Learning Generalizable Vision-Tactile Robotic Grasping Strategy for Deformable Objects via Transformer

Yunhai Han, *Student Member, IEEE*, Kelin Yu, *Student Member, IEEE*, Rahul Batra, *Student Member, IEEE*, Nathan Boyd, *Student Member, IEEE*, Chaitanya Mehta, *Student Member, IEEE*, Tuo Zhao, *Member, IEEE*, Yu She, *Member, IEEE*, Seth Hutchinson, *Fellow, IEEE*, and Ye Zhao, *Senior Member, IEEE*

***Abstract*—Reliable robotic grasping, especially with deformable objects such as fruits, remains a challenging task due to underactuated contact interactions with a gripper, unknown object dynamics and geometries. In this study, we propose a transformer-based robotic grasping framework for rigid grippers that leverage tactile and visual information for safe object grasping. Specifically, the transformer models learn physical feature embeddings with sensor feedback through performing two predefined explorative actions (pinching and sliding) and predict a grasping outcome through a multilayer perceptron with a given grasping strength. Using these predictions, the gripper predicts a safe grasping strength via inference. Compared with convolutional recurrent networks, the transformer models can capture the long-term dependencies across the image sequences and process spatial–temporal features simultaneously. We first benchmark the transformer models on a public dataset for slip detection. Following that, we show that the transformer models outperform a CNN + LSTM model in terms of grasping accuracy and computational efficiency. We also collect a new fruit grasping dataset and conduct online grasping experiments using the proposed framework for both seen and unseen fruits. In addition, we extend our model to objects with different shapes and demonstrate the effectiveness of our pretrained model trained on our large-scale fruit dataset.**

***Index Terms*—Deep learning, perception for grasping and manipulation, visual and tactile sensing.**

Manuscript received 24 July 2023; revised 11 April 2024; accepted 3 May 2024. Recommended by Technical Editor D. Naso and Senior Editor Q. Zou. This work was supported in part by the Office of Naval Research (ONR) under Grant #0023 N000142312223, in part by National Science Foundation (NSF) under Grant #0023 FRR-2328254, and in part by the USDA under Grant #2023-67021-41397. *(Yunhai Han and Kelin Yu contributed equally to this work.) (Corresponding author: Ye Zhao.)*

Yunhai Han, Kelin Yu, Rahul Batra, Nathan Boyd, Chaitanya Mehta, Seth Hutchinson, and Ye Zhao are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: yhan389@gatech.edu; kyu85@gatech.edu; gtg693m@gatech.edu; nboyd31@gatech.edu; cmehta43@gatech.edu; seth@gatech.edu; yezhao@gatech.edu).

Tuo Zhao is with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: tourzhao@gatech.edu).

Yu She is with the School of Industrial Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: shey@purdue.edu).

Our codes and dataset are public on GitHub at https://github.com/GTLIDAR/DeformableObjectsGrasping.git.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2024.3400789.

Digital Object Identifier 10.1109/TMECH.2024.3400789

I. Introduction

ROBOT manipulation has been widely used in industries for decades, but mostly for repetitive tasks in structured environment where there is little uncertainty or contact deformation in manipulated objects. For the tasks where object contact parameters are prone to vary, such as fruit grasping, they are still challenging for robotic systems [1]. Loose grips with small grasping forces can cause objects to slip, while large grasping forces can cause damage. In addition, object contact geometry and frictional properties may also affect the optimal grasping forces for safe grasping. To learn general-purpose grasping skills, robots need to leverage with dense notions of contact information from in-hand interactions.

To model the dynamic interactions between the object and its environment, vision-based sensing frameworks have been studied based on a sequence of visual observations obtained by external cameras [2], [3]. However, these methods are not sensitive to the dense local deformation near contact regions, which could lead to errors between the perceived and actual states of a grasp. To address this issue, tactile sensing has gained increasing popularity recently [4]. Among various tactile sensors, the ones with internal cameras, such as GelSight sensor [5], have the capability of capturing high-resolution image data regarding local contact geometry. Other tactile designs [6], [7] have also demonstrated a variety of manipulation tasks with similar methods. Compared to force sensors, tactile sensors can capture an object's deformation during contact. Moreover, tactile data can be readily integrated by modern learning methods for classification and task-oriented control policy learning [8]. The authors in [9] and [10] demonstrated that grasping performance can be significantly improved by incorporating visual and tactile sensing.

In this article, we employ two state-of-the-art transformer models—TimeSformer [11] and ViViT [12]—to determine safe grasping forces from the visual and tactile image sequences collected during predesigned explorative actions (e.g., pinching

and sliding). The idea of designing task-oriented explorative actions is inspired by Wang et al. [8] and the motivations of introducing the transformer models are as follows.

1) Compared with recurrent networks, such as LSTM, they do not suffer from the forgetting issue.
2) Compared with convolutional networks used for extracting local features, they have larger receptive fields that are helpful to understand the global context.
3) Compared with CNN + LSTM models for processing image sequences, they can extract the spatial–temporal features simultaneously. While for CNN + LSTM models, the per-frame spatial features are always encoded (CNN) prior to the temporal decoding (LSTM). Thus, the transformer models are more adaptable to complex tasks.

In our framework, the transformer models learn low-dimensional embeddings in a supervised fashion for each sensor modality and then output a fused physical feature embedding. First, we take this embedding as input and combine it with a given grasping force threshold to predict the final grasping outcomes through a multilayer perceptron (MLP). The grasping outcomes are categorized into three labels: safe grasping, slippery, and potential damage. A force threshold for safe grasping is then searched for using the learned predictor during online deployments. Second, the fused physical feature embedding is used to classify grasped fruit types through a different MLP layer in order to place them into separate bins automatically.

To begin with, we benchmark the transformer models against a CNN + LSTM model on a public dataset for slip detection [9]. Both transformer methods (TimeSformer and ViViT) outperformed the CNN + LSTM model by 3.1% and 2.0% in detecting slip, and are much more computationally efficient, making them more suitable for online tasks. Then, in order to validate the grasping framework, we perform grasping experiments on various deformable fruits for data collection. We train the models using both camera and GelSight inputs and test their performance via grasping outcome classifications on unseen fruits and online grasping success rate for both seen and unseen fruits. Furthermore, we employ our pretrained model, the one trained on a dataset of multiple fruits, for training with an unseen banana dataset. Notably, employing the pretrained model accelerates the training process by four times and leads to a 10% improvement in training accuracy in the new banana dataset.

The contributions of our work are summarized as follows.

1) We propose a transformer-based grasping framework for fruit grasping and demonstrate the superior efficacy and efficiency of the transformer models against a CNN + LSTM baseline model.
2) We design a learning-based control framework that incorporates safe grasping force estimation using tactile and visual information obtained via two explorative actions: pinching and sliding, which do not require any prior knowledge of physical contact or geometrical models. Besides, the control parameter is directly formulated as the depth value read from the tactile feedback without any aid of external force–torque sensors.
3) We experimentally evaluate the proposed grasping framework on a diverse set of fruits and achieve an end-to-end demonstration of fruit grasping. Besides, by performing the attention analysis, we show that the trained transformer models take advantage of the attention mechanism to: i) incorporate more contact area information for the grasping task, such as local contact region in tactile images and fruit surface near the gripper's fingertips in visual images; and to ii) capture long-term dependencies between initial and final grasping status. Furthermore, we conduct detailed sensitivity analysis for tactile images and visual images with different data qualities in Section V-A. We evaluate the robustness of our transformer models according to the difference in outputs.
4) By employing our model trained on a large fruit dataset as the pretrained model for our collected banana dataset, we observe improved training efficiency and higher success rates in both the training process and online grasping experiments.

II. Related Work

A. Robotic Grasping

Robotic grasping has been a widely explored topic involving different gripper designs, different model-based control methods, and learning-based methods with various sensor modalities [13], [14]. As for gripper design, previous works have explored novel design principles for efficient manipulation of deformable objects. The three fingered Barrett Hand has been widely used in grasping framework for objects with different shapes and hardness [15], [16]. The sensorized hand proposed in Friedl and Roa [17] work uses a multimodel observer framework and a variety of sensors, including cameras, tendon force sensors, and proximity sensors, to achieve successful grasping. However, these approaches often involves sophisticated hardware design and potentially limit the flexibility in deployment across different robotic systems. Instead, our work employs a simple parallel jaw gripper for grasping and focuses on leveraging high-fidelity tactile sensing to learn reliable grasping strategies for a diverse set of deformable objects. As for grasping control strategies, model-based frameworks have also been widely used in deformable objects manipulation [16], [18]. The study in Zaidi et al. [16] work develops a grasp planning pipeline which emphasizes the physical contact constraints for precise control of forces and object deformations. The work of Chang and Padır [18] contributes by modeling and automating the manipulation of linear flexible objects to enhance grasping accuracy. While model-based methods can achieve high success rates in grasping, their ability to generalize across objects of varying shapes and softness is limited. The generalizability of deformable object grasping is one of the main objectives in our study. Pozzi et al. [19] proposed a vision-based framework for grasp learning of deformable objects using an anthropomorphic, underactuated, compliant hand. It renders a promising future direction to employ advanced hand mechanisms for grasping. Recently, motivated by human's intense dependence of tactile feedback for the grasping process, tactile sensors have thus begun to play an important role in robotic grasping [20]. Calandra et al. [21], used deep-learning methods to obtain a grasping

policy for rigid grippers. However, the grasping success rate on deformable objects was not ideal since they only adjusted the grasping position but fixed the grasping force. Kim et al. [22] estimated the optimal grasping force empirically but assumed the object weights were known. In [23] and [24], the gripper's opening was controlled to stabilize the grasped objects under external disturbances by detecting the slip occurrences. Similarly, Wettels et al. [25] made use of tactile sensing to stabilize the grasped object by controlling the grasping force. However, all of these studies assumed that the objects were already steadily grasped in hand. In this work, we aim at estimating safe grasping force for deformable objects through a learning framework.

B. Vision-Tactile Sensor Fusion

We can improve the manipulation performance by fusing the information obtained from visual and tactile sensors. Calandra et al. [10] proposed a multimodal sensing framework for grasping outcome prediction. Their subsequent work of Calandra et al. [21] investigated a learned regrasp policy based on visuotactile data after executing an initial grasp. Their results indicated that incorporating tactile readings substantially improves grasping performance. However, the manipulated objects used in their experiments are primarily rigid objects, which do not require accurate force control. In other works [9], [26], [27], they used CNN + LSTM models to classify the slip occurrence, to recognize the object instance, and to perceive the physical properties of objects. Nonetheless, these methods can only be used for classification tasks and are not applicable to learning control policy for safe manipulation.

C. Transformers for Robotics

Transformer models were originally proposed for natural language processing [28] and computer vision [29], [11], [12], and [30]. Recently, transformers have drawn increasing attention in robotics. Shridhar et al. [31] proposed a transformer framework for tabletop tasks, which encodes language goals and RGB-D voxel observations and output discretized six-DoF actions. Monastirsky et al. [32] explored the use of transformers to predict robot action commands for accurate object throwing. Yang et al. [33] addressed quadrupedal locomotion tasks using reinforcement learning with a transformer-based model. All of these works showed significant improvements over baseline methods on task performance and training efficiency. However, to the best of our knowledge, no existing study has ever explored the use of transformers for robotic grasping using tactile and visual images.

III. Methods

In this section, we describe the details of the grasping framework and each transformer model. To give robots the ability to estimate the safe grasping force, we first let the robot obtain physical information of the target objects (fruits in this work) by performing two explorative actions, *pinching* and *sliding*, on the objects. To avoid potential damage, these actions have minimum interaction with the objects. To monitor the interactions and record the data, the robot is equipped with two different sensors. Next, a force threshold for safe grasping will be extracted via inference from the obtained physical information and adopted for execution. In the following, we first describe the sensors in Section III-A, and then we discuss the transformer models in Section III-B, and finally we propose the grasping framework in Sections III-C and III-D. The pseudocode code is shown in Appendix B in the supplementary material. The line of the appendix is: https://drive.google.com/file/d/1JhOoCgqlDYf12YNWXCDOrsryCRRk1FH8/view?usp=sharing

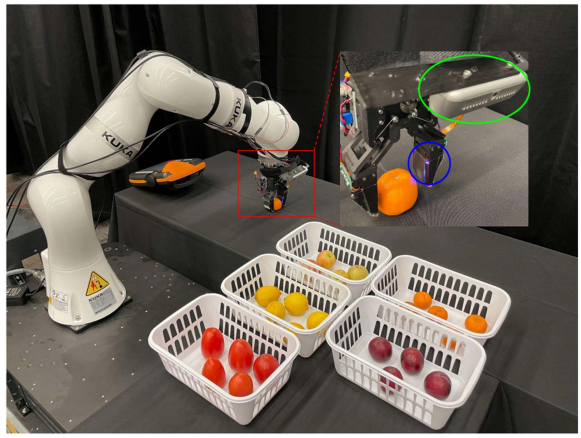


Fig. 1. Demonstration of our robotic grasping experimental setup. The robot gripper safely grasps the fruits on the table and sorts them into the target bins via the learned framework. Robot setup: A KUKA LBR iiwa robot is equipped with a gripper (red box), of which both fingers are equipped with a GelSight sensor (blue circle). A Realsense D435 is mounted above the gripper (green ellipsoid).

A. Sensing Modalities

1) Tactile: The GelSight [5] sensor provides the robot with dense visual information (high-resolution image) about the contact region between the objects and the robot's fingertips. For this purpose, the contact surface of the sensor is covered with a soft elastomer such that the sensor can measure the object's compliance by observing the elastomer's vertical and lateral deformation. In out experiments, the gripper has two GelSight sensors installed on the fingertips, but we only use one to demonstrate a minimum system setup.

2) Vision: A RealSense D435 camera is used in this work, and we only consider the RGB datastream. The camera is wrist mounted at an angle of 15° such that the image is centered on grasped objects (see Fig. 1 for the setup).

B. Transformer Model

We apply the transformer models for two robotic manipulation tasks: slip detection and safe grasping force estimation. For slip detection, we replace the CNN + LSTM model used in Li et al. [9] work with the transformer models but keep the last fully connected (FC) layer with two outputs (i.e., a stable grasp or slip) as the final classification results. For safe grasping force estimation, the outputs from transformer models are used as

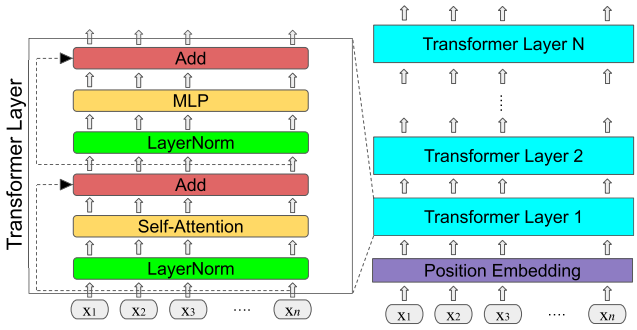


Fig. 2. Illustration of the transformer model structure. The left figure shows one transformer layer and the right figure shows the encoding structure. $\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \ldots, \mathrm{x}_n$ are the input vectors which are first linearly embedded and then added with the position embeddings before transformer layer 1.

inputs to the subsequent models in the grasping framework that will be thoroughly described in Sections III-C and III-D.

Two lightweight transformer models are explored for robotic tasks in this work: TimeSformer [11] and ViViT [12]. Each model uses similar self-attention mechanism [28], [34], which brings main advantages over CNN + LSTM models, while the difference between these models lies in the factorizing strategy for spatial–temporal attention. The choice of TimeSformer and ViViT is pivotal due to their specialized architectures, which enable the attention strategy across spatial and temporal dimensions in video data. This strategy enhances the performance of complex classification tasks in robotic manipulation by leveraging spatial and temporal dynamic information.

A transformer layer contains a self-attention layer and an MLP layer. To stack the transformer layers for a deeper encoding structure, the MLP layer does not change the vector size. Also, before and after both layers, there is a LayerNorm and a residual connection, respectively. One transformer layer is shown in Fig. 2 (left-hand side subfigure), where the outputs of the current layer will be the inputs for the next (right-hand side subfigure). Before the first transformer layer, all the input vectors will be linearly embedded and then added with position embeddings, the elements of which represent the positions of each vector, to retain the useful sequence knowledge [28].

Self-attention mechanism: The self-attention mechanism [28], [34] allows all the inputs to interact with each other and identify the one that should be paid more attention to, which renders their main advantages over CNN + LSTM models. Specifically, this mechanism can be described as mapping a query (Q) and a key (K)-value (V) pair to the outputs.

For a single self-attention block (or a single head), the query, key, and value vectors can be computed by projecting the same input matrix $\mathrm{X} \in \mathbb{R}^{n \times d_x}$ (each row of X corresponds to an input vector with size d_x) to Q, K, V as follows:

$$\mathrm{Q} = \mathrm{X}\mathrm{W}^Q, \mathrm{K} = \mathrm{X}\mathrm{W}^K, \mathrm{V} = \mathrm{X}\mathrm{W}^V \tag{1}$$

where $\mathrm{W}^Q \in \mathbb{R}^{d_x \times d_k}$, $\mathrm{W}^K \in \mathbb{R}^{d_x \times d_k}$, and $\mathrm{W}^V \in \mathbb{R}^{d_x \times d_v}$ are learnable matrices with $d_k = d_v = d_x$.

The outputs of the self-attention mechanism are obtained through (2), which represents the weighted sums of the value vectors (V) with the weights assigned based on a compatibility function between the query vectors (Q) and the corresponding key vectors (K) at the same vector index. Note that, the dot-product between Q and K is scaled by $\sqrt{d_k}$, as suggested by Vaswani et al. [28]

$$\mathrm{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{Q}\mathrm{K}^T}{\sqrt{d_k}}\right)\mathrm{V} \tag{2}$$

$$\mathrm{SingleHead}(\mathrm{X}) = \mathrm{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V})\mathrm{W}^O. \tag{3}$$

As shown in (3), another learnable matrix $\mathrm{W}^O \in \mathbb{R}^{d_v \times d_x}$ projects the intermediate results to a new matrix with the same dimension as X.

In practice, to allow the model to attend to information from different combinations of input space representations, we employ a MultiHead strategy by projecting the Q, K, V matrices h times with different sets of weights W_i^Q, W_i^K, W_i^V for $i = 1, \ldots h$. This strategy leads to more effective representations and improved performance. As a result, it is always employed by transformer models.

In addition to the self-attention blocks, there is an FC MLP layer applied to each vector position separately and identically. It has two linear transformations and a GeLU activation function in between.

Factorization of spatial–temporal attention: For image-based tasks, to generate input vectors from raw image(s), in Dosovitskiy et al. [29] work, they split an image into fixed-size patches and embed each of them via linear transformation. Our framework handles image sequences instead of single images and must consider the temporal dimension within each self-attention layer. To accomplish this, we incorporate spatial–temporal factorization using TimeSformer [11] and ViViT [12].

TimeSformer: In this model, spatial–temporal dimensions are processed sequentially: within each self-attention layer, the attention is first applied on the temporal dimension of the inputs at the same spatial position, followed by the spatial dimension among all inputs from the same temporal position. There are also residual connections between each operation. This approach is visualized in Fig. 3. In our work, the input image sequence is denoted as $\mathrm{X}_I \in \mathbb{R}^{N \times H \times W}$, where N, H, W are the number of images, image height pixels, and image width pixels, respectively. We first extract the patches $\mathrm{X}_P \in \mathbb{R}^{P_n \times P_h \times P_w}$, where (P_h, P_w) is the resolution of each patch and $P_n = \frac{NHW}{P_h P_w}$. Next, these patches are flattened and then linearly embedded to vectors of size D with a positional embedding being added to each of them. We further add a CLS (classifier) token to the sequence of embedded vectors, which is designed to extract task-level representations [35] by attending to all the other vectors and forming an augmented sequence $X = \{\mathrm{CLS}, x_1, x_2, \ldots, x_N\}$, where $x_i \in \mathrm{X}_P$. $E(X)$, which is the transformer encoder function in ViT, then takes X as input and generates an encoded sequence of representations $\{h_{CLS}, h_1, h_2, \ldots, h_N\}$, where h_i is the representation of the ith patch. Finally, the output of the CLS token h_{CLS} is used for different tasks. In slip detection (see Section IV-A), it is passed through an MLP layer to classify whether or not a slip occurs.

ViViT: Our implementation of ViViT is similar to TimeSformer, except for the following differences: First, both dimensions are processed in parallel. Specifically, half of the

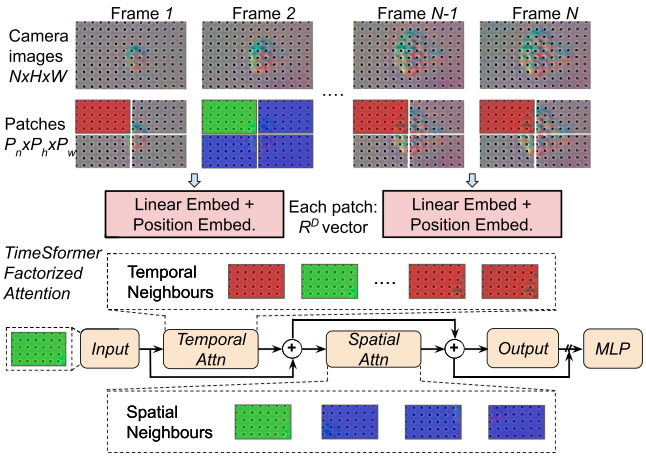


Fig. 3. Visualization of space–time attention approach for TimeSformer. The top three rows show an input GelSight image sequence, the generated 4 N image patches and the patch embeddings. We denote one image patch in green and its spatial–temporal neighbors in blue and red, respectively. Within each self-attention layer, for each image patch, the attentions across temporal neighbors and spatial neighbors are sequentially processed and the output will be the input of the MLP layer in transformer model after LayerNorm.

heads attend to the spatial dimension and the other half to the temporal dimension (factorized dot-product attention). We then combine each output by concatenation and add a linear transformation to halve the size. Second, there is no CLS tokens added to the embedded input vectors because of the ambiguities when dot-producting the temporal and spatial attention. Instead, we take the average of all patch outputs from the last transformer layer and pass it (size D) to the MLP layer to classify whether or not a slip occurs

C. Grasping Framework for Safe Force Estimation

The main goal of our grasping framework is to predict the grasping outcome given a grasping force threshold and to estimate the force threshold for safe grasping via inference.

1) Grasping Outcome Prediction: As shown in Fig. 4, this framework is composed of five main components: Control parameter (force threshold), transformer, sensor fusion model, action fusion model, and prediction model.

Force threshold: GelSight is a vision-based tactile sensor, which lacks the capability of estimating the grasping force (contact normal force) directly. To address this issue, Yuan et al. [6] showed that the contact normal force can be estimated from the depth value (unit: pixel) with accurate gel calibration. On the other side, She et al. [36] directly used the mean value of the marker displacement provided by GelSight images to approximate the resultant frictional force. Inspired by this work, we employ the maximum depth value as the approximation of grasping forces. If the maximum depth value reaches the selected threshold for three continuous frames during the execution, the gripper will begin to grasp the fruit with a constant force corresponding to the depth threshold value. For all subsequent operations, we maintain the same grasping force but solely control the robot arm motion for fruit transportation. Also, the force threshold will be sent into the prediction model. Note that the unit of force threshold is pixel, which will be omitted in Section IV for readability.

Transformer: For each explorative action, the image sequence from a sensor modality outputs a vector of size D via the transformer models, as thoroughly described in Section III-B. In Fig. 4, since we predefine two explorative actions and there are two sensor modalities, we have four vectors: $\mathrm{v}^{\mathrm{pinch}}_{\mathrm{visual}}$, $\mathrm{v}^{\mathrm{pinch}}_{\mathrm{tactile}}$, $\mathrm{v}^{\mathrm{slide}}_{\mathrm{visual}}$, $\mathrm{v}^{\mathrm{slide}}_{\mathrm{tactile}}$.

Action fusion model and sensor fusion model: We concatenate each two vectors obtained from the same exploration action and achieve: $\mathrm{v}^{\mathrm{pinch}} = [\mathrm{v}^{\mathrm{pinch}}_{\mathrm{visual}}, \mathrm{v}^{\mathrm{pinch}}_{\mathrm{tactile}}]$ and $\mathrm{v}^{\mathrm{slide}} = [\mathrm{v}^{\mathrm{slide}}_{\mathrm{visual}}, \mathrm{v}^{\mathrm{slide}}_{\mathrm{tactile}}]$. We then fuse them as a vector $\mathrm{v}^{\mathrm{fused}} = [\mathrm{v}^{\mathrm{pinch}}_{\mathrm{visual}}, \mathrm{v}^{\mathrm{pinch}}_{\mathrm{tactile}}, \mathrm{v}^{\mathrm{slide}}_{\mathrm{visual}}, \mathrm{v}^{\mathrm{slide}}_{\mathrm{tactile}}] \in \mathbb{R}^{4\times D}$. Then, we use a linear transformation operation to project it to a low-dimensional space with an output size of N. The linear transformation can be represented as

$$\mathrm{Y}^{\mathrm{fused}} = \mathrm{v}^{\mathrm{fused}} \cdot \mathbf{W}^{\top} + \mathbf{b} \tag{4}$$

where $\mathrm{Y}^{\mathrm{fused}} \in \mathbb{R}^{N\times 1}$ is the output vector, which is a fused physical feature embedding. $\mathbf{W} \in \mathbb{R}^{N\times 4D}$ is the weight matrix which is a learnable parameter trained through backpropagation, which optimizes a loss function that measures the discrepancy between the predicted and truth labels across a set of training examples, and then used to perform the linear mapping. $\mathbf{b} \in \mathbb{R}^{N\times 1}$ is another learnable bias which is used as an offset to the output.

2) Safe Force Threshold Estimation: We aim to identify the control parameter, i.e., the safe grasping force threshold. As shown in Fig. 4, the prediction model takes the low-dimensional physical embedding obtained from performing two explorative actions and a force threshold candidate as inputs and outputs the upcoming grasping outcome via learnable neural network layers. Next, we can uniformly sample the thresholds and feed each of them into the prediction model and select the one that predicts a safe grasping. When there are multiple viable choices, we select the average value.

D. Grasping Framework for Fruit Classification

Our grasping framework includes a goal of fruit type classification for pick-and-place operations. Specifically, we use a single-layer MLP network for fruit classification, which takes the fused physical feature embedding of sensor fusion model as input and outputs the grasped fruit type (*fruit type* block in Fig. 4). In detail, the MLP network learns a nonlinear mapping from the high-dimensional feature space of the sensor fusion model, which encodes physics knowledge, to the categorical labels of the fruit types. During training, the weights of the transformer models that generate the embedding are frozen. Only the MLP network is trained in this learning scheme.

IV. Experiments

In this section, we present our experiments using the transformer models. The robot setup is shown in Fig. 1.

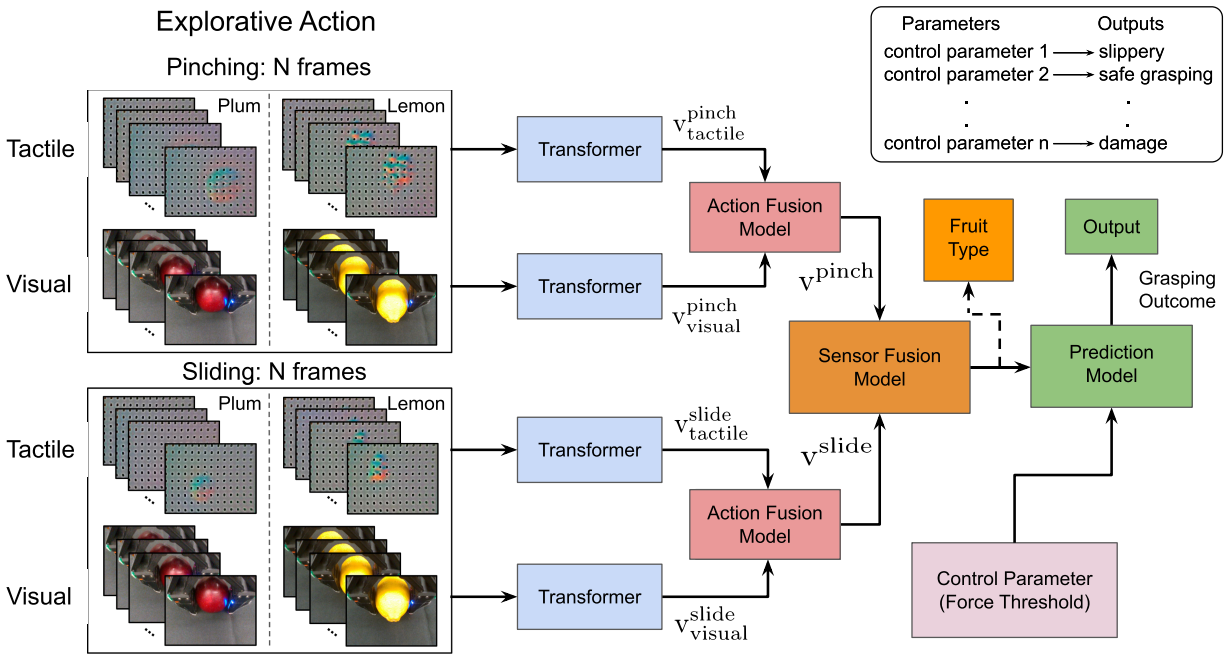


Fig. 4. *Overview of grasping framework.* The robot first performs two explorative actions. 1) Pinching the fruit. 2) Sliding along the fruit surface in the optical axis. Many fruit examples can be found on our GitHub page. Each image sequence is processed by an individual transform network into a vector of size D. The fusion models concatenate these vectors and project it into a low-dimensional fused physical feature embedding. This embedding is further processed by a fruit classification model to classify the grasped fruit type. Besides, the prediction model takes the same embedding and control parameter (force threshold) as inputs and predicts the final grasping outcome. Through inference, a set of control parameters is first generated and then the parameter with the safe grasping outcome is selected to perform online grasping. This procedure is shown in the top-right black box. If there are multiple viable choices, we select the **average** value.

A. Transformers for Slip Detection

To begin with, we benchmark the transformer models against a CNN + LSTM model on a public dataset for slip detection with different sensor modalities.

1) Experiment Setup: We conduct experiments for a slip detection task. The dataset released by Li et al. [9] is used and can be directly downloaded online.[1] During implementation, the entire dataset is split into training, validation, and test sets, where the test data uses unseen objects in the training data. Since the size of dataset is relatively small, we randomly split the dataset five times and train the model on each of them to mitigate the effect of overfitting. The final detection accuracy on the test set is averaged. For each model, we analyze the performance with three different data source inputs (vision-only, tactile-only, and vision and tactile). For the CNN + LSTM model, ResNet18 is chosen as the CNN architecture over other options, such as VGG or Inception [9], due to its advantage of fewer parameters. As a result, the ResNet18 architecture can be initialized randomly without the need of loading a pretrained model.

A sequence of *14 continuous frames* are used as input for each sensor data. During training, we use cross-entropy (two categories) as the loss function and apply an Adam optimizer [37]. For both transformer models, the input embedding size (D), number of transformer layers, and number of heads are set to 256, 8, 16, respectively. The experiment results and execution time are shown in Table I.

TABLE I
EXPERIMENTAL RESULTS ON SLIP DETECTION DATASET [9]

accuracy \ model / Modality	CNN + LSTM ResNet18	TimeSformer	ViViT
Vision-only	71.7% (0.4%)	78.7% (0.7%)	78.9% (1.2%)
Tactile-only	80.6% (0.8%)	81.0% (0.5%)	81.8% (0.5%)
Vision and tactile	81.9% (0.3%)	**85.0**% (0.4%)	83.9% (0.3%)
Execution time of Feedforward test	9.61s	2.46s	2.43s

TimeSformer and ViViT outperform the CNN + LSTM method by 3:1% and 2:0%, respectively. recorded values are the average across 5 dataset splits and their variances in parenthesis.

2) Experiment Analysis: From Table I, we can see that the transformer models can provide more accurate classification results. Also, when tested on the same dataset split as in Li et al. [9] work, the transformer models can achieve better results (92.3% for TimeSformer and 90.0% for ViViT) than reported in Li et al. [9] work (88.0%) using both sensor inputs. One potential reason for the efficacy of transformer models is that in this application, the final grasping outcomes may be inferred partially from the initial grasping status and transformer models have the capacity of capturing these long-term temporal dependencies more effectively compared with recurrent networks [38]. Another potential reason is related to the architecture of the transformer models: since each transformer layer is stacked in a sequence, spatial and temporal information can be extracted

[1][Online]. Available: https://drive.google.com/file/d/1NPcZYStp2pLPyeWLwv3-jbltz04RUuSp/view?usp=drive_link

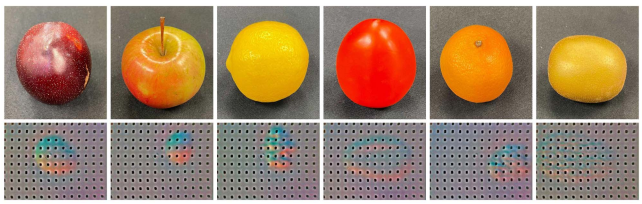


Fig. 5. Top row: Fruits used in experiments. From the left to the right: plums, apples, lemons, tomatoes, oranges, and kiwifruits. Bottom row: the Gelsight images collected at the final frame during pinching for each fruit grasping. It can be seen that the fruit deformation sensed by Gelsight varies as they share different hardness and surface texture.

simultaneously via self-attention mechanism, which does not hold for the CNN + LSTM models.

In addition, it takes significantly less time for feedforward computation of the trained networks during the robotic deployment. As shown in Table I, using both vision and tactile inputs from the same test dataset and selecting the same batch size, the execution time of both TimeSformer and ViViT models on the same machine (NVIDIA GeForce RTX 2070) is 2.46 s (25.6%) and 2.43 s (25.3%), compared to the CNN + LSTM model (ResNet18: 9.61 s, VGG16: 21.39 s). Therefore, it can be concluded that transformer enables the robots to make decisions within a much shorter time.

For the CNN + LSTM model, tactile-only significantly outperforms vision-only, as also reported in Li et al. [9] work. The transformer models perform similarly for each single sensor case while also showing better performance for the multisensor case. This indicates that multisensor input provides better cues for the slip detection.

These advantages highlighted previously altogether motivate us to exploit transformer models on safe fruit grasping.

B. Transformers for Safe Fruit Grasping

In this section, we examine our framework for grasping deformable fruits.

1) Experiment Setup: We collect our own dataset[2] on fruit grasping involving six different types of fruits: plums, oranges, lemons, tomatoes, apples, and kiwifruits, as shown in Fig. 5 (top row). We perform the fruit grasping with various grasping force thresholds (discussed in Section III-C1) on them for 782 times in total to train the models. For each type of fruit, due to the variations in fruit hardness and surface texture, the grasping force thresholds are different as we ensure a balanced training data distribution among the three grasping outcomes (i.e., the count numbers of three grasping outcomes are similar). For example, the sequence of force threshold used for apple (the hardest fruit) is sampled as integers from 4 to 16, and for orange (the softest fruit) it is from 4.0 to 10.0 with 0.5 intervals (both have 13 force thresholds). Since each fruit has a different range of force threshold, we include decimal values to guarantee that the numbers of each fruit samples in training data are close. Each fruit is clearly visible with a black backdrop relative to the camera frame. On the bottom row in Fig. 5, we show that the fruit deformation varies during pinching as they differ in hardness and surface texture. In summary, our collected dataset is comprehensive. For each fruit, it covers a large force range which includes both successful grasping (initial grasp plus transportation) and failures. This comprehensive dataset allows our framework to effectively enable the robot to select the appropriate grasping force that guarantees the success of both the initial grasp and the subsequent transportation. More importantly, our framework could achieve this across different fruits with various textural and hardness. The data are collected by the RealSense camera and GelSight at 30 Hz and with 640×480, 200×150 resolutions, respectively. The visual images are then resized to 160×120 resolution for computational efficiency. For both pinching and sliding actions, we use the *first frame of every three continuous frames* for a total of eight frames (frame index: $1, 4, 7, 10, 13, 16, 19, 22$). For both transformer models, the patch sizes are set as (20, 15) for tactile data and (16, 12) for visual data. The input embedding size (D), number of transformer layers, and number of heads are set to be 256, 16, and 8, respectively.

2) Experimental Evaluation: We aim to address the following questions in this experimental evaluation.

1) Can our transformer models outperform the CNN + LSTM models in terms of grasping outcome prediction for unseen fruits?
2) Is it plausible to deploy the trained frameworks for online fruit grasping applications?
3) What patterns do the transformer models learn from data? In other words, where do the transformer models attend to?

3) Grasping Outcome Prediction on Unseen Fruit: We use a cross-validation technique, partitioning one type of fruit grasping data as a testing set and others as a training set, to compare the accuracy of grasping outcome prediction. After training, the transformer models achieve 80.2% and 76.0% accuracy of grasping outcome prediction on the test dataset (kiwifruit) for TimeSformer and ViViT, respectively. For CNN + LSTM model with Resnet18 as the CNN architecture, it achieves 75.0% accuracy on the test dataset. *building connection with the later training with banana.*

4) Online Fruit Grasping Evaluation: Then, the trained frameworks are deployed on a seven-DOF KUKA LBR iiwa robot manipulator to estimate the safe grasping force via inference for both seen and unseen fruits. During inference, we sample the force thresholds as integers between 4 and 16, and for each sample, we adopt the same fused physical embedding obtained from performing two predefined explorative actions. The robot then grasps each fruit 50 times. Table II shows the success rate and the average on-board computation time for one sample. It can be seen from Table II that the transformer models outperform the CNN + LSTM model significantly, for both seen and unseen fruit grasping.

We demonstrate the total computation time for each successful grasping. As we show in Table II, our framework with ViViT takes 0.29 s for each force threshold sample. In total, we have 13 samples (all integers between 4 and 16), so the total on-board computation time is around 3.77 s. We also allow 15 s for the endure time for the two explorative actions. As a result, for

[2][Online]. Available: https://drive.google.com/file/d/1qBGmeEmLYGI4gPBAbp3y8d3_Yl4u01RH/view?usp=drive_link

TABLE II
EXPERIMENTAL RESULTS OF SUCCESS RATE FOR ONLINE FRUIT GRASPING (50 TRIALS FOR EACH FRUIT) AND THE AVERAGE COMPUTATION TIME

success rate / model / Fruit	CNN + LSTM ResNet18	TimeSformer	ViViT
Plum	66%	88%	86%
Orange	64%	86%	90%
Lemon	60%	90%	90%
Tomato	74%	86%	86%
Apple	68%	92%	92%
Kiwifruit (unseen)	52%	74%	80%
Computation time for one sample	0.52 s	0.39 s	0.29 s

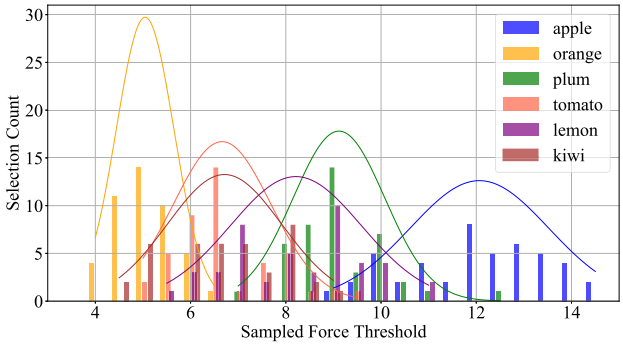


Fig. 6. Times each force threshold candidate is selected for safe grasping. The positional order of the color bars at each sample is shown in the dashed block.

each successful grasping, it takes around 20 s in total, and this indicates our framework can accomplish 180 graspings per hour. It should be noted that since we only use CPU (11th Gen Intel(R) Core(TM) i9-11900 K @ 3.50 GHz) during online deployment, we believe our framework shows the potential for real-time industrial application, with further computational improvement or even GPU implementations.

It is noteworthy that in spite of the *variations in grasping position* caused by manual fruit reloading, *slight fruit spoilage* caused by squeezing, and *unseen fruit type (kiwi)*, the framework is still able to select the safe grasping threshold for each grasp. This shows that the framework demonstrates some level of generalizability under the uncertainty of the local contact surface texture and fruit ripeness. Besides, Fig. 6 shows the times each force threshold candidate is selected for the successful fruit grasping when using ViViT. Their values are proportional to the grasping force that should be exerted on the fruit. It is observed that the selected force thresholds for safe grasping of each fruit distribute over a finite range. Take orange as an example, force threshold 4.5 is selected 11 times for safe grasping of *softer oranges* and 6.0 is also selected five times for *harder oranges*. This force threshold variation indicates our framework's adaptation to the fruit's inherent variability, which is infeasible by hard coding a fixed force threshold, even for the same fruit.

We also test the fruit classification model on the seen fruits during online deployments (video can be found here[3]), which enables the KUKA LBR iiwa robot to place each fruit into separate bins using its built-in position-based waypoint tracking controller after successful grasp. For this, we predefine five different waypoints for each fruit and when the robot reaches the desired waypoint, it would drop off the grasped fruit immediately. In Fig. 7, we show one case that the robot first grasps the orange from the table and then places it in the target bin using the proposed framework. It should be noted that the purpose of this operation is to illustrate that our framework can be potentially used for an integrated pick-and-place task. Therefore, our fruit classification model is not compared with other existing methods since it is not the focus of this work.

[3][Online]. Available: https://www.youtube.com/watch?v=W7o8DsTivTk

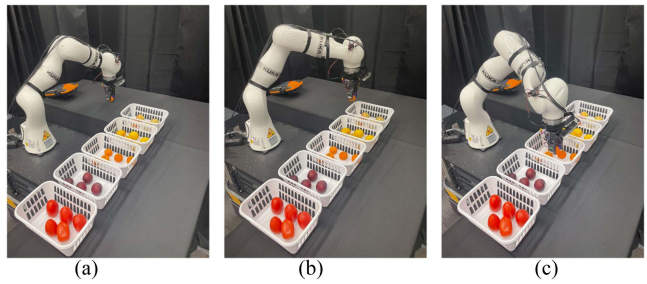


Fig. 7. Snapshots of fruit picking operation. (a) Grasp. (b) Move. (c) Place. Other fruit experiments are shown in the attached video.

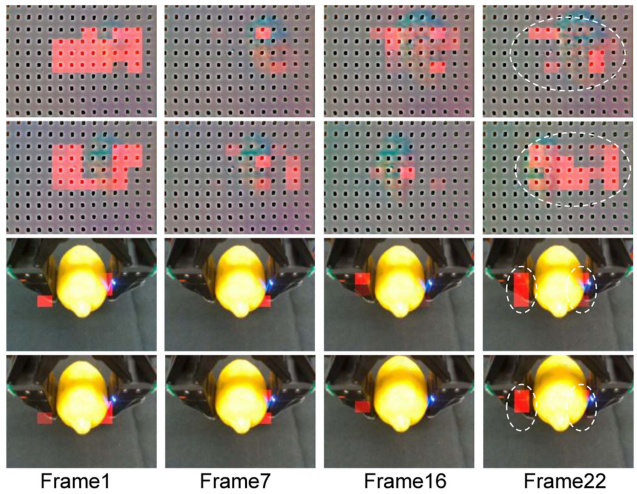


Fig. 8. Visualization of temporal attention from selected image patches at the final frame to their temporally preceding neighbors during a lemon grasping. We only show the results of four frames here. From top to bottom rows, the images are collected from pinching (tactile), sliding (tactile), pinching (visual), and sliding (visual), respectively. The image at each frame is split into 10×10 patches, among which 24 and 6 patches are selected (denoted within the dashed ellipsoid in the final frame) to present the temporal attention for tactile and visual images. The brighter the patch color is, the more attention is paid from its temporal neighbor at the final frame.

5) Attention Analysis: One important component of our framework is the transformer models learned entirely from data. Therefore, we now examine what pattern has our models learned qualitatively. Take TimeSformer as an example, we use the attention rollout method [39] to visualize the learned temporal attentions across vision and tactile image sequences on several selected image patches, as shown in Fig. 8. It can be seen that

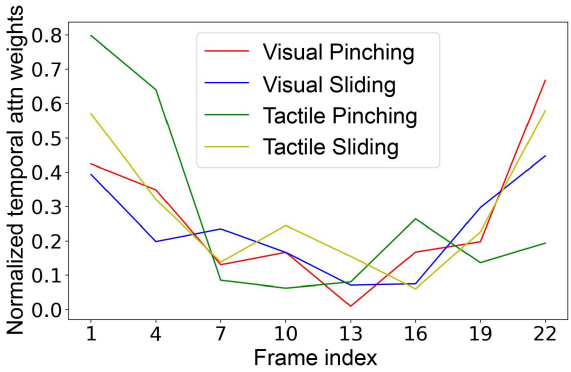


Fig. 9. Normalized temporal attention weights to all selected image patches at each frame from their corresponding temporal neighbors at the final frame.

the image patches at the final frame do not only attend to themselves, but also their temporal neighbors at preceding frames (red color brightness denotes the attention weights). In addition, Fig. 9 shows the normalized temporal attention weights of all selected image patches. An intriguing observation is that the image patches at the first two frames, when the gripper initially touches the objects, share larger attention weights compared with succeeding intermediate frames. Our conjecture of this observation is due to the fact that as the initial and ending contact information is more inferable to the physical status of the manipulated objects, as well as to the grasping outcome. On the contrary, the gradient flow in recurrent networks, even for LSTM architecture, can gradually lose the information on the previous inputs, especially of the first few inputs, resulting in the difficulty of capturing long-term temporal dependencies [40]. However, transformer is able to mitigate this problem as demonstrated.

Furthermore, we show the spatial attention at the final frame for apple, plum, (seen during training), and kiwifruit (unseen during training) grasping in Fig. 10. For tactile images, the TimeSformer model mostly attends to the local contact region, and for visual images, it attends to the fruit surface near gripper's fingertips. Therefore, the transformer models can incorporate more contact information for the grasping task.

We highlight that the interpretability of attention mechanisms may provide an alternative way of analyzing how deep learning methods understand the object's physical deformation properties captured by tactile and visual sensors during contact-rich tasks.

V. Additional Experiments and Analysis

We conduct three more experiments to showcase the sensitivity of our framework in images with large disturbance and the effectiveness of our framework in handling objects with irregular shapes in this section.

A. Sensitivity Analysis of Visual and Tactile Feedback

In this section, we evaluate the sensitivity of our framework in visual and tactile images with different data qualities.

1) Experiment Setup: We employ our proprietary fruit-grasping dataset for sensitivity analysis, aiming to evaluate

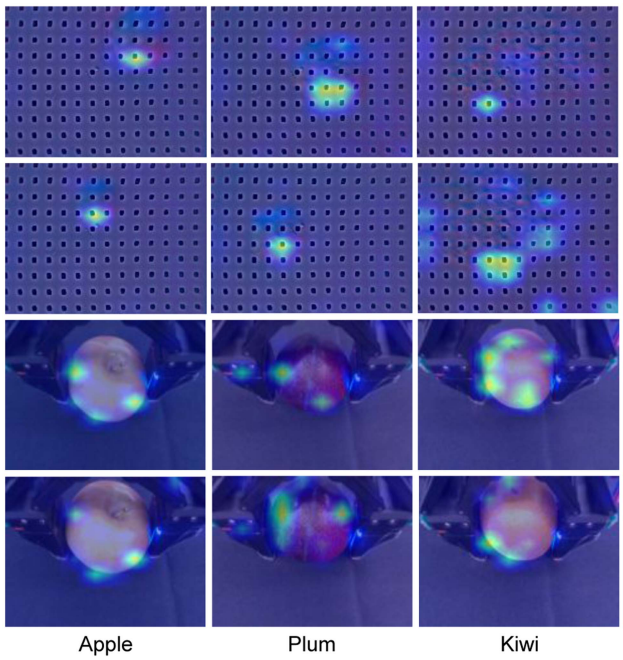


Fig. 10. Visualization of spatial attention from output token to the input image space at Frame 22 during fruit grasping. From top to bottom, the images are collected from pinching (tactile), sliding (tactile), pinching (visual), and sliding (visual), respectively. The image brightness corresponds to spatial attention weights. It is clear that the model mostly attends to the *local contact region* on tactile images and attends to the fruit surface near gripper's fingertips on visual images. It should be noted that kiwi is unseen during training.

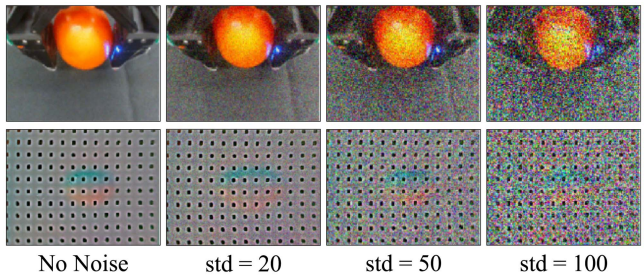


Fig. 11. Top row: Visual images for tomato with different noises. Bottom row: Gelsight images for tomatoes with different noises. From left to right, the Gaussian noises added on images are no noise, $\sigma = 20, 50,$ and 100, respectively.

our framework's performance across varying image qualities. We add different noises to the images in our training set and evaluated the performance with the trained model. We assume Gaussian noise, mimicking natural variations and imperfections in real images. Mathematically, Gaussian noise added on an image is shown as $\mathcal{N}(0, \sigma)$, where $\mathcal{N}(0, \sigma)$ represents the Gaussian distribution with a mean of 0 and a standard deviation of σ, which is a pixel value in range $[0, 255]$.

Gaussian with different standard deviations $(\sigma = 20, 50, 100)$ are added to the images, as shown in Fig. 11. We use apples (hard fruit) and tomatoes (soft fruit) from our training set for analysis. The force thresholds used for apples are integers from 4 to 18, while tomatoes employed force thresholds ranging from 4.0 to 11.0 with 0.5 intervals. For

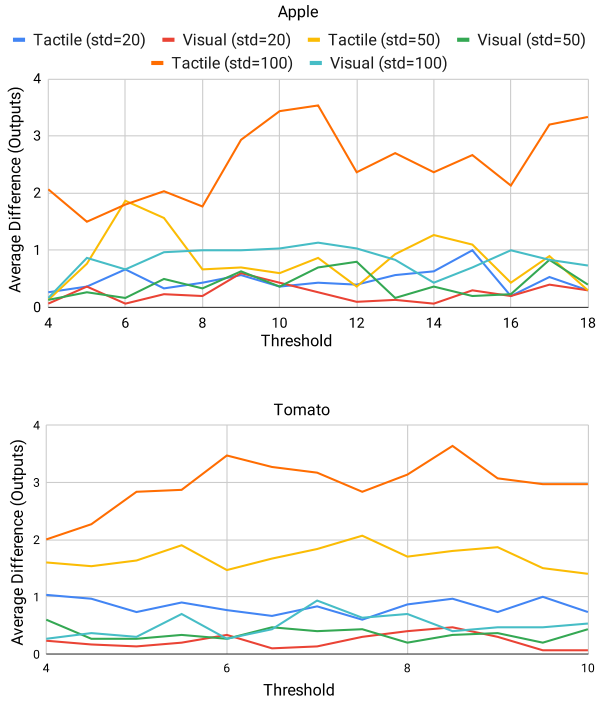


Fig. 12. These two plots show the relationship between sample force thresholds (*x*-axis) and average outcomes difference (*y*-axis) with noise and the outcomes without noise for apples and tomato. The lines with different colors represent different levels of Gaussian noises.

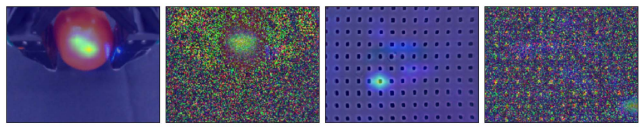


Fig. 13. This figure shows the comparison of spatial attention results for the tomato in both visual and tactile images between zero noise and $\sigma = 100$.

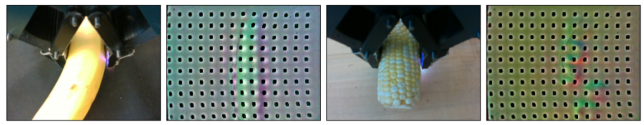


Fig. 14. This figure shows the visual images and Gelsight images for corn and banana.

TABLE III
EXPERIMENTAL RESULTS OF SUCCESS RATE FOR ONLINE FRUIT GRASPING (50 TRIALS FOR EACH FRUIT)

success rate / model / Fruit	CNN + LSTM ResNet18	TimeSformer	ViViT
Corn	46%	68%	72%
Banana	18%	38%	38%

each input, our framework predicted three values, such as $[14.7, 3.7, -27.9]$ for each label of $[0, 1, 2]$ which represented [slipping, safe grasping, and damage], selecting the label with the highest value. To assess the impact of noise and thresholds, ten model predictions are conducted for each level of noise and threshold, calculating the absolute value of the average difference between outcomes with and without noise. For example, an outcome without noise $[14.7, 3.7, -27.9]$, and an average outcome with noise $[14.7, 3.8, -28.0]$, resulted in differences of $[0, 0.1, 0.1]$. The average difference is then calculated as 0.067, representing the final result. This procedure is repeated for tactile and visual images of both apples and tomatoes, encompassing 15 different thresholds and three noise levels.

2) Experiment Evaluation: Upon conducting experiments with both tactile images and visual images of apples and tomatoes, we present the results in Fig. 12. Visual noise analysis reveals the largest difference in outcomes at the highest noise level ($\sigma = 100$). However, when considering the average outcomes in Section V-A1, this discrepancy represents only 6.5% of the outcomes. Hence, visual noise does not significantly impact the model's prediction performance. Conversely, tactile image noise exhibits a more pronounced difference. Substantial tactile noise ($\sigma = 100$) can lead to a difference of approximately 3.5, accounting for 20% of the outcomes. The results in both apples and tomatoes validate the superior robustness of our framework to visual images than tactile images

3) Attention Analysis: In addition, we conducted attention analysis on images with varying levels of noise using the same method, as described in Section IV-B5. Our focus in this and subsequent subsections is to evaluate spatial attention. When visualizing the spatial attention of apple with different noise levels (see Fig. 13), we observe that the TimeSformer model primarily attended to localized contact regions in visual images, even in the presence of high noise levels ($\sigma = 100$). However, under the same noise conditions, the model struggled to effectively attend to the fruit surface in tactile images. These findings further solidify our conclusion on the superior robustness of our framework to visual images than tactile images.

B. Fruit Grasping Evaluation for Unseen Irregular Objects

In this section, we evaluate our framework with unseen irregular objects (i.e., corn and banana) during online experiments, the same as Section IV-B1. Unlike kiwi, corn and bananas are new objects with irregular shapes and different contact surface textures, as shown in Fig. 14. Note that, we use a new gripper in this section for this experiment, which has almost exactly the same mechanical properties as the previous one.

1) Experiment Evaluation: We deploy our model trained from five different round fruits shown in Section IV-B1 on corn and banana to estimate the safe grasping. Similar to the experiments for other fruits, we sample the force threshold as integers between 4 and 16. The robot then grasps each fruit 50 times. Table III shows the success rate.

Analyzing the results depicted in Table III, we observe that our model achieves a high accuracy of 72% for the corn grasping, where the visual and tactile images of the corn have different patterns from those of the round fruits in our training set. This result highlights the robustness of our model in dealing with varying

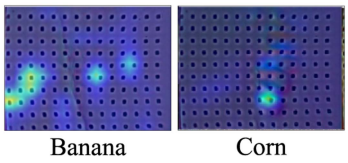


Fig. 15. This figure visualizes the spatial attention for tactile images of a banana and a corm.

visual images. However, the accuracy for bananas is comparatively lower at 38%, indicating limited generalizability when encountering highly varying tactile images. Fig. 14 illustrates the distinctive contact surface textures of a banana compared to the circular contact surface textures typically found in fruits with a round geometry. While the contact surface textures of corn are also irregular, our conjecture is that it consists of multiple smaller circles that bear resemblance to round objects, enabling our model to perform well in corn-grasping tasks.

2) Attention Analysis: We visualize the spatial attention of corn and bananas using the same method as Section IV-B5. Fig. 15 illustrates our model can attend to the contact surface in tactile images of corn, but it is not doable in bananas.

Again, we attribute the reason to the fact that the contact surface of corn is composed of small circles, which are similar to the training objects. However, our model fails to attend to either the banana or the corn in visual images due to their disparate camera views.

C. Fine-Tuning Pretrained Model for a Novel Object Shape

Since our model cannot generalize to the banana because of the significant shape difference, we collect a banana dataset[4] and train a new model to validate that our framework is applicable to bananas. Then, we examine our framework for grasping bananas after fine-tuning a pretrained model on the banana dataset.

1) Experiment Setup: We use the same method as Section IV-B1 to collect a new dataset on banana grasping. We perform the fruit grasping with various grasping force thresholds on them 64 times. The sequence of force thresholds uses for bananas ranges from 3.0 to 6.5 with 0.5 intervals.

2) Training: When acquiring a new dataset, it is common to result in smaller dataset sizes compared to the original dataset. For example, the size of our banana dataset is only one-tenth of that of the original dataset. Owing to the scarcity of data, training the model solely on the new dataset often fails to yield satisfactory performance. In this section, we aim to leverage the model trained on the original dataset described in Section IV-B3 as a pretrained model to solve this challenge.

During the training process, we split the dataset into a training set and a validation set with a ratio of 7 : 1.

When training solely with the banana dataset, each epoch requires approximately 23 s. When using the pretrained model, the time per epoch is reduced to 8 s. This represents a significant reduction in training time, leading to a substantial improvement in training efficiency.

[4][Online]. Available: https://drive.google.com/file/

TABLE IV
EXPERIMENTAL RESULTS OF SUCCESS RATE FOR ONLINE BANANA GRASPING WITH DIFFERENT MODELS (50 TRIALS FOR EACH MODEL)

success rate / model / Fruit	CNN + LSTM ResNet18	TimeSformer	ViViT
Banana (w/o training)	18%	38%	38%
Banana (w/o pretrained)	48%	76%	76%
Banana (with pretrained)	52%	80%	84%

In terms of training results, employing the pretrained model results in a training accuracy of 75%, surpassing the accuracy achieves without the pretrained model by 10%. Moreover, training without the pretrained model exhibits large oscillations in both validation loss and accuracy due to the limited size and single type of the banana dataset. In this case, employing our pretrained model, which is trained on a larger dataset, mitigates these oscillations and enhances the model's performance.

3) Experiment Evaluation: We deploy our model trained from the banana dataset to a real banana grasping experiment to estimate the online grasping results. Similar to the setting in the Section IV-B4, we sample the force thresholds as integers between 3 and 12. For each sample, we adopt the same fused physical embedding obtained from performing two predefined explorative actions. In Table IV, we present the results obtained from three different scenarios: the original model, which is not trained on the banana dataset; the model trained solely on the banana dataset; and the model trained on the banana dataset using the pretrained model.

After training on the banana set, the model reveals a significant performance improvement. This demonstrates the effective learning capabilities of our model, even when dealing with fruits of complex and irregular shapes, leading to enhanced grasping performance. Also, the performance gap becomes more pronounced when handling intricate, irregular objects such as the banana. With the use of the pretrained model, our framework achieves a high success rate of up to 84%, while the baseline model reaches only 52%.

Furthermore, the advantages of employing the pretrained model successfully extend to the online grasping experiment. When using the pretrained model, our framework achieves a higher grasping success rate of 84%, compared to 76% without employing the pretrained model. It highlights the versatility of our model, as it can not only be used in safe grasping for round fruits but also serve as a pretrained model for training on new datasets, even when the new fruit exhibits a completely different and irregular shape.

4) Attention Analysis: Same as Section IV-B5, we examine the performance of spatial attention after training with the banana set. As shown in Fig. 16, the TimeSformer model can mostly attend to the local contact region of visual images and tactile images for both banana and apple (in the original dataset) after training with the banana set, although they have very different shapes and contact surface textures. Therefore, the transformer models can incorporate more contact information for the grasping task, even with fruits with irregular shapes.

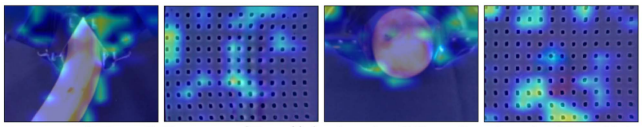


Fig. 16. This figure visualizes the spatial attention of banana and apple after training with the pretrained model.

VI. Conclusions and Discussions

Our experiments demonstrate that the transformer models can enable robotic grasping tasks in both the object classification and robot control domain. The results indicate that they outperform traditional models, such as CNN + LSTM, for classification tasks such as slip detection and grasping outcome prediction. In addition, our transformer-based grasping framework is able to select the grasping strength to safely grasp fruits with varying hardness and surface texture. We also visualize the attention flows of the transformer models, which can potentially explain their effectiveness and efficacy. With the attention analysis, we find that our model has greater robustness with various data qualities in visual images than that in tactile images. Furthermore, our model can effectively learn from fruits with complex and irregular shapes and serve as a pretrained model for training on new datasets. However, it is worth noting that the transformer models are still model-free methods relying on the learned attention from rich data. Performance could be expected to be improved by incorporating model-based methods, such as physical contact models, as future work. Also, improving grasping task robustness and generalization via adversarially regularized policy learning [41] would be another direction to explore. On the other hand, considering another common scenario of grasping objects in a cluttered scene, we can integrate our framework with a high-level task planner to decide the collision-free grasping.

References

[1] C. Blanes, M. Mellado, C. Ortiz, and Ã Valera, "Review technologies for robot grippers in pick and place operations for fresh fruits and vegetables," *Spanish J. Agricultural Res.*, vol. 9, pp. 1130–1141, 2011.

[2] Z. Xu, J. Wu, A. Zeng, J. B. Tenenbaum, and S. Song, "DensePhysNet: Learning dense physical object representations via multi-step dynamic interactions," in *Proc. Conf. Robot. Sci. Syst.*, 2019.

[3] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 127–135, 2015.

[4] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, pp. 54–67, 2017.

[5] S. Wang, Y. She, B. Romero, and E. Adelson, "Gelsight wedge: Measuring high-resolution 3D contact geometry with a compact robot finger," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6468–6475.

[6] W. Yuan, S. Dong, and E. H. Adelson, "GelSight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, 2017, Art. no. 2762.

[7] N. Kuppuswamy, A. Alspach, A. Uttamchandani, S. Creasey, T. Ikeda, and R. Tedrake, "Soft-bubble grippers for robust and perceptive manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9917–9924.

[8] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, "SwingBot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5633–5640.

[9] J. Li, S. Dong, and E. Adelson, "Slip detection with combined tactile and visual information," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7772–7777.

[10] R. Calandra et al., "The feeling of success: Does touch sensing help predict grasp outcomes?" in *Proc. 1st Annu. Conf. Robot Learn.*, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., 2017, vol. 78, pp. 314–323. [Online]. Available: https://proceedings.mlr.press/v78/calandra17a.html

[11] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?," in *Proc. Int. Conf. Mach. Learn.*, 2021, Art. no. 4.

[12] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6816–6826.

[13] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, and F. Iida, "Soft manipulators and grippers: A review," *Front. Robot. AI*, vol. 3, 2016, Art. no. 69.

[14] R. Li and H. Qiao, "A survey of methods and strategies for high-precision robotic grasping and assembly tasks—some new trends," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 6, pp. 2718–2732, Dec. 2019.

[15] M. R. Hasan, R. Vepa, H. Shaheed, and H. Huijberts, "Modelling and control of the Barrett Hand for grasping," in *Proc. UKSim 15th Int. Conf. Comput. Modelling Simul.*, 2013, pp. 230–235.

[16] L. Zaidi, J. A. C. Ramon, L. Sabourin, B. C. Bouzgarrou, and Y. Mezouar, "Grasp planning pipeline for robust manipulation of 3D deformable objects with industrial robotic hand arm systems," *Appl. Sci.*, vol. 10, no. 23, 2020, Art. no. 8736. [Online]. Available: https://www.mdpi.com/2076-3417/10/23/8736

[17] W. Friedl and M. A. Roa, "Clash—a compliant sensorized hand for handling delicate objects," *Front. Robot. AI*, vol. 6, 2020, Art. no. 138.

[18] P. Chang and T. Padır, "Model-based manipulation of linear flexible objects: Task automation in simulation and real world," *Machines*, vol. 8, no. 3, 2020, Art. no. 46. [Online]. Available: https://www.mdpi.com/2075-1702/8/3/46

[19] L. Pozzi et al., "Grasping learning, optimization, and knowledge transfer in the robotics field," *Sci. Rep.*, vol. 12, no. 1, pp. 1–11, 2022.

[20] J. Jiang, G. Cao, A. Butterworth, T.-T. Do, and S. Luo, "Where shall i touch? Vision-guided tactile poking for transparent object grasping," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 1, pp. 233–244, Feb. 2023.

[21] R. Calandra et al., "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3300–3307, Oct. 2018.

[22] D. Kim, J. Lee, W.-Y. Chung, and J. Lee, "Artificial intelligence-based optimal grasping control," *Sensors*, vol. 20, no. 21, 2020, Art. no. 6390.

[23] Y. Zhang, W. Yuan, Z. Kan, and M. Y. Wang, "Towards learning to detect and predict contact events on vision-based tactile sensors," in *Proc. Conf. Robot Learn.*, 2020, pp. 1395–1404.

[24] S. Dong, D. Ma, E. Donlon, and A. Rodriguez, "Maintaining grasps within slipping bounds by monitoring incipient slip," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3818–3824.

[25] N. Wettels, A. R. Parnandi, J.-H. Moon, G. E. Loeb, and G. S. Sukhatme, "Grip control using biomimetic tactile sensing systems," *IEEE/ASME Trans. On Mechatron.*, vol. 14, no. 6, pp. 718–723, Dec. 2009.

[26] J. Lin, R. Calandra, and S. Levine, "Learning to identify object instances by touch: Tactile recognition via multimodal matching," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3644–3650.

[27] W. Yuan, S. Wang, S. Dong, and E. Adelson, "Connecting look and feel: Associating the visual and tactile properties of physical materials," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4494–4502.

[28] A. Vaswani et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 6000–6010, 2017.

[29] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[30] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2X-ViT: Vehicle-to-everything cooperative perception with vision transformer," in *Proc. Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Springer, 2022, pp. 107–124.

[31] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proc. 6th Conf. Robot Learn.*, 2023, vol. 205, pp. 785–799. [Online]. Available: https://proceedings.mlr.press/v205/shridhar23a/shridhar23a.pdf

[32] M. Monastirsky, O. Azulay, and A. Sintov, "Learning to throw with a handful of samples using decision transformers," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 576–583, Feb. 2023.

[33] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," in *Proc. Int. Conf. Learn. Representations*, 2022.

[34] A. Nagrani, S. Yang, A. Arnab, A. Jansen, C. Schmid, and C. Sun, "Attention bottlenecks for multimodal fusion," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 14200–14213, 2021.
[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. North Amer. Chap. Ass. Comput. Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399
[36] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *Int. J. Robot. Res.*, vol. 40, no. 12–14, pp. 1385–1401, 2021.
[37] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, San Diega, CA, USA, 2015.
[38] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha, "Transformer Hawkes process," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11692–11702.
[39] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proc. 58th Annu. Meeting Ass. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., 2020, pp. 4190–4197. [Online]. Available: https://aclanthology.org/2020.acl-main.385
[40] S. Hochreiter et al., "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks*. Hoboken, NJ, USA: Wiley, 2001.
[41] Z. Zhao, S. Zuo, T. Zhao, and Y. Zhao, "Adversarially regularized policy learning guided by trajectory optimization," in *Proc. 4th Annu. Learn. Dynam. Contr. Conf.*, R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, and M. Kochenderfer, Eds., 2022, vol. 168, pp. 844–857.

Yunhai Han (Student Member, IEEE) received the B.S. degree in mechanical engineering from Yanshan University, Qinhuangdao, China, in 2019, and the M.S. in mechanical engineering from University of California San Diego, San Diego, CA, USA, in 2021. He is currently working toward the Ph.D. degree in robotics with the Georgia Institute of Technology, Atlanta, GA, USA.

Kelin Yu (Student Member, IEEE) received the B.S. degree in electrical engineering and mathematics from the Georgia Institute of Technology, Atlanta, GA, USA, in 2022, where he is currently working toward the M.S. degree in computer science.

Rahul Batra (Student Member, IEEE) received the bachelor's degree in computer engineering from The University of Texas at Austin, Austin, TX, USA, in 2007, the master's degree in software engineering from The University of Texas at Austin, in 2017, and the master's degree in computer science from the Georgia Institute of Technology, Atlanta, GA, USA, in 2020.

He is currently a Senior Member of technical staff with Advanced Micro Devices Inc., Santa Clara, CA, USA.

Nathan Boyd (Student Member, IEEE) received the bachelor's degree in mechanical engineering from California State University, Northridge, Northridge, CA, USA, in 2020, and the master's degree in mechanical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2022.

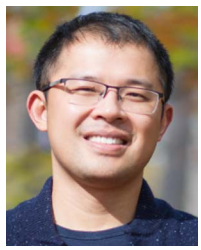
He is currently a Senior Robotics Engineer with Apptronik, Austin, TX, USA.

Chaitanya Mehta (Student Member, IEEE) received the B.Tech. degree in mechanical engineering from the Indian Institute of Technology Indore, Indore, India, in 2020. He is currently working toward the M.S. degree in robotics with the Georgia Institute of Technology, Atlanta, GA, USA.

His research interests include legged locomanipulation and tactile sensing.

Tuo Zhao (Member, IEEE) received the Ph.D. degree in computer science from Johns Hopkins University, Baltimore, MD, USA, in 2016.

He is currently an Assistant Professor with the School of Industrial and Systems Engineering and the School of Computational Science and Engineering (By Courtesy), Georgia Institute of Technology, Atlanta, GA, USA. His research focuses on nonconvex optimization algorithms and practical theories for machine learning.

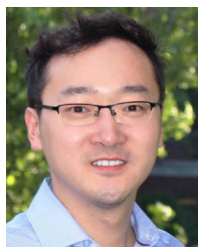


Yu She (Member, IEEE) received the Ph.D. degree in mechanical engineering from Ohio State University, Columbus, OH, USA, in 2018.

He is currently an Assistant Professor with the School of Industrial Engineering, Purdue University, West Lafayette, IN, USA. From 2018 to 2021, he was a Postdoctoral Researcher with Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA. His research interests include design, modeling, sensing, control of intelligent mechanical systems, soft robotics, human-safe collaborative robots, tactile sensing, and robotic manipulation.

Seth Hutchinson (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently the Executive Director of the Institute for Robotics and Intelligent Machines at the Georgia Institute of Technology, Atlanta, GA, USA, where he is also a Professor and the KUKA Chair for robotics with the School of Interactive Computing. He is currently an Emeritus Professor of electrical and computer engineering (ECE) with the University of Illinois in Urbana-Champaign (UIUC), Champaign, IL, USA. In 1990, he joined UIUC, where he was a Professor of ECE until 2017, and the Associate Department Head of ECE from 2001 to 2007.



Ye Zhao (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from The University of Texas at Austin, Austin, TX, USA, in 2016.

He is currently an Assistant Professor with the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA. He was a Postdoctoral Fellow with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. His research interests include robust task and motion planning, contact-rich trajectory optimization, formal methods for legged locomotion and navigation.