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

Jeremy A. Collins[1*], Cody Houff[1*], You Liang Tan[1*], Charles C. Kemp[1]
[*]Equal contribution

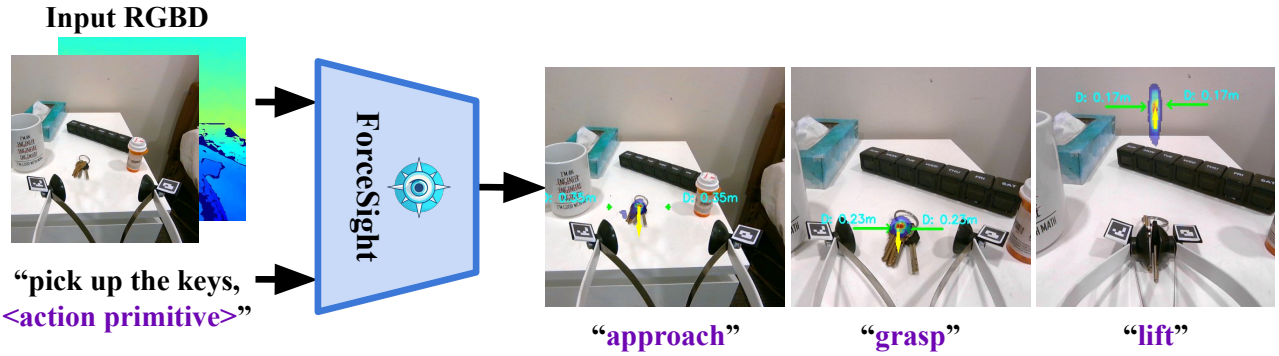

Fig. 1. ForceSight is an RGBD-adapted, text-conditioned vision transformer. Given an RGBD image and a text prompt, ForceSight produces visual-force goals for a mobile manipulator. Action primitives, shown below each image, are appended to the text input by a simple low-level controller. D refers to the estimated depth of the goal location in meters.

***Abstract*— We present ForceSight, a system for multi-task, text-guided mobile manipulation that predicts visual-force goals using a deep neural network. Using a single RGBD image combined with a text prompt, ForceSight determines a target end-effector pose in the camera frame (kinematic goal) and the associated forces (force goal). Together, these two components form a visual-force goal. Prior work has demonstrated that deep models outputting human-interpretable kinematic goals can enable dexterous manipulation by real robots. Forces are critical to manipulation, yet have typically been relegated to lower-level execution in these systems. When deployed on a mobile manipulator equipped with an eye-in-hand RGBD camera, ForceSight performed tasks such as precision grasps, drawer opening, and object handovers with an 81% success rate in unseen environments with object instances that differed significantly from the training data. In a separate experiment, relying exclusively on visual servoing and ignoring force goals dropped the success rate from 90% to 45%, demonstrating that force goals can significantly enhance performance. Paper appendix, videos, code, and trained models are available at https://force-sight.github.io/.**

I. 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 or flat object off a surface can benefit from first making fingertip contact with the surface and then sliding the fingertips across the surface to pick up the object [1]. 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 and applied force provide strong cues that appropriate contact has been achieved for a task.

We introduce ForceSight, a transformer-based, text-conditioned robotic planner that outputs visual-force goals, enabling the human-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 and output a visual-force goal relevant to the current subtask. A visual-force goal consists of a kinematic goal and a force goal. The kinematic goal specifies a target configuration for the end-effector as a 3D position, a yaw angle, and the distance between the fingertips. The force goal specifies a target grip force and a target applied force measured by a wrist-mounted force-torque sensor.

We present the following contributions:

- **Deep Model to Infer Visual-Force Goals:** We present a deep model that infers visual-force goals, given an RGBD image from an eye-in-hand camera and a natural language prompt, and show that force goals significantly improve performance over visual servoing alone.
- **System to Perform Real-World Tasks:** We present a system that uses visual-force goals from the deep model to perform a variety of tasks with unseen object instances in novel environments.
- **Open Source:** We release our code, dataset, and trained models.

II. Related Work

Imitation learning has been widely adopted in robotics, enabling robots to learn from human demonstrations for complex tasks. Recent work has explored imitation learning to create robot policies from combined text and image input [2], [3], [4], [5], [6]. However, practical considerations such as occlusion and depth ambiguity limit the performance of such

[1]The authors are with the Institute for Robotics and Intelligent Machines at the Georgia Institute of Technology (GT). This work was supported in part by NSF Award # 2024444 and AI-CARING Award # 2112633. Charles C. Kemp is an associate professor at GT. He also owns equity in and works part-time for Hello Robot Inc., which sells the Stretch RE1. He receives royalties from GT for sales of the Stretch RE1.

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 [3], [4], [7], [8] 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 exhibit decreased out-of-distribution performance. Other works prioritize versatility but demand an extensive amount of data [2], [5]. Our approach lies between these two regimes, generalizing across environments and perspectives with a modest amount of data collection effort.

Several existing works have used kinematic objectives for robotic planning [3], [4], [9], [10], [11]. 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 shown evidence of contact and force prediction from visual input, and their effectiveness in facilitating control during object manipulation for both humans and robots [12], [13]. Prior works have also performed robotic planning with the use of contact points [14], [15], but are often restricted to grasp generation. Many real-world tasks involve complex interactions with the environment that can significantly benefit from tactile information, particularly for tasks requiring dexterous manipulation, such as grasping small objects, opening doors and drawers, pressing buttons, and twisting knobs [16], [17]. Contact and force are modalities that are largely invariant to different environments and objects for the same task, offering greater generalizability than other modalities such as vision or joint states.

The idea of combined visual-force servoing has been explored with both classical and data-driven methods in limited settings such as peg-in-hole and contour following [18], [19]. The conceptualization of course-to-fine visual servoing has also been proposed in prior work using data-driven methods, where many initial poses lead to a singular "bottleneck pose" [7], [8], [20]. However, these 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 keyframes, which are similar to bottleneck poses, to complete tasks. In contrast to methods with similar data collection schemes, ForceSight is capable of composing behaviors to complete multi-step tasks in a variety of environments and generalizes to objects semantically similar to those in the training set.

III. ForceSight: A Force-Based Robotic Planner

We present ForceSight, a system that represents robotic tasks as sequences of visual-force goals. A visual-force goal includes target fingertip locations in the camera frame, a target grip force, and a target force applied by the gripper.

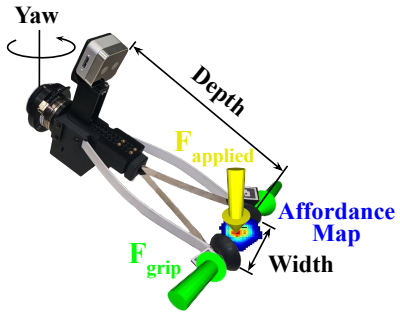

Fig. 2. Our representation of visual-force goals includes future fingertip locations (green arrow tips), future grip force (green arrow magnitudes), and future applied force (yellow arrow). The affordance map represents a probability distribution describing the goal gripper location in pixel space. These goals are represented in the camera frame.

A. Representing Visual-Force Goals

We define a visual-force goal G as the combination of a kinematic goal G_K and a force goal G_F. (Figure 2) The kinematic goal, which defines the desired 4-DOF pose (x, y, z, yaw) of the gripper, is parameterized as $G_K = \{C_{xy}, C_z, \psi, W\}$.

$C_{xy} \in \mathbb{R}^2$ is the 2D pixel coordinate of the 3D goal position projected onto the input image, and $C_z \in \mathbb{R}$ is the depth estimate of the 3D goal position with respect to the camera frame. Together, C_{xy} and C_z define a 3D point corresponding to the desired gripper location. In order to obtain a prediction for C_{xy}, we introduce an affordance map $\mathcal{A} \in \mathbb{R}^{H \times W}$, which is a probability distribution in pixel space describing the likelihood that the gripper should move to each pixel location. C_{xy} is obtained by taking $\arg\max_{x,y} \mathcal{A}$, the most likely pixel coordinate associated with the goal position of the gripper. Similarly, to obtain C_z, we also introduce a depth map $D \in \mathbb{R}^{H \times W}$. The depth estimate C_z is obtained by taking the pixel value from the 2D depth map at C_{xy}. W (gripper width) is the Euclidean distance between the predicted fingertip locations, and ψ is the predicted yaw of the gripper with respect to the camera frame, in radians. From these values, we can derive the 4-DOF kinematic goal of the gripper, i.e. the desired gripper position, yaw, and aperture.

In addition to a kinematic goal, we define a force goal $G_F = \{F_A, F_G\}$. The applied force $F_A \in \mathbb{R}^3$ is a vector that represents the desired applied force associated with the next step of a task, measured by a force/torque sensor mounted to the wrist of the robot, then transformed into the camera frame. Grip force, F_G represents a scalar value which is determined through a small force estimation neural network (see Section III-B for details).

Given an RGBD observation and a text prompt, ForceSight predicts visual-force goals G_K and G_F one keyframe into the future, with respect to the current camera frame. The output representations for ForceSight and eye-in-hand camera setup make the future predictions more amenable to visual servoing [21], which uses closed-loop control for improved robustness. Moreover, the same keyframe predictions have the potential to be useful to other embodied systems. For

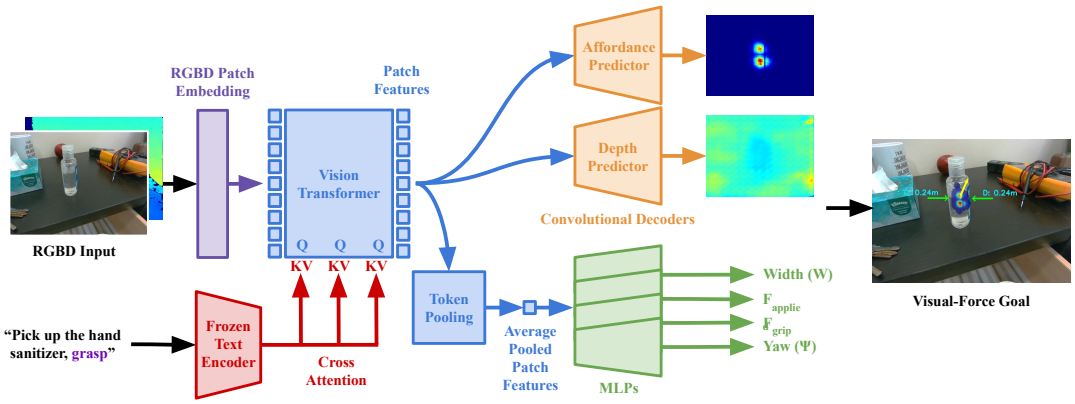

Fig. 3. ForceSight is a text-conditioned RGBD vision transformer. An RGBD image is first divided into patches and passed into an RGBD-adapted patch encoder that transforms image patches into visual tokens. These visual tokens are fed into a vision transformer. After every transformer block inside the 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 gripper width, applied force, grip force, and yaw.

example, another two-fingered robot can utilize the kinematic and force goals with little to no adaptation, as the predicted goals are not dependent on the camera mounting location (Figure 6). Consequently, a low-level policy can be adapted to visually servo a robot to reach its predicted goals.

B. Data Collection

We collect a dataset D, with $D_i = \{I, T, C_{LR}, F_G, F_R\}$, where $I \in \mathbb{R}^{H \times W \times 4}$ is an RGBD image captured by a gripper-mounted camera, T is a text prompt associated with a task and includes an action primitive, $C_{LR} \in \mathbb{R}^{2\times 3}$ is a set of two 3D fingertip locations in the camera frame associated with the next keyframe, F_G is the grip force, and F_A is the applied force. These combined elements constitute the data points in the dataset, providing a rich representation of a robot's interaction with its environment. From these datapoints, the kinematic and force goals (G_K and G_F) that serve as ground truth for our network can be derived.

This dataset contains over 26,000 high-quality datapoints, the equivalent of roughly 10,000 task demonstrations. Data was collected by teleoperating a Stretch RE1 mobile manipulator [22] for approximately 30 hours. We accomplish this rate of approximately 11 seconds per demonstration by associating multiple input images with the same visual-force goal, essentially collecting multiple views of a given goal from various perspectives. We also sampled from states that are unlikely to occur during a successful execution to promote recovery from errors. This allows our algorithm to be resilient to potentially imprecise predictions, thereby strengthening the robustness of the system. This is related to the concepts of funneling [23] and pre-image backchaining [24] that have inspired recent work in robust feedback motion planning [25].

To further enhance the efficiency of data collection, objects relevant to other tasks were deliberately included in many input images. This allows us to map the same input image to all goals present in the image, consequently generating substantially more data points per image in the dataset.

In order to collect visual data, we mount an Intel® RealSense™ D405 [26] to the robot's gripper to capture the RGBD image I. The force applied to the gripper is measured by a wrist-mounted force/torque sensor [27]. To obtain the grip force, which is not natively available on the Stretch RE1 gripper, we train an MLP to estimate the grip force F_G, given the gripper motor position, motor current, and fingertip positions. To provide ground truth for the grip force model, we grasp a force/torque sensor [27] at various grip strengths and grasp widths, and record the measured force magnitude.

To collect ground truth, the gripper's fingertips are first localized 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 each fingertip's surface, which we denote as the fingertip locations $C_{LR} \in R^{2\times 3}$. The fiducial markers could potentially be replaced with other methods of pose estimation. Subsequently, the robot's forward kinematics are utilized to map the 3D fingertip locations from the camera frame where the goal was achieved to the camera frame where the input image was captured.

C. Network Architecture

Our proposed architecture, leverages a large-scale Vision Transformer [28] (ViT-large, 304M parameters) and a frozen T5 text encoder [29] to output precise visual-force goals (Figure 3). Visual-force goals are composed of fingertip locations, grip force, and applied force, enabling robotic manipulation based on tactile objectives.

We initialize our vision transformer with weights from pre-training on ImageNet 21k [30]. 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 fusion technique described in [31]. 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 a text embedding vector that offers context for a given task. To

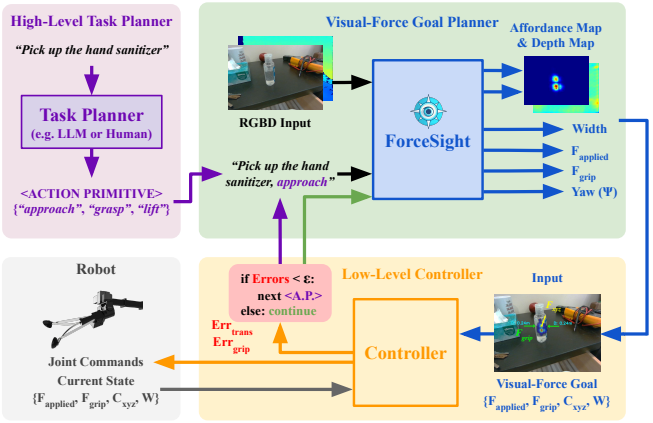

Fig. 4. Overall System Architecture.

effectively utilize this text information, we incorporate a cross-attention mechanism across all layers of the vision transformer, allowing the network to represent relationships between text prompts and visual features at multiple levels of abstraction. Inspired by developments in text-conditioned image generation [32], this operation applies cross-attention using projections of visual features as query vectors and projections of text features as key and value vectors.

The network output includes a 224×224 affordance map $\mathcal{A}$ and a depth map D. The ViT image encoder generates patch features associated with the input image's particular regions, which are transformed into an affordance map 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. The depth map prediction is supervised via an L_1 loss, masked by the ground truth affordance map, to focus only on the predicted gripper position. Hence, the depth map is conditioned on the affordance map, producing a depth estimate C_z in pixel space. Here, we define $C_z = D[C_{xy}]$, where C_{xy} represents the coordinate with the maximum likelihood in the affordance map, given by $C_{xy} = \arg\max_{x,y}(\mathcal{A})$. This representation allows our network to propose multiple hypotheses, exhibiting robustness in 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 task execution parameters such as gripper width, yaw, applied force, and grip force.

D. System Architecture

Our low-level controller receives kinematic and force goals located in the camera frame from ForceSight and executes incremental joint commands 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.

The objective of the low-level controller is to minimize the kinematic and force error 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 command $M_{\text{end-effector}} \in \mathbb{R}^3$ and $M_{\text{gripper}} \in \mathbb{R}$ in Cartesian space, and is then executed by the controller in a step-wise manner.

$$\begin{aligned} M_{\text{end-effector}} &= K_{ee}(T_k\mathcal{E}_{\text{translation}} + \lambda_{\text{applied}}T_f\mathcal{E}_{\text{applied-force}}) \\ M_{\text{gripper}} &= K_g(\mathcal{E}_{\text{width}} + \lambda_{\text{grip}}\mathcal{E}_{\text{grip-force}}) \end{aligned} \tag{1}$$

Where K_{ee} and K_g are proportional gain matrices, T_k and T_f are matrices that transform kinematic and force errors to robot motions, $\mathcal{E}$ denotes errors relative to the goals, and λ denotes weights on force errors relative to kinematic errors.

E. Action Primitives

The use of Large Language Models (LLMs) has demonstrated its efficacy in addressing sequential long-horizon robotic tasks [33], [34], [35], [36], [37]. 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, ungrasp, lift, and pull*, which are appended to the model prompt input. The low-level controller switches to the next action primitive once the errors between current states and target goals are adequately minimized, as outlined in the overall system architecture in Figure 4. This can result in the system appearing to make multiple attempts prior to achieving success, serving as a form of error recovery. We implement the switching between action primitives using predetermined sequences incremented by a simple state machine, but one could plausibly automate this process with the use of an LLM, as is demonstrated on our website.

IV. Training Details

For training, we processed 224x224 RGBD images. Ground truths for the affordance map take the form of multi-hot encodings, using circles with a radius of 10 pixels instead of single pixels to indicate the tool center point's coordinates. This denser representation proved efficient in generating rich heatmaps without compromising performance.

To improve model robustness and encourage generalization, the RGB channels of our input images underwent brightness, saturation, contrast, and hue augmentations. Additionally, during preprocessing, we applied a data filtering step to exclude examples with ground truths lying outside the camera's field of view, ensuring minimal prediction inaccuracies due to field of view constraints.

ForceSight was trained over 20 epochs, equating to 500,000 iterations, using batches of eight using the Adam optimizer [38] with a learning rate of 5e-5.

The loss function for ForceSight is given by:

$$L = \sum_{i\in\mathcal{L}} \lambda_i L_i \tag{2}$$

Where: $\mathcal{L} = A, D, F_A, F_G, W, \psi$ represents the set of all loss components and L_A, L_D, L_{F_A}, L_{F_G}, L_W, and L_ψ

Task	Action Primitives
Pick up the apple	Approach, Grasp, Lift
Pick up the medicine bottle	Approach, Grasp, Lift
Pick up the keys	Approach, Grasp, Lift
Pick up the paperclip	Approach, Grasp, Lift
Pick up the hand sanitizer	Approach, Grasp, Lift
Pick up the cup	Approach, Grasp, Lift
Place object in the trash	Approach, Ungrasp
Place object in the hand	Approach, Ungrasp
Turn off the light switch	Approach, Push
Open the drawer	Approach, Grasp, Pull

TABLE I

TASKS AND ACTION PRIMITIVES.

Fig. 5. The real-world experiments consist of 6 picking tasks, 2 placing tasks, a drawer opening task, and a light switch flipping task, encompassing various objects and environments not present in the training set.

correspond to the affordance map weighted cross-entropy loss, masked depth map MAE, applied force MSE, grip force MSE, gripper width MSE, and yaw MSE respectively. Depth loss is masked by the affordance map ground truth, and thus is only applied at locations where the affordance map ground truth is nonzero.

V. EXPERIMENTAL RESULTS

We use a Stretch RE1 [22] 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 (Table I). The robot exclusively interacts with unseen object instances, i.e. objects that are semantically similar to those seen during training but are visually distinct from objects in the training dataset.

A. Experimental Setup

To evaluate ForceSight, we select a set of 10 household tasks and perform 10 trials on each, totaling 100 trials. We also perform real-world experiments on 5 ablations of our model, each for 20 trials. For each trial, we randomize the robot's pose within 3 unseen environments such that the target pose is in the camera's field of view and is within a distance of 1 meter. We also randomize the pose of the held-out target objects and distractor objects. We define task success as achieving all subgoals associated with a task. Achieving a subgoal means that the kinematic and force errors have met the criteria specified in the low-level controller, which runs at a frequency of 8Hz.

B. Metrics

We employ the following set of metrics for evaluation:

Average Fingertip Distance: The mean L_2 distance between predicted and actual fingertip locations in meters (m). Smaller values signify better accuracy.

RMSE: Calculates the root-mean-square error between predicted and actual values of applied and grip forces (F_A and F_G), reflecting force estimation accuracy.

Task Success Rate: The percentage (%) indicating the system's success rate in completing tasks across trials.

C. Baselines

We train and evaluate Perceiver-Actor (PerAct) [4] on our dataset as a baseline. We adapt PerAct to predict goals in the camera frame rather than a global frame, allowing their algorithm to be applied to mobile manipulation. To further enhance the model's robustness, we introduced color jitter as a data augmentation strategy, supplementing the augmentations already employed by PerAct.

We use an input image resolution of 640×480, as opposed to ForceSight's 224×224. We also integrate action primitive prompts into the PerAct model for a balanced comparison against ForceSight. All remaining parameters, such as the PerceiverIO latent dimension and a voxel grid size of 100^3 (1m spatial volume), in addition to their data augmentation techniques such as translation and rotation perturbations, were retained from the original PerAct implementation.

As seen in Table II, we also train, we additionally evaluate several ablations of ForceSight, including removing the forces from the low-level controller (*w/o forces*), training without data augmentation (*w/o augmentation*), using only RGB inputs without depth (*w/o depth*), not pre-training on ImageNet 21k (*w/o pre-training*), and predicting visual-force goals from RGBD information only, without a text prompt (*w/o text cond.*).

D. Test Set Results

We collect a test set in unseen environments containing several unseen objects (Figure 5). 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 fingertip distance. ForceSight outperforms the PerAct baseline and all ablations on our fingertip distance metric, demonstrating that our representation is capable of generating accurate kinematic goals in unseen situations.

E. Real-World Results

When evaluated on several real-world tasks in novel environments with unseen object instances, ForceSight achieves a task success rate of 81%. According to the results presented in Table II, we demonstrate the significance of force as a modality for successfully executing tactile tasks, such as picking up a paperclip. 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. For example, using only kinematic objectives tends to result in items falling due to the lack of grip force consideration. Similarly, neglecting applied forces often results in excessive force or failure to make proper contact with surfaces like tables and drawers, thus worsening grasp success rates.

	Task Success	Avg. Fingertip Dist. (m)	RMSE Applied Force (N)	RMSE Grip Force (N)
ForceSight (Ours)	**81/100 (81%)**	**0.036**	**0.404**	1.524
Perceiver-Actor [4]	-	0.078	-	-
w/o forces	10/20 (50%)	**0.036**	-	-
w/o augmentation	5/20 (25%)	0.049	1.181	1.759
w/o depth	4/20 (20%)	0.063	1.493	1.32
w/o pre-training	4/20 (20%)	0.078	0.583	1.576
w/o text cond.	0/20 (0%)	0.075	0.907	**1.260**

TABLE II

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.

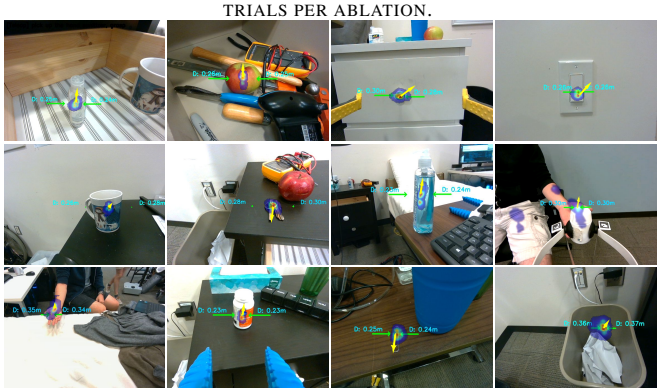

Fig. 6. ForceSight is capable of providing accurate estimations of fingertip location, applied forces, and grasping forces associated with a task in a variety of environments. Our network additionally generalizes to unseen object instances, does not depend on a specific camera setup, and works in situations with partial occlusion.

In a more controlled experiment comparing ForceSight performance with and without force goals, the success rate plunged from 90% (18/20) to 45% (9/20) when force goals were overlooked. Common failure scenarios involved excessive force, gripping issues, and positioning errors.

Additionally, omitting depth input during training made the robot prone to early grasping due to compromised depth perception. Moreover, data augmentation was found crucial for better performance in varied environments and with unfamiliar objects.

VI. LIMITATIONS

While ForceSight exhibits encouraging performance across diverse and challenging tasks, we recognize certain limitations and areas for future improvement.

One of the recurrent failure modes observed in our model pertains to inaccuracies in depth predictions, particularly those further away from the camera (1m). Despite that, the adaptive capability of visual servoing accounts for the discrepancies, as errors diminish at closer distances.

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.

Our keyframe representations do not provide complete information about the gripper's pose, assuming pitch and roll to be constant. However, this can be addressed by adding additional MLP heads to the model's output.

While our study showcases success in real-world tasks like pick-and-place, it remains limited in scope. However, the scalability of transformer models and our efficient data collection suggest the potential for broader complex tasks. Additionally, our experiments, conducted solely with the RE1 robot, should be expanded to different robotic platforms in future studies to truly gauge our model's versatility and robustness.

VII. DISCUSSION

After training, we observed multiple emergent behaviors exhibited by ForceSight (Figure 7 and 8). Most notable was ForceSight's ability to generalize to unseen object instances despite only having observed one type of each object during training, likely due to the high-level visual associations retained from pre-training. ForceSight also exhibited adaptability in its treatment of action primitives in relation to various objects. ForceSight managed to apply 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 subtasks being absent in the training data.

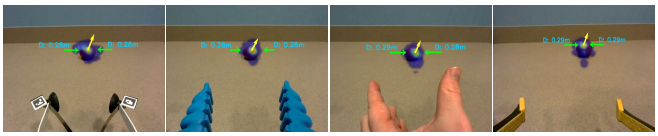

Fig. 7. Predictions from ForceSight are agnostic to the agent and camera perspective, as shown in this example for the apple grasping task.

Fig. 8. We observe that ForceSight is able to make predictions for action primitives that are more than one keyframe into the future, despite having been trained to predict goals associated with the next keyframe.

VIII. 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] V. Babin and C. Gosselin, "Picking, grasping, or scooping small objects lying on flat surfaces: A design approach," *The International journal of robotics research*, vol. 37, no. 12, pp. 1484–1499, 2018.

[2] 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.

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

[4] ——, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[5] 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*. PMLR, 2022, pp. 991–1002.

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[7] 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)*. IEEE, 2022, pp. 8614–8621.

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

[9] 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*, vol. 32, 2019.

[10] 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*. Springer, 2022, pp. 132–157.

[11] 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*, 2022, pp. 13 739–13 748.

[12] J. R. Flanagan, M. C. Bowman, and R. S. Johansson, "Control strategies in object manipulation tasks," *Current opinion in neurobiology*, vol. 16, no. 6, pp. 650–659, 2006.

[13] A. Jain and C. C. Kemp, "Improving robot manipulation with data-driven object-centric models of everyday forces," *Autonomous Robots*, vol. 35, pp. 143–159, 2013.

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

[15] 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)*. IEEE, 2021, pp. 13 438–13 444.

[16] 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)*. IEEE, 2022, pp. 3628–3635.

[17] 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.

[18] J. Baeten, H. Bruyninckx, and J. De Schutter, "Integrated vision/force robotic servoing in the task frame formalism," *The International Journal of Robotics Research*, vol. 22, no. 10-11, pp. 941–954, 2003.

[19] 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)*. IEEE, 2021, pp. 1–6.

[20] 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.

[21] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[22] 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)*. IEEE, 2022, pp. 3150–3157.

[23] M. Mason, "The mechanics of manipulation," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1985, pp. 544–548.

[24] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *The International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.

[25] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

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

[27] ATI Industrial Automation. (2022) F/T Sensor: mini45. [Online]. Available: https://www.ati-ia.com/products/ft/ft_models.aspx?id=mini45

[28] 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.

[29] 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*, vol. 21, no. 1, pp. 5485–5551, 2020.

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

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

[32] 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*, 2022, pp. 10 684–10 695.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

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