribution, we instead formulated our optimization as a classification problem. To additionally take advantage of the knowledge present in our pretrained ViT, we decouple the representation into pixel-wise classification and depth-wise regression. This has the additional benefit of highly interpretable affordance predictions, as the affordance prediction is a distribution that may be overlayed on the input image. We initially treated contact points individually, but found that locations associated with the left and right fingertip didn't necessarily correspond to one another, often leading to physically implausible predictions due to the lack of correspondence between contact locations.

To address this correspondence issue, we reparameterized our affordance representations to instead classify the pixel location associated with the center of the contact points, perform regression on the depths associated with the pixel location, and additionally predict the grasping width and the yaw so that the individual contact locations could be derived. The ablation of the output representation is shown in Table 2.

F. Overall System Architecture

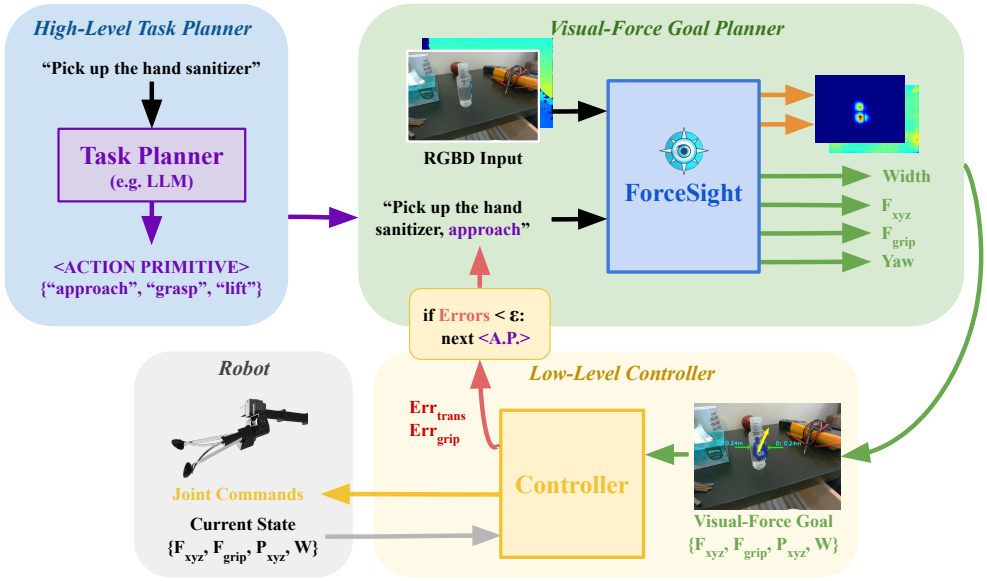

Figure 9: Overall System Architecture for Force Sight.

The ForceSight system architecture comprises several components that work together to accomplish a text-conditioned task. It begins with a High-level Task Planner, which takes a text input and generates a sequence of action primitives representing subgoals. These action primitives, along with the RGBD input, are then passed to the ForceSight transformer model.

The ForceSight model processes the input and produces force-based objectives. These objectives are subsequently fed into the low-level controller, which generates joint motion commands for the robot to execute the task and its action primitive. This is done by using visual-servo control.

To determine when to switch to the next action primitive, the low-level controller compares the error between the current states and visual-force goals with a predefined threshold. If the error is below the threshold, the low-level controller initiates the switch to the next action primitive.

This entire process loop operates at a frequency of 8Hz, allowing for multiple iterations until the task is completed successfully.

G. Low-Level Controller

Our low-level controller receives 3D target contact and force goals located in the camera frame from ForceSight and executes an action after each image frame observation in order to achieve these goals. Using an eye-in-hand camera, the low-level controller uses visual-force servoing to move the gripper closer to the visual-force goal. This approach reduces relative error and is insensitive to global calibration.

In our low-level control framework, we divide operations into two main components: Translation Control and Gripper Control.

End-Effector Control: Minimizing Euclidean Translation and Wrist Force Error Our aim here is to minimize both the Euclidean translation and the wrist force error along the XYZ axis:

$$\begin{aligned} E_{\text{wrist, translation}} &= ||T_{\text{wrist, predicted}} - T_{\text{wrist, current}}||_2 \\ E_{\text{wrist, force}} &= ||F_{\text{wrist, predicted}} - F_{\text{wrist, current}}||_2 \end{aligned} \tag{1}$$

Where $T_{current} \in \mathbb{R}^3$ is the current translation of the fingertip centroid; $T_{predicted} \in \mathbb{R}^3$ is the predicted translation of the fingertip centroid; $F_{current} \in \mathbb{R}^3$ is the current force; $F_{predicted} \in \mathbb{R}^3$ is the predicted force.

Grasping Control: Minimizing Grasp Width and Grip Force Error For this objective, the goal is to minimize discrepancies in the grasp width and grip force, expressed as:

$$\begin{aligned} E_{\text{width}} &= ||W_{\text{predicted}} - W_{\text{current}}||_2 \\ E_{\text{grip force}} &= ||F_{\text{predicted}} - F_{\text{current}}||_2 \end{aligned} \tag{2}$$

Where $W_{current} \in \mathbb{R}$ is the current grasp width; $W_{predicted} \in \mathbb{R}^3$ is the predicted grasp width; $F_{current} \in \mathbb{R}$ is the current grip force; $F_{predicted} \in \mathbb{R}$ is the predicted grip force. The goal is to minimize the kinematic and force error in all directions in a step-wise manner. To better incorporate both kinematic and force modalities into the movement error, we define a joint objective combining both errors. This is expressed as a movement control value, M in cartesian space, and then is executed by the controller in a step-wise manner.

$$\begin{aligned} M_{\text{end-effector}} &= E_{\text{translation}} + \lambda_{\text{wrist}} E_{\text{wrist force}} \\ M_{\text{gripper}} &= E_{\text{width}} + \lambda_{\text{grip}} E_{\text{grip force}} \end{aligned} \tag{3}$$

The movement value determines the necessary movement that the robot should execute in order to reach the predicted objectives. By combining both kinematic and force objectives, the task can be executed more delicately, leading to improved performance. This approach is particularly useful in determining whether the current action primitive has been successfully completed, and whether the next action primitive should be passed as an input to ForceSight. Results are shown in Table 4.

We recognize that there are many options for implementing the low-level controller; for example, our policy can easily be substituted for a goal-conditioned reinforcement learning policy [31]. We could also plausibly replace the F/T sensor with a system that estimates these values from motor current or from vision [32, 13].

H. Training Details

We train ForceSight for 20 epochs using the Adam optimizer [33], corresponding to a total of 500,000 iterations. Our training procedure processes 224x224 RGBD images in batches of eight with a learning rate of 5e-5.

ForceSight's loss function is as follows:

$$L_A = -(\beta * A log(\hat{A}) + (1 - A) log(1 - \hat{A})) \tag{4}$$

$$L_D = A * ||D - \hat{D}||_1 \tag{5}$$

$$L_F = ||F_R - \hat{F_R}||_2 \tag{6}$$

$$L_G = ||F_G - \hat{F_G}||_2 \tag{7}$$

$$L_W = ||W - \hat{W}||_2 \tag{8}$$

$$L_Y = ||Y - \hat{Y}||_2 \tag{9}$$

$$\tag{10}$$

ForceSight is trained with the weighted sum of the individual losses:

$$L = \lambda_A L_A + \lambda_D L_D + \lambda_F L_F + \lambda_G L_G + \lambda_W L_W + \lambda_Y L_Y \tag{11}$$

Balancing the impact of distinct loss components, we assign coefficients as follows: $\lambda_A = 1, \lambda_D = 5e4, \lambda_F = 0.2, \lambda_G = 0.2, \lambda_W = 0.2$, and $\lambda_Y = 0.2$. In our weighted cross-entropy loss associated with affordance map prediction (Equation 4), we place a stronger emphasis on the correct localization of contact points by assigning a β value of 100, emphasizing the importance of correctly identifying locations of interest.

Our ground truths for pixel-wise classification take the form of multi-hot encodings, where circles with a radius of 10 pixels pinpoint the tool center point's coordinates. We found that this denser representation of the ground truth produced rich heatmaps while maintaining good performance.

To further improve the robustness of our model and encourage generalization, we apply brightness, saturation, contrast, and hue augmentation to the RGB channels of the input.

We also include a data filtering step that excludes examples where the ground truth lies outside the camera's field of view. This process minimizes potential inaccuracies that could arise from attempting to predict beyond the field of view.

Additionally, we find that appending subgoals to the text prompts enhances the model's performance. This strategy clearly defines the boundaries between keyframes, offering a more granular approach that improves performance (Table 2).