

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.

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

14 **Keywords:** Transformers, Imitation Learning, Manipulation, Force Sensing, Lan-
15 guage Grounding

16 1 Introduction

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

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

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

32 We present the following contributions:

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

42 2 Related Work

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

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

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

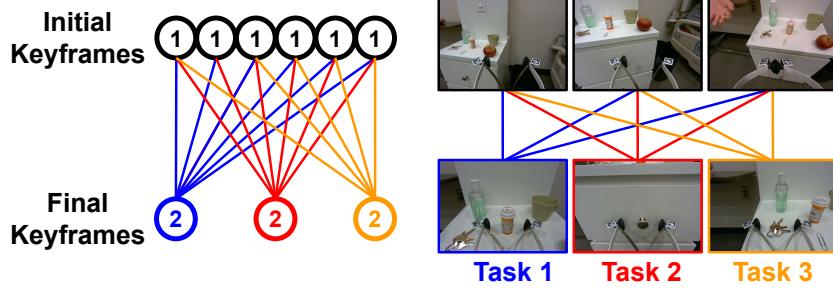


Figure 3: **Left:** During data collection, we partition each task into pairs of hand-specified keyframes. During each data collection session, we pair keyframes associated with several initial gripper poses to a single final keyframe. **Right:** We are also able to pair these initial keyframes with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

105 3.2 Data Collection

106 We collect a dataset D , with $D_i = \{I, T, C, F_G, F_R\}$, where $I \in \mathbb{R}^{H \times W \times 4}$ is an RGBD image
 107 captured by a gripper-mounted camera, T is a text prompt associated with a task along with an
 108 action primitive associated with the current timestep, $C \in \mathbb{R}^{2 \times 3}$ is a set of two 3D contact locations
 109 in the camera frame associated with the next keyframe, $F_G \in \mathbb{R}$ is a scalar value representing the
 110 grasp force at the next keyframe, and $F_R \in \mathbb{R}^3$ is a vector in the camera frame representing the
 111 resultant force applied by the gripper at the next keyframe. Grasp force F_G is represented as a scalar
 112 value in Newtons, and is measured by the output of a small neural network as described in Section
 113 3. These combined elements constitute the data points in the dataset, providing a comprehensive
 114 representation of a robot’s interaction with its environment.

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

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

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

132 3.3 Architecture

133 Our proposed architecture, ForceSight, leverages a large-scale Vision Transformer [18] (ViT-large,
 134 304M parameters) and a frozen T5 text encoder [19] to output precise visual-force goals (Figure
 135 4). Visual-force goals are composed of fingertip contact locations, grip force, and resultant force,
 136 enabling robotic manipulation based on tactile objectives.

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

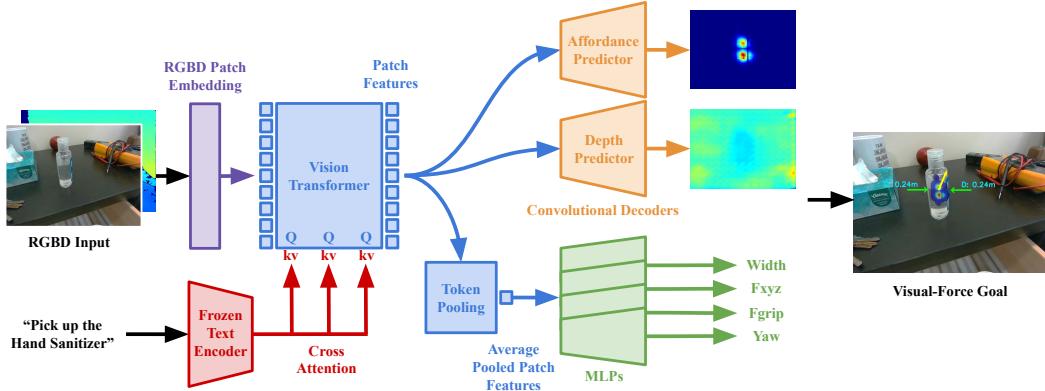


Figure 4: ForceSight is a text-conditioned RGBD vision transformer. An RGBD image is first divided into patches and passed into an RGBD-adapted patch encoder that transforms image patches into visual tokens. These visual tokens are then taken as input into a vision transformer. After every transformer 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.

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

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

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

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

162 3.4 Action Primitives

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

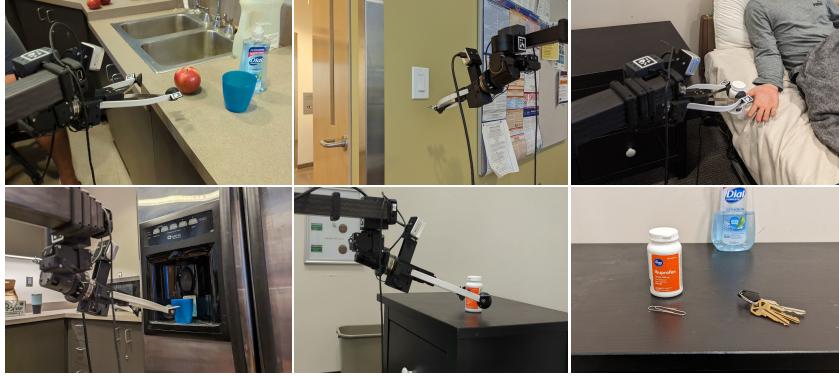


Figure 5: For evaluating ForceSight and conducting ablations, we select a set of 10 tasks from real-world experiments with the Stretch RE1 robot, encompassing various objects and environments not present in the training set.

170 4 Experimental Results

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

176 4.1 Tasks and Environments

177 We select a set of 10 household tasks and perform 10 trials on each. For details about task success
 178 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

179 4.2 Test Set Results vs. Baselines

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

Representation	Avg. Contact Dist.
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.

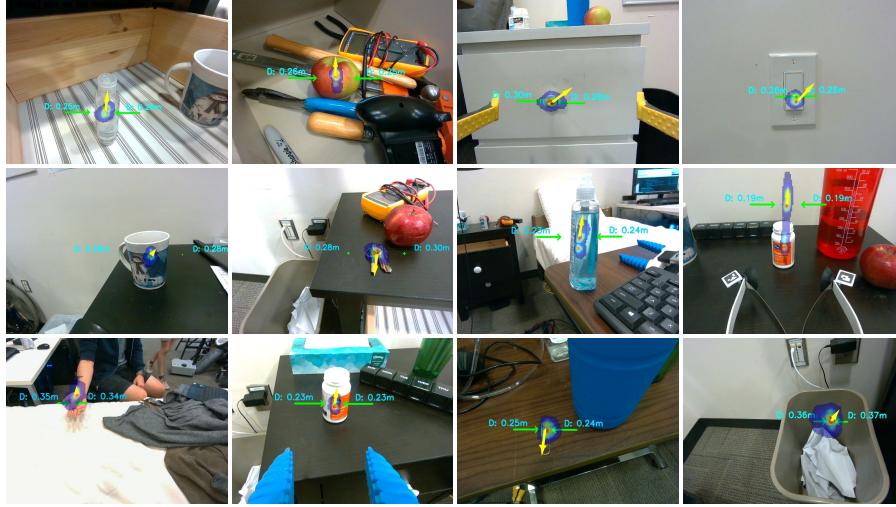


Figure 6: ForceSight is capable of providing accurate estimations of contact location, resultant forces, and grasping forces associated with a task in a variety of environments. Our network additionally generalizes to unseen object instances and does not depend on a specific camera setup.

185 4.3 Real-World Results

186 We run the model with a lower 3 Hz frequency for stability, providing kinematic and force objec-
 187 tives for visual-force servoing. Through these experiments, we show that our proposed ForceSight
 188 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*.

189 4.4 Ablations

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

195 Furthermore, we observed that training the model without depth input leads to a tendency for the
 196 robot to prematurely grasp the object. This outcome can be attributed to the lack of depth percep-
 197 tion, which affects the robot’s ability to accurately assess the object’s position and make informed
 198 grasping decisions. In addition, data augmentation proves to be an effective approach for enhancing
 199 performance in diverse environments with unseen objects. In 6, the model also shows generalizabil-
 200 ity 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.

201 5 Limitations

202 While ForceSight exhibits encouraging performance across diverse and challenging tasks, we rec-
 203 ognize certain limitations and areas for future improvements in our current methodology.

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

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

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

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

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

229 6 Conclusion

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

234 **References**

- 235 [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.
- 236
- 237
- 238 [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.
- 239
- 240 [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.
- 241
- 242 [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.
- 243
- 244
- 245 [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.
- 246
- 247
- 248
- 249 [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.
- 250
- 251
- 252 [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.
- 253
- 254
- 255 [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.
- 256
- 257
- 258 [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.
- 259
- 260
- 261 [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.
- 262
- 263
- 264 [11] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspsnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.
- 265
- 266
- 267 [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.
- 268
- 269
- 270 [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.
- 271
- 272 [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.
- 273
- 274
- 275 [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.
- 276
- 277

- 278 [16] B.-S. Lu, T.-I. Chen, H.-Y. Lee, and W. H. Hsu. Cfvs: Coarse-to-fine visual servoing for 6-dof
 279 object-agnostic peg-in-hole assembly. *arXiv preprint arXiv:2209.08864*, 2022.
- 280 [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.
- 282 [18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. De-
 283 hghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transform-
 284 ers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- 285 [19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu.
 286 Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of
 287 Machine Learning Research*, 21(1):5485–5551, 2020.
- 288 [20] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor. Imagenet-21k pretraining for the
 289 masses. *arXiv preprint arXiv:2104.10972*, 2021.
- 290 [21] G. Tzifas and H. Kasaei. Early or late fusion matters: Efficient rgb-d fusion in vision trans-
 291 formers for 3d object recognition.
- 292 [22] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image syn-
 293 thesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer
 294 Vision and Pattern Recognition*, pages 10684–10695, 2022.
- 295 [23] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg. Text2motion: From natural language
 296 instructions to feasible plans. *arXiv preprint arXiv:2303.12153*, 2023.
- 297 [24] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor. Chatgpt for robotics: Design principles
 298 and model abilities. *Microsoft Auton. Syst. Robot. Res*, 2:20, 2023.
- 299 [25] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan,
 300 K. Hausman, A. Herzog, et al. Do as i can, not as i say: Grounding language in robotic
 301 affordances. *arXiv preprint arXiv:2204.01691*, 2022.
- 302 [26] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson,
 303 Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint
 304 arXiv:2303.03378*, 2023.
- 305 [27] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch,
 306 Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language
 307 models. *arXiv preprint arXiv:2207.05608*, 2022.
- 308 [28] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich. The design of stretch: A com-
 309 pact, lightweight mobile manipulator for indoor human environments. In *2022 International
 310 Conference on Robotics and Automation (ICRA)*, pages 3150–3157. IEEE, 2022.
- 311 [29] Intel® Realsense™ D405 – intelrealsense.com. <https://www.intelrealsense.com/depth-camera-d405/>.
- 313 [30] ATI Industrial Automation. F/T Sensor: mini45, 2022. URL https://www.ati-ia.com/products/ft/ft_models.aspx?id=mini45.
- 315 [31] M. Liu, M. Zhu, and W. Zhang. Goal-conditioned reinforcement learning: Problems and
 316 solutions. *arXiv preprint arXiv:2201.08299*, 2022.
- 317 [32] J. A. Collins, P. Grady, and C. C. Kemp. Force/torque sensing for soft grippers using an
 318 external camera. *arXiv preprint arXiv:2210.00051*, 2022.
- 319 [33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint
 320 arXiv:1412.6980*, 2014.

321 **Appendix**

322 **A. Task Definition**

- 323 For tasks that involve picking (e.g. “*pick up the apple*”), the robot should successfully approach,
 324 grasp, and lift the target object above the surface for more than 5 seconds.
- 325 For placing tasks (e.g. “*place the object in the hand*”), the robot should successfully approach the
 326 desired target object, ungrasp and place the held object to the target object. In this case, into a trash
 327 can or a static human hand.
- 328 For the “*turn off the light switch*” task, the light switch should be completely pressed in to be a
 329 success.
- 330 Finally, for “*open the drawer*” task, the robot must successfully pull the drawer open, extending to
 331 its full range of motion.
- 332 Additionally, we label the execution of a task a failure if the trial does not terminate within a 1-
 333 minute time window.

334 **B. Data Collection Setup**

- 335 In order to collect visual data, we mount an Intel® RealSense™ D405 [29] to the robot’s gripper to
 336 capture the RGBD image I . The fingertips are first located in the image via ArUco tags attached to
 337 the gripper, and a transformation is applied to map these to the point at the center of the fingertip
 338 surface, which we call the contact locations C . The contact locations are mapped from the future
 339 camera frame to the current frame by utilizing the robot’s forward kinematics. We use a Stretch
 340 mobile manipulator for data collection. Certainly, this can also be substituted with different data
 341 collection setup as long as the ground truth future fingertip location can be obtained.
- 342 The resultant force applied to the gripper is measured by a wrist-mounted force/torque sensor [30].
 343 This sensor provides accurate readings of the forces exerted by the gripper, enabling our system to
 344 predict future forces and enabling the robot to perform tasks requiring force control.

345 **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	11/13	6/7
w/o augmentation	5/20 (25%)	11/20	7/11	5/7
w/o depth	4/20 (20%)	8/20	6/8	3/3
w/o pretraining	4/20 (20%)	5/20	4/5	3/3
w/o text conditioning	0/20 (0%)	9/20	0/9	-

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

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

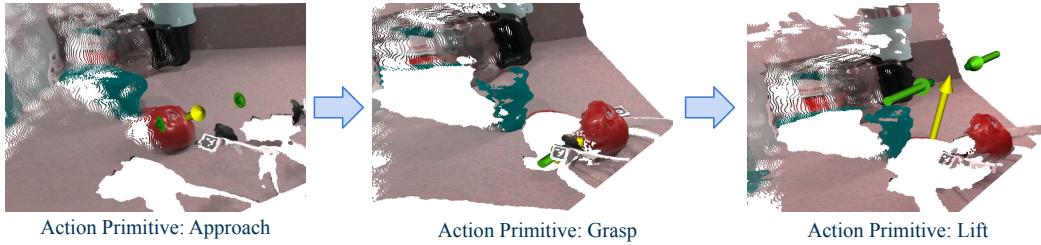


Figure 7: Force objectives prediction in 3D perspective when conducting “pick up the apple” task in a kitchen environment. Yellow arrow represents wrist force and green arrows represent grip force acting on the target object.



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

359 D. Emergent Properties

360 After training, we observed multiple surprising behaviors exhibited by ForceSight. Notably, Force-
 361 Sight demonstrated a surprising flexibility and adaptability in its treatment of action primitives in
 362 relation to various objects. For instance, ForceSight managed to apply certain action primitives to
 363 objects without any explicit representation of such actions during the training phase. For example,
 364 if given the action primitive “grasp” in combination with a prompt concerning a light switch or the
 365 action primitive “push” in combination with a prompt concerning a medicine bottle, the model re-
 366 sponded with predictions corresponding to the specified actions, despite these specific actions being
 367 absent in the training data.

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

372 E. Determining Output Representation

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

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

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

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

396 F. Low-Level Controller

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

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

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

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

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

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

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

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

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

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

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

426 G. Training Details

427 We train ForceSight for 20 epochs using the Adam optimizer [33], corresponding to a total of
 428 500,000 iterations. Our training procedure processes 224x224 RGBD images in batches of eight
 429 with a learning rate of 5e-5.

430 ForceSight’s loss function is as follows:

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

$$L_D = A * \|D - \hat{D}\|_1 \quad (5)$$

$$L_F = \|F_R - \hat{F}_R\|_2 \quad (6)$$

$$L_G = \|F_G - \hat{F}_G\|_2 \quad (7)$$

$$L_W = \|W - \hat{W}\|_2 \quad (8)$$

$$L_Y = \|Y - \hat{Y}\|_2 \quad (9)$$

$$(10)$$

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

$$L = \lambda_A L_A + \lambda_D L_D + \lambda_F L_F + \lambda_G L_G + \lambda_W L_W + \lambda_Y L_Y \quad (11)$$

432 Balancing the impact of distinct loss components, we assign coefficients as follows: $\lambda_A = 1$, $\lambda_D =$
 433 $5e4$, $\lambda_F = 0.2$, $\lambda_G = 0.2$, $\lambda_W = 0.2$, and $\lambda_Y = 0.2$. In our weighted cross-entropy loss associated
 434 with affordance map prediction (Equation 4), we place a stronger emphasis on the correct local-
 435 ization of contact points by assigning a β value of 100, emphasizing the importance of correctly
 436 identifying locations of interest.

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

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

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

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