Monocular Vision Based Control Framework for Grasping

Shail Jadav[1] and Dongheui Lee[1,2]

***Abstract*— Grasping in unstructured environments requires handling objects with widely different mechanical properties, from soft and deformable items to rigid everyday objects. Most existing approaches address these categories separately and often rely on tactile sensing, object-specific models, or specialized grippers. In this paper, we present a unified monocular vision-based grasping framework that targets both soft and rigid objects within a single control pipeline, using only RGB input and a position-controlled gripper. The proposed system combines open-vocabulary object detection, image segmentation, boundary-aware point assignment, real-time point tracking, and monocular depth estimation to recover object motion and geometry from visual observations. A key component of the framework is a language-based stiffness estimation model that infers an object's expected compliance from its semantic description and provides an object-level prior for selecting the grasping strategy before contact. For deformable objects, grasp adaptation is governed by a Procrustes-based dissimilarity measure computed from tracked keypoints, which acts as a visual proxy for deformation. For rigid objects, the gripper width is regulated through the scaling of tracked point distances. We validate the proposed method in real-world pick-and-place experiments on a Franka Emika Research 3 arm using objects with substantially different mechanical properties, including lettuce, fresh mozzarella cheese, croissants, paper towels, and hard plastic bottles. Results demonstrate that the framework achieves stable grasping across both soft and rigid objects using visual feedback alone, highlighting a practical, sensor-efficient, and generalizable approach for food handling and household manipulation.**

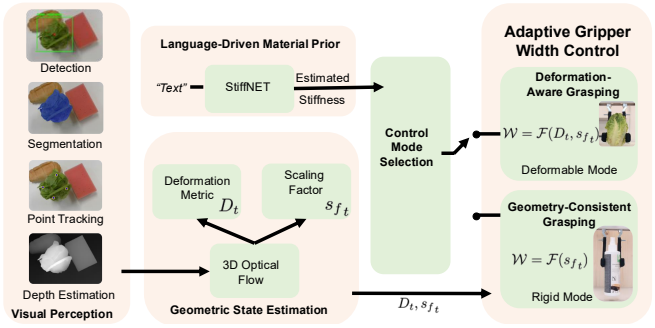

Fig. 1: **Overview of the proposed framework.** A unified monocular vision-based grasping system enables a standard position controlled gripper to handle both compliant and rigid objects using RGB input alone. By combining semantic priors from language with visual feedback during manipulation, the framework selects an appropriate grasping behavior before contact and adapts online to maintain stable grasps across diverse everyday objects.

I. INTRODUCTION

As robots increasingly integrate into our daily lives, their ability to manipulate various objects, particularly deformable items, becomes crucial [1]. These objects include a wide range of items, from food products to flexible materials like cloths and rigid objects. The successful grasping of deformable objects offers significant potential for many industries, especially in food processing and home automation [2]. However, the manipulation of deformable objects presents a unique challenge due to their high degrees of freedom and variable material properties [2]. Traditional force-based grasping methods are often inadequate for manipulating deformable objects, as they do not consider object deformation [2], [3]. Consequently, advanced sensing and control strategies are necessary to overcome these challenges. One promising advancement is the integration of vision-based tactile sensors in grasping systems [4]–[9]. These sensors operate by using cameras to monitor deformations in elastomeric surfaces, allowing the detection of both normal and shear forces during grasping. While vision-based tactile sensors offer significant advantages in enhancing grasping precision, they also present limitations. Prior studies have shown that objects softer than the sensor elastomer produce weaker and less distinctive tactile signatures, as they undergo greater deformation at the contact interface than the elastomer itself [10]. Another major drawback is the degradation of elastomers over time with continued use, which leads to reduced sensitivity and performance variability. Furthermore, the cost of these sensors is substantial, including both the initial cost of the system and the ongoing expense of replacing worn elastomers [11].

Another approach to deformable object manipulation involves mechanics based object modelling, such as those utilizing finite element analysis (FEA) [18]. FEA-based methods are commonly used to model strain and control the forces required for manipulating deformable objects [12], [13], [19]. Recent work has also explored using large language models (LLM) to estimate object interaction properties (e.g., mass, friction, and an effective stiffness/compliance proxy) from semantic descriptions to guide grasping policies [14]. These methods can be highly effective when the material properties and environmental dynamics are accurately modelled. However, in real-world applications, obtaining precise material properties is challenging. Many deformable objects exhibit time-varying and nonlinear behaviors, making it difficult to maintain accurate models over time and across varying conditions. For example, material properties such as

[1]Shail Jadav and Dongheui Lee are with Autonomous Systems, Technische Universität Wien (TU Wien), Vienna, Austria (e-mail: shail.jadav@tuwien.ac.at, dongheui.lee@tuwien.ac.at).

[2]Dongheui Lee is also with the Institute of Robotics and Mechatronics (DLR), German Aerospace Center, Wessling, Germany.

TABLE I: Comparison with representative prior approaches in terms of their core enabling assumption, use of semantic priors, and demonstrated object regime. Prior work typically derives robustness from tactile instrumentation, explicit mechanics models, or specialized end-effector design. In contrast, the proposed framework operates with monocular RGB and a standard position-controlled gripper while addressing both deformable and rigid objects.

Approach family	**Representative works**	**Core enabling assumption**	**Semantic prior**	**Demonstrated scope**
Vision–tactile deformable grasping	[4]	Vision-based tactile sensing integrated with a position-controlled gripper	No	Deformable-object grasping
Vision-based tactile sensing and slip monitoring	[5]–[9]	Dedicated tactile fingertips or tactile sensing platforms	No	Contact-rich stabilization and tactile behaviors
Mechanics-based deformable-object control	[12], [13]	Explicit deformation modeling, often combined with richer 3D perception	No	Deformable and soft-object manipulation
Semantic adaptive grasping	[14]	Force-controllable gripper with gripper-side force/depth sensing	Yes	Adaptive grasping from delicate to rigid objects
Specialized compliant end-effectors	[15]–[17]	Soft, compliant, or variable-stiffness end-effector design	No	Primarily delicate and deformable-object handling
Proposed framework	**This work**	**Monocular RGB perception with a standard position-controlled gripper**	**Yes**	**Unified grasping of deformable and rigid objects**

stiffness in food products change over time due to factors like moisture content or age, complicating the modelling process [20].

Recent advancements in gripper technology have focused on the development of specialized structures for grasping deformable objects [15]. In particular, soft grippers have shown significant promise in handling delicate items such as food products [16]. These grippers are typically actuated either by pneumatic pressure or tendons, and some utilize shape memory alloys for actuation [17]. Due to their compliance, soft grippers are well-suited for tasks requiring gentle handling, as they do not exert large forces, which is ideal for fragile objects. However, this very compliance poses a limitation when it comes to the generalization of these grippers. Specifically, they struggle to generate the necessary forces required for manipulating rigid objects effectively, thereby limiting their versatility in more demanding applications [21].

As advancements in robotic grasping technologies continue to progress, significant inspiration can be drawn from the way humans manipulate objects. Humans skillfully use the same hands to handle a wide variety of objects, both deformable and rigid, by seamlessly integrating visual and tactile feedback [22]–[24]. While tactile feedback is crucial for fine manipulation and adjusting grip forces, especially during rapid movements where visual feedback may lag, humans can still perform grasping tasks relying solely on visual feedback, albeit with reduced precision and increased effort compared to when both sensory modalities are utilized [25], [26]. Beyond perception, humans also exploit semantic knowledge: object and material words can provide a prior over expected mechanical properties (e.g., soft vs. stiff), with material names alone eliciting a structured softness representation comparable to visually and haptically derived spaces [27]. Inspired by this human adaptability, we aim to develop a framework that relies exclusively on visual feedback for robotic grasping tasks. This approach seeks to simplify sensory requirements and enhance the applicability of robotic systems in environments where tactile sensing is impractical or unavailable.

With advancements in grasping technologies, there have also been significant breakthroughs in the field of computer vision, which can greatly enhance object-grasping capabilities. Recent progress in object detection and image segmentation is exemplified by the release of SAM 2, a transformer-based foundational model for image segmentation, which excels at segmenting objects of interest in cluttered environments [28]. For grasping deformable objects, real-time state tracking can be achieved using point-tracking algorithms like TAPIR, developed by Google DeepMind, which tracks individual query points [29]. On the other hand, Meta's CoTracker, designed for tracking multiple points simultaneously, is more suited for rigid objects [30]. Moreover, recent developments in depth estimation models enable the inference of relative depth from RGB images [31]. By combining 2D point tracking with relative depth information, one can track query points on deformable objects in real-time and in 3D space, offering valuable insights into object deformation during grasping. This approach could be utilized to dynamically adjust the gripper's width for improved control during grasping tasks. As a result, this paper presents a monocular vision-based framework for grasping both deformable and rigid objects without tactile sensing, specialized grippers, or explicit mechanics-based object models, as illustrated in Fig. 1 and Table I. The core technical contributions of this work are as follows:

- We present a unified monocular vision-based grasping framework that enables a standard position controlled gripper to handle both compliant and rigid objects using RGB input alone.
- We show that semantic knowledge from language can provide useful object-level priors about compliance before contact, allowing the system to choose an appropriate grasping behavior without tactile sensing or explicit mechanics models.
- We demonstrate that visual feedback alone can support

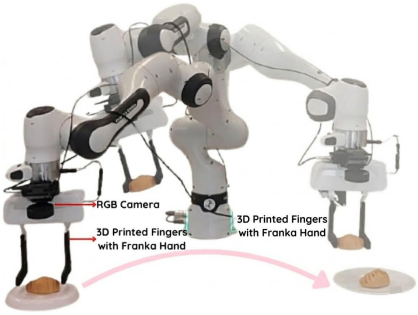

Fig. 2: **Experimental setup for grasping & placing experiment**:Involving a generic USB camera, 3D-printed fingers, and a Franka arm with gripper.

adaptive and stable grasping across a broad range of everyday objects, highlighting a practical and sensor-efficient approach for food handling and household manipulation.

Finally, we validate the proposed system through a series of experiments with real-world objects, showcasing its potential in real-world scenarios. The framework is implemented using a position-controlled Franka Emika Hand (gripper), allowing for precise grasping, as shown in Fig. 2. We evaluate the performance of our method on objects that are challenging to manipulate, such as fresh mozzarella cheese, lettuce, croissants, and paper towels, as well as rigid objects like hard plastic bottles, as shown in Fig 3. These objects present unique challenges for robotic grasping due to their diverse physical properties. Lettuce and croissants exhibit nonhomogeneous and anisotropic material properties, leading to unpredictable mechanical responses during manipulation. Fresh mozzarella cheese is slippery, introducing slippage issues that complicate stable grasping. Paper towels and rigid plastic bottles serve as examples of everyday items with differing rigidity and surface textures, further testing the adaptability and robustness of our grasping method. Our framework successfully completed pick-and-place operations with these objects, demonstrating its potential for food handling and home automation applications.

II. METHODS

In this section, we present our framework for grasping both deformable and rigid objects. The object detection algorithm receives a prompt of the object of interest and camera feed as input, identifying the object's location within the frame. This location is then passed to an image segmentation network, which isolates the object of interest from the background. The resulting segmented mask is forwarded to the grid point assignment algorithm, where query points for tracking are selected. These tracking query points are transmitted to a tracking network, providing real-time 2D coordinates within the image. Depth information is subsequently acquired using a depth estimation model, and with both 2D coordinates and depth, a full 3D representation of the object is constructed. Procrustes analysis is then performed by comparing the current tracking points with the points from the initial conditions to assess dissimilarity, which correlates with the force exerted by the gripper on deformable objects. This dissimilarity arises from the deformation of key points within the object. The gripper controller uses this dissimilarity to adapt the gripper's width, ensuring a stable grasp. For rigid objects, the gripper width is directly controlled based on the scaling factor derived from the ratio of the pairwise distance between the tracked points to their pairwise distance at the initial condition. The system utilizes monocular vision and real-time control of the Franka Hand. An overview of the proposed framework is depicted in Fig. 1.

A. Object Detection & Segmentation

Let $\mathbf{I} \in {\mathbb{R}_+}^{3\times H\times W}$ represent the input RGB image frame, and let $\mathbf{p}$ denote the object of interest $\mathbf{O}$. The image frame $\mathbf{I}$, together with the textual prompt $\mathbf{p}$ specifying the object of interest, is passed to the object detection algorithm. Our framework employs the YOLOv8x-worldv2 model for real-time open-vocabulary object detection [32]. The output of the detection process provides the coordinates of $\mathbf{O}$ as

$$\mathbf{o}_c = (x_2 - c, \frac{y_1 + y_2}{2}), \tag{1}$$

where $x_2, y_1, \& \ y_2$ are the bounding box coordinates for the object $\mathbf{O}$, and $c \in \mathbb{Z}_+$ is a user-defined constant scalar value, allowing the selection of segments near the edge of the bounding box. The coordinates $\mathbf{o}_c \in \mathbb{Z}_+^2$ are passed to the Meta SAM2 (Segment Anything Model [28]) for image segmentation, which generates the mask $\mathbf{M} \in \{0,1\}^{H\times W}$ corresponding to $\mathbf{O}$. The mask $\mathbf{M}$ is then provided to the point assignment algorithm to create query points for the tracking point network. Here we employ Tracking Any Point with per-frame Initialization and Temporal Refinement (TAPIR) model for point tracking [29]. Because point tracking degrades on low-texture objects, query-point assignment is formulated as a boundary-weighted clustering problem that selects evenly distributed, maximally separated points near the mask contour to improve tracking robustness.

B. Point Assignment

We begin by extracting the contours from the binary mask $\mathbf{M}$, which defines the region of interest. The contours represent the boundaries between these distinct regions in the mask. From the detected contours, we select the largest contour, denoted $\mathcal{C}$, based on the area enclosed by the contour. Next, we gather all the points within the mask where the pixel value is positive. These points form the set $\mathcal{P}$, which is defined as:

$$\mathcal{P} = \{(x, y) \mid \mathbf{M}(x, y) > 0\}. \tag{2}$$

For each point $\mathbf{p} = (x, y) \in \mathcal{P}$, we compute its distance to the nearest contour point $\mathbf{c} \in \mathcal{C}$. This distance is defined as

$$d(\mathbf{p}) = \min_{\mathbf{c}\in\mathcal{C}} \|\mathbf{p} - \mathbf{c}\|. \tag{3}$$

We assign a weight to each point $\mathbf{p} \in \mathcal{P}$ inversely proportional to its distance from the contour as

$$w(\mathbf{p}) = \frac{1}{1 + d(\mathbf{p})}. \tag{4}$$

This weighting scheme emphasizes points closer to the contour, assigning them larger weights. We then apply a weighted k-means clustering algorithm to partition the set of points $\mathcal{P}$ into k clusters, where k is the desired

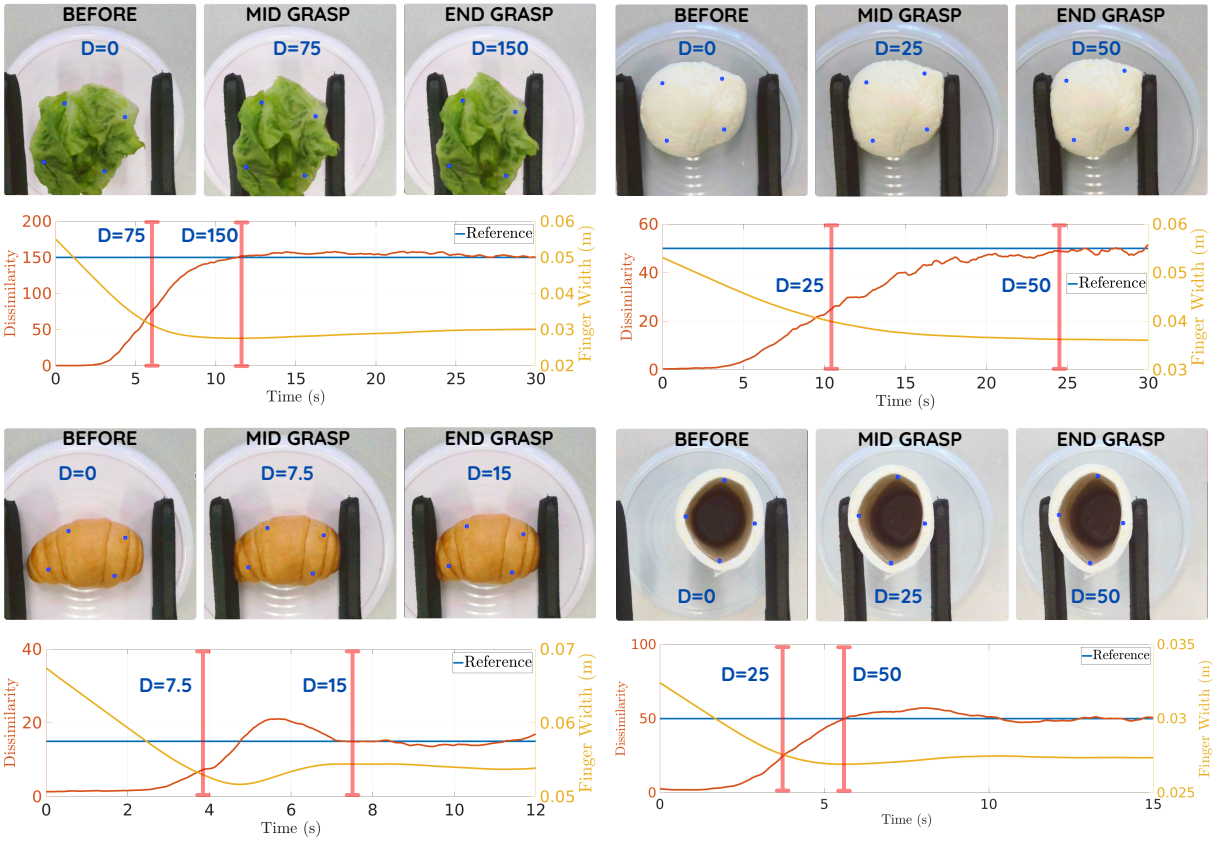

Fig. 3: **Grasping multiple objects**: We demonstrate successful grasping during a pick-and-place operation for deformable objects, including lettuce, mozzarella cheese, croissant bread, and a paper roll. The proposed controller adjusts the gripper finger width to track the desired dissimilarity and maintain a stable grasp throughout the manipulation. D indicates the dissimilarity of the tracking points compared to the initial condition, and red lines show the instances of particular dissimilarity values.

number of grid points selected by the user. The k-means clustering algorithm seeks to minimize the within-cluster variance by optimizing the placement of cluster centers $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k\}$, which will serve as the final grid points. The objective function to be minimized is

$$\mathcal{J}(\mathbf{G}) = \sum_{i=1}^{k} \sum_{\mathbf{p}_j \in \mathcal{K}_i} w(\mathbf{p}_j) \|\mathbf{p}_j - \mathbf{g}_i\|^2, \tag{5}$$

where $\mathbf{p}_j \in \mathcal{P}$ represents a point in the mask, $\mathcal{K}_i$ is the set of points assigned to the i-th cluster, $\mathbf{g}_i \in \mathbb{R}^2$ is the center of the i-th cluster, and $w(\mathbf{p}_j)$ is the weight of point $\mathbf{p}_j$. The cluster centers $\mathbf{g}_i$ are updated iteratively using the weighted mean of the points assigned to each cluster

$$\mathbf{g}_i = \frac{\sum_{\mathbf{p}_j \in \mathcal{K}_i} w(\mathbf{p}_j) \mathbf{p}_j}{\sum_{\mathbf{p}_j \in \mathcal{K}_i} w(\mathbf{p}_j)}. \tag{6}$$

This process repeats until convergence, ensuring that the cluster centers minimize the weighted variance of points in each cluster. As a result, the grid points $\mathbf{G}$ are distributed across the mask with a denser concentration near the contour due to the weighting scheme. After clustering, we verify that each grid point $\mathbf{g}_i = (x_i, y_i) \in \mathbf{G}$ lies within the mask $\mathbf{M}$. Specifically, for each $\mathbf{g}_i$, we check $\mathbf{M}(x_i, y_i) > 0$. If any grid points lie outside the mask, they are discarded. If the number of valid grid points is less than the desired number k, we iteratively refine the grid by adding new points. To do this, we identify the pair of points $(\mathbf{g}_i, \mathbf{g}_j) \in \mathbf{G}$ that are farthest apart $(\mathbf{g}_i, \mathbf{g}_j) = \arg\max_{i,j} \|\mathbf{g}_i - \mathbf{g}_j\|$. The midpoint between $\mathbf{g}_i$ and $\mathbf{g}_j$ is computed as $\mathbf{g}_{\text{new}} = \frac{\mathbf{g}_i + \mathbf{g}_j}{2}$. If $\mathbf{g}_{\text{new}} \in \mathbf{M}$, it is added to the set $\mathbf{G}$; otherwise, the closest valid point within $\mathcal{P}$ is selected. The final set of grid points $\mathbf{G}$ is distributed within the mask, with emphasis on regions closer to the contour.

C. Point Tracking and Depth Estimation

The grid points $\mathbf{G}$, generated during the point assignment process, serve as input query points to the TAPIR model, which is optimized for efficiently tracking arbitrary points in video sequences. TAPIR operates in two stages. In the matching stage, it independently identifies candidate matches for each query point across frames. Then, in the refinement stage, it updates the point trajectories and query features by utilizing local correlations, improving both tracking accuracy and consistency over time. The output of TAPIR is the set of tracked points $\mathbf{G}^t$, where t represents time. Once real-time 2D tracking points are obtained from TAPIR, we apply the Depth Anything model for monocular depth estimation, converting the 2D points into 3D tracking points in space [31]. Both TAPIR and Depth Anything operate in real-time, enabling dynamic point tracking even in complex environ-

ments. Additionally, depth can be inferred using a scaling factor, assuming the object remains within the same plane of reference and does not rotate. This scaling factor assists in depth estimation, while the Depth Anything model provides relative depth estimates, which enhance the determination of object orientation in 3D space.

D. Procrustes Analysis

Procrustes analysis is used to compare the shapes of two data matrices by optimally transforming one to match the other. The objective is to minimize the dissimilarity between the two matrices. This transformation involves applying translations, rotations, reflections, and scaling to achieve the best fit. In this context, we calculate the dissimilarity between the initial grid points $\mathbf{G}^0$ and the current grid points $\mathbf{G}^t$. The $\mathbf{G}^0$ is the output of the point assignment algorithm and $\mathbf{G}^t$ is the output of TAPIR with Depth Anything model, only TAPIR used if orientation is not required. First, both grid points are centered by subtracting their mean vectors, ensuring that their centroids coincide with the origin

$$\mathbf{G}_c^0 = \mathbf{G}^0 - \mathbf{1}_n \bar{\mathbf{G}}^0, \quad \mathbf{G}_c^t = \mathbf{G}^t - \mathbf{1}_n \bar{\mathbf{G}}^t, \tag{7}$$

where $\bar{\mathbf{G}}^0$ and $\bar{\mathbf{G}}^t$ are the mean vectors of the respective grid points, and $\mathbf{1}_n$ is an n-dimensional vector of ones. Further, we normalize the grid points as

$$\mathbf{G}_n^0 = \frac{\mathbf{G}_c^0}{\|\mathbf{G}_c^0\|}, \quad \mathbf{G}_n^t = \frac{\mathbf{G}_c^t}{\|\mathbf{G}_c^t\|}. \tag{8}$$

After the normalization, we find the optimal rotation and scaling factor for the minimal dissimilarity. We perform singular value decomposition to find optimal rotation matrix and scaling factor as

$$\mathbf{U}, \mathbf{S}, \mathbf{V}^\top = \text{SVD}(\mathbf{G}_n^{0\top} \mathbf{G}_n^t). \tag{9}$$

The optimal rotation matrix $\mathbf{R}$ and the scaling s are given by

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\top, \quad s = \sum \mathbf{S}. \tag{10}$$

Finally, the dissimilarity D_t between the grid points is computed as

$$D_t = \Gamma \sum_{i=1}^{k} \|\mathbf{G}_n^{0\,(i)} - s\mathbf{R}\mathbf{G}_n^{t\,(i)}\|^2, \tag{11}$$

where $\Gamma \in \mathbb{R}_+$ is the sensitivity gain. The dissimilarity D_t provides a quantitative assessment of the dissimilarity between the two gridpoints, where smaller values of D_t indicate greater similarity. We further define scaling factor as

$$s_{f_t} = \frac{\sqrt{\sum_{i=1}^{k}\sum_{j=1}^{k} \|\mathbf{g}_i^t - \mathbf{g}_j^t\|^2}}{\sqrt{\sum_{i=1}^{k}\sum_{j=1}^{k} \|\mathbf{g}_i^0 - \mathbf{g}_j^0\|^2}}, \tag{12}$$

where $g^t \in \mathbf{G}^t$ and $g^0 \in \mathbf{G}^0$ are the individual grid points.

E. Language-Based Stiffness Estimation (StiffNET)

We estimate object stiffness directly from language by learning a scalar mapping from an object's text description to its physical stiffness. The key idea is to embed each object name into a semantic feature space and then learn a function that places objects on a one-dimensional *log-stiffness axis*. Training combines two complementary supervision sources: 1) pairwise hardness comparisons, which provide relative ordering, and 2) sparse ground-truth stiffness measurements, which anchor the absolute scale.

a) Text representation.: Let $x \in \mathcal{X}$ denote the text associated with an object, such as its category name or short semantic descriptor. A pretrained text encoder $\mathcal{E}(\cdot)$ maps x to a normalized embedding

$$\mathbf{z}(x) = \mathcal{E}(x) \in \mathbb{R}^d, \qquad \|\mathbf{z}(x)\|_2 = 1. \tag{13}$$

In our implementation, $\mathcal{E}$ is a frozen CLIP text encoder. Thus, the semantic representation is fixed during training, and only the downstream stiffness predictor is learned.

b) Stiffness network.: The main trainable model is a neural network $f_\theta : \mathbb{R}^d \to \mathbb{R}$ that predicts the *log-stiffness* of an object:

$$\hat{\ell}(x) = f_\theta(\mathbf{z}(x)) + b_\theta, \tag{14}$$

where $b_\theta \in \mathbb{R}$ is a learned scalar bias. The predicted stiffness is then recovered as

$$\hat{k}(x) = \exp\big(\hat{\ell}(x)\big). \tag{15}$$

Predicting log-stiffness rather than stiffness directly improves numerical conditioning and is more suitable when stiffness spans multiple orders of magnitude.

c) Pairwise comparison supervision.: We first consider a dataset of pairwise hardness comparisons

$$\mathcal{D}_P = \Big\{\big(x_p^{(1)}, x_p^{(2)}, y_p, c_p\big)\Big\}_{p=1}^{N_P}, \tag{16}$$

where $x_p^{(1)}$ and $x_p^{(2)}$ are the two objects in comparison p, $y_p \in \{+1, -1\}$ indicates which object is harder, and $c_p \in [0, 1]$ is the annotation confidence. Specifically, $y_p = +1$ means

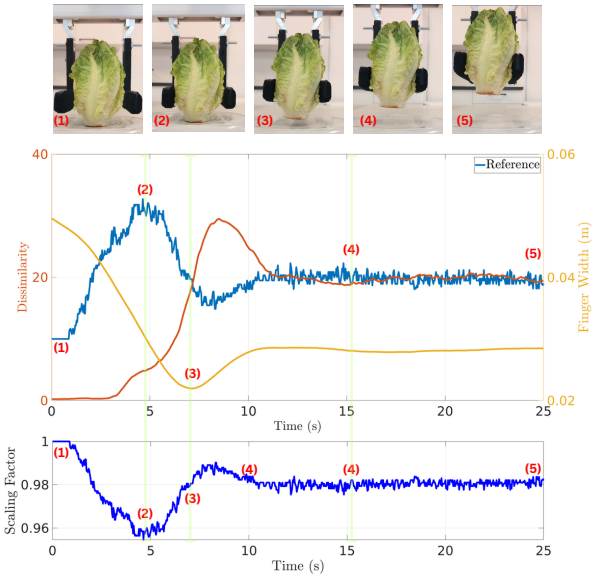

Fig. 4: **Grasping lettuce**: During the robot's upward movement, a decrease in the scaling factor is observed when the lettuce is not grasped. This reduction increases the reference dissimilarity, leading to a decrease in the gripper's finger width. Consequently, the gripper gradually closes until the lettuce is grasped; after grasping, all parameters stabilize. Red numbers and green lines in the figures indicate specific instances in the experiments.

$x_p^{(1)}$ is harder than $x_p^{(2)}$, and $y_p = -1$ means the opposite. We generate this dataset by prompting LLM (Gemma 3) to compare pairs of daily life objects and provide confidence scores for its comparisons. For example, "*Which is harder, a banana or a metal bolt*?" and ask it to provide a confidence score for its answer. This approach allows us to leverage the LLM's extensive world knowledge to generate a rich set of pairwise comparisons without requiring manual annotation. For each pair, the stiffness network predicts

$$\hat{\ell}_p^{(1)} = \hat{\ell}\Big(x_p^{(1)}\Big), \qquad \hat{\ell}_p^{(2)} = \hat{\ell}\Big(x_p^{(2)}\Big), \tag{17}$$

and the corresponding predicted log-stiffness difference is

$$\Delta\hat{\ell}_p = \hat{\ell}_p^{(1)} - \hat{\ell}_p^{(2)}. \tag{18}$$

If $y_p = +1$, the desired outcome is $\Delta\hat{\ell}_p > 0$; if $y_p = -1$, the desired outcome is $\Delta\hat{\ell}_p < 0$. Thus, pairwise supervision teaches the network the *relative ordering* of objects along the stiffness axis.

d) Auxiliary margin network.: To allow different comparison pairs to have different separation requirements, we introduce an auxiliary margin network h_ϕ. This network is used only during training; it is not needed at inference time. For pair p, we define an auxiliary GT-derived feature

$$\Delta\ell_p^\star = \begin{cases} \left|\log k^\star\Big(x_p^{(1)}\Big) - \log k^\star\Big(x_p^{(2)}\Big)\right|, & \text{if ground-truth,} \\ 0, & \text{otherwise,} \end{cases} \tag{19}$$

where $k^\star(\cdot)$ denotes measured stiffness. The margin network receives the concatenated feature vector

$$\mathbf{u}_p = \left[\mathbf{z}\Big(x_p^{(1)}\Big); \mathbf{z}\Big(x_p^{(2)}\Big); \Delta\ell_p^\star\right] \in \mathbb{R}^{2d+1}. \tag{20}$$

and predicts a pair-specific margin score

$$\hat{m}_p = h_\phi(\mathbf{u}_p), \tag{21}$$

where h_ϕ is neural network. This design allows the model to adapt the ranking margin to the semantic content of the pair and, when available, to the magnitude of known stiffness differences. The pairwise supervision is enforced through a hinge loss in log-space:

$$\mathcal{L}_{\text{rank}} = \frac{1}{|\mathcal{B}_P|} \sum_{p \in \mathcal{B}_P} \beta_p \left[\hat{m}_p - y_p \Delta\hat{\ell}_p\right]_+, \tag{22}$$

where $[a]_+ = \max(a, 0)$, $\mathcal{B}_P$ is a minibatch of pairwise samples. This loss is zero whenever the predicted ordering is correct and the signed log-stiffness difference exceeds the required margin. Otherwise, it pushes the stiffness network to move the harder object upward and the softer object downward on the learned log-stiffness axis.

e) Ground-truth regression supervision.: Pairwise supervision alone is insufficient to recover an absolute stiffness scale, since it only constrains relative ordering. To anchor the predictions numerically, we use a second dataset of sparse ground-truth stiffness values

$$\mathcal{D}_G = \left\{\left(x_q, k_q^\star\right)\right\}_{q=1}^{N_G}, \tag{23}$$

where $k_q^\star > 0$ is the measured stiffness of object x_q. The corresponding log-space target is

$$\ell_q^\star = \log(k_q^\star + \varepsilon). \tag{24}$$

We define the regression loss as

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{B}_G|} \sum_{q \in \mathcal{B}_G} \text{Huber}\big(\hat{\ell}(x_q), \ell_q^\star\big), \tag{25}$$

where $\mathcal{B}_G$ is a minibatch of GT objects. This term directly pulls the predicted log-stiffness toward measured values and therefore determines the absolute scale of the learned stiffness axis.

f) Joint training of the stiffness network.: The stiffness network f_θ is trained jointly by the pairwise ranking loss and the GT regression loss:

$$\mathcal{L} = \lambda_{\text{rank}}\,\mathcal{L}_{\text{rank}} + \lambda_{\text{reg}}\,\mathcal{L}_{\text{reg}}. \tag{26}$$

The critical point is that the stiffness network receives gradients from *both* terms. The ranking loss teaches *relative order*, while the regression loss teaches *absolute scale*. Consequently, the learned predictor does not merely separate hard from soft objects; it also places them at physically meaningful locations in log-stiffness space.

In contrast, the auxiliary margin network h_ϕ is updated only through $\mathcal{L}_{\text{rank}}$. Its purpose is to shape the ranking constraint during training, whereas the final stiffness prediction is entirely produced by f_θ.

F. Grasp Control

StiffNET provides an object-specific prior on mechanical behavior directly from language, enabling the controller to select an appropriate grasping strategy before contact and without requiring tactile sensing or explicit material models. Specifically, the stiffness estimate $\hat{k}$ produced by StiffNET is compared against a threshold k_{th}: objects with $\hat{k} < k_{\text{th}}$ are treated as deformable/compliant, whereas objects with $\hat{k} \geq k_{\text{th}}$ are treated as rigid. Inspired by the way humans use semantic priors during manipulation, we then employ a control strategy that adapts the gripper width using visual feedback. For deformable objects, the dissimilarity of the tracked points is used as a proxy for interaction-induced deformation and, consequently, the applied grasp force. In parallel, the scaling factor captures changes in the relative

TABLE II: Comparison of Young's modulus estimates (GPa) produced by GPT-5.3 and **StiffNET**.

Object	**True (GPa)**	**GPT-5.3**	**StiffNET Prediction**
Tofu	0.0001	0.0001	0.0001002
Wooden Block	10	10	9.092
Steel	200	200	210.3
PVC (Polyvinyl Chloride)	2.5	2.5	2.568
Cucumber	-	0.002	0.03191
Walnut	-	6	18.18
Scissors	-	200	157
Power Bank	-	3	5.451
Plastic Bottle	-	1.5	2.1

distance between the object and the camera, which is particularly informative for rigid-object grasping.

The grasp controller is given by

$$\mathcal{W}_{t+1} = \mathcal{W}_t - \lambda(D_{ref_t} - D_t), \tag{27}$$

where $\mathcal{W}_t$ is commanded gripper width at time t, $\lambda \in \mathbb{R}_+$ is the adaptation gain, and

$$D_{ref_t} = D_{min} + \Omega(1 - s_{f_t}) \tag{28}$$

is the reference dissimilarity. Here, D_{min} represents the constant minimum dissimilarity set by the user or also can be estimated through StiffNET, $\Omega \in \mathbb{R}_+$ is the scaling factor gain, and s_{f_t} is the scaling factor. Initially, the controller produces a desired dissimilarity based on D_{min}. If the object begins to slip during grasping, the scaling factor s_{f_t} decreases, causing D_{ref} to increase. The gripper width is then adjusted proportionally until the slipping stops, resulting in a secure and stable grasp of the deformable object. While this control strategy is effective for deformable objects, it is not suitable for rigid objects since dissimilarity remains minimal due to the lack of deformation. For rigid objects, the control relies on the scaling factor and is expressed as

$$\mathcal{W}_t = (s_{f_t} - s_{f_{min}}) \frac{(\mathcal{W}_{max} - \mathcal{W}_{min})}{(1 - s_{f_{min}})} + \mathcal{W}_{min}, \tag{29}$$

where $\mathcal{W}_{max}$ and $\mathcal{W}_{min}$ are the gripper's maximum and minimum widths, and $s_{f_{min}}$ is the minimum scaling factor and a user-defined sensitivity parameter to control the gripper width. This strategy allows for the gripper width to be updated until the scaling factor remains variable, ensuring a secure grasp of rigid objects.

III. EXPERIMENTS

To validate the performance of the proposed framework, we conducted grasping experiments using various objects. These included deformable objects such as lettuce, mozzarella cheese, croissant bread, and paper towels, as well as a rigid object like a hard plastic bottle. For these experiments, we used a Franka Emika Research 3 robotic arm equipped with a Franka Hand, as shown in Fig 2. A RAZER Kiyo-X generic webcam was used for vision input. Additionally, we extended the fingers of the Franka Hand with 3D-printed PLA extensions with foam cushions. The models were deployed on an Nvidia RTX A4000 GPU, and with four grid points ($k = 4$), we were able to run the algorithm at 30 frames per second.

In the first experiment involving lettuce, we passed the prompt *green vegetables* to the object detection algorithm, which subsequently provided the coordinates for object segmentation. We used $k_{\text{th}} = 0.1$ and set the minimum dissimilarity as $D_{min} = 10$, after which the robot began the grasping process. During grasping, the robot exhibited upward motion, causing the scaling factor to decrease, as depicted in Fig. 4. This reduction in the scaling factor led to an increase in the reference dissimilarity, D_{ref_t}, as the scaling factor gain was defined by $\Omega = 500$. The adaptation gain for the finger width was set to $\lambda = 7 \times 10^{-6}$. Other parameters are as follows $\Gamma = 10^4$ and $c = 5$. The

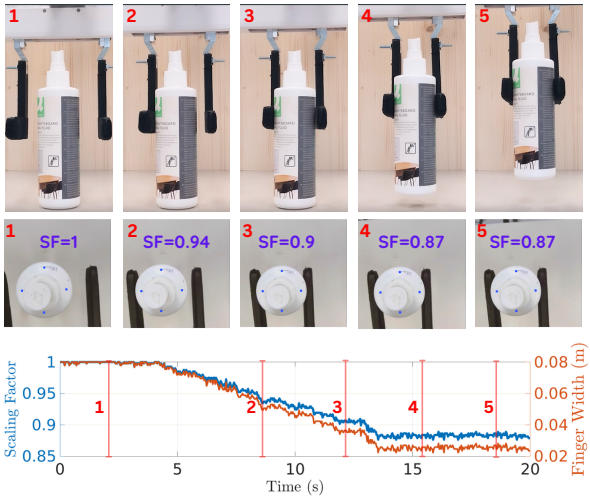

Fig. 5: **Grasping bottle**: As the robot ascends, the distance between the camera and the bottle's tracking points increases, reducing the scaling factor. This leads the gripper controller to command narrower finger widths. Once the bottle is grasped, both the scaling factor and finger width stabilize. The top row of images illustrates the robot's upward motion during the bottle-grasping process, while the bottom row shows the scaling factor (SF) and finger width. Red markings indicate instances in the experiment.

scaling factor converges as the robot achieves a stable grasp, allowing it to successfully execute the pick-up operation. It is important to highlight that the finger width presented in the results corresponds to the commanded finger width. Due to the physical structure of the Franka Hand, when the commanded finger width is set to zero, the actual physical finger width remains at 0.02 meters. It is important to note that the measured dissimilarity can vary across trials and across objects (e.g., the lettuce trials in Fig. 4 versus the multi-object results in Fig. 3) due to variation in object properties. Therefore, we use the estimated stiffness only for control-mode selection.

In our grasping experiments, the desired dissimilarity for any object is not known a priori. However, by conducting an initial grasping trial with upward movement, we can observe at what point the dissimilarity stabilizes. We then use this stabilized value as the minimum dissimilarity for subsequent grasping attempts. It is important to note that this approach does not account for dynamic cases, such as scenarios involving high acceleration, which could lead to slippage during grasping. Therefore, to ensure robustness, a safety factor should be introduced when determining the minimum dissimilarity. This safety factor is multiplied with D_{min} in (28), which compensates for higher accelerations that may occur during manipulation. For example, during our experiments with lettuce, we applied a safety factor of 7.5. Using this method, we successfully demonstrate pick-and-place manipulation across multiple objects, as shown in Fig. 3. In each trial, the system first adjusts to the desired dissimilarity before performing a successful manipulation. The finger width adapts accordingly, ensuring desirable tracking performance and stable grasps.

We further demonstrated grasp manipulation with a rigid object, selecting a hard plastic bottle for the experiment. For the rigid object manipulation, we chose the following parameters: minimum scaling factor $s_{f_{\min}} = 0.85$, maximum gripper width $\mathcal{W}_{\max} = 0.08\,\mathrm{m}$, and minimum gripper width $\mathcal{W}_{\min} = 0.01\,\mathrm{m}$. The experiment begins with the gripper fully open, after which the robot moves upward. This movement causes a reduction in the scaling factor, s_{f_t}, resulting in a decrease in finger width until the gripper successfully grasps the bottle. Once the bottle is grasped, the relative distance between the bottle and the camera remains fixed, leading to the stabilization of both the scaling factor and the finger width, as shown in Fig. 5.

IV. CONCLUSION

In conclusion, we present a novel framework for manipulating deformable and rigid objects using only an RGB camera, eliminating the need for complex sensors, mechanical models, or specialized grippers. Our control strategy adapts the gripper width based on a dissimilarity measure and a scaling factor of tracking points on the object. The dissimilarity correlates with the applied force on deformable objects, while the scaling factor correlates with the object's distance from the camera. Tested on the Franka Emika Research 3 robotic arm and gripper, our system effectively handles various objects, from soft items like lettuce and mozzarella cheese to rigid plastic bottles. The framework's success lies in integrating real-time object detection, segmentation, point tracking, and a grasp controller that adapts to varying levels of object compliance.

REFERENCES

[1] B. Zitkovich and et al., "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds. PMLR, 06–09 Nov 2023.

[2] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, no. 7, 2018.

[3] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, no. 3, 1988.

[4] Y. Han, K. Yu, R. Batra, N. Boyd, C. Mehta, T. Zhao, Y. She, S. Hutchinson, and Y. Zhao, "Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer," *IEEE/ASME Transactions on Mechatronics*, 2024.

[5] S. Dong, D. Ma, E. Donlon, and A. Rodriguez, "Maintaining grasps within slipping bounds by monitoring incipient slip," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[6] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[7] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, no. 12, 2017.

[8] Y. Du, G. Zhang, Y. Zhang, and M. Y. Wang, "High-resolution 3-dimensional contact deformation tracking for fingervision sensor with dense random color pattern," *IEEE Robotics and Automation Letters*, no. 2, 2021.

[9] A. Yamaguchi and C. G. Atkeson, "Implementing tactile behaviors using fingervision," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017.

[10] W. Yuan, M. A. Srinivasan, and E. H. Adelson, "Estimating object hardness with a gelsight touch sensor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 208–215.

[11] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer *et al.*, "Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robotics and Automation Letters*, no. 3, 2020.

[12] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, and C. Duriez, "Fem-based deformation control for dexterous manipulation of 3d soft objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[13] M. O. Fonkoua, F. Chaumette, and A. Krupa, "Deformation control of a 3d soft object using rgb-d visual servoing and fem-based dynamic model," *IEEE Robotics and Automation Letters*, no. 8, 2024.

[14] W. Xie, M. Valentini, J. Lavering, and N. Correll, "Deligrasp: Inferring object properties with llms for adaptive grasp policies," in *Proceedings of The 8th Conference on Robot Learning*. PMLR, 2024.

[15] L. He, Q. Lu, S.-A. Abad, N. Rojas, and T. Nanayakkara, "Soft fingertips with tactile sensing and active deformation for robust grasping of delicate objects," *IEEE Robotics and Automation letters*, no. 2, 2020.

[16] Z. Wang, D. S. Chathuranga, and S. Hirai, "3d printed soft gripper for automatic lunch box packing," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2016.

[17] W. Wang and S.-H. Ahn, "Shape memory alloy-based soft gripper with variable stiffness for compliant and effective grasping," *Soft robotics*, no. 4, 2017.

[18] J. Bonet and R. D. Wood, *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[19] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, no. 54, 2021.

[20] Z. Goranova, M. Baeva, S. Stankov, and G. Zsivanovits, "Sensory characteristics and textural changes during storage of sponge cake with functional ingredients," *Journal of Food physics*, 2015.

[21] H. X. Trinh, H.-H. Nguyen, T.-D. Pham, and C. A. My, "A novel rigid-soft gripper for safe and reliable object handling," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, no. 4, 2024.

[22] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Reviews Neuroscience*, no. 5, 2009.

[23] I. Camponogara and R. Volcic, "Grasping movements toward seen and handheld objects," *Scientific reports*, no. 1, 2019.

[24] A. R. Sobinov and S. J. Bensmaia, "The neural mechanisms of manual dexterity," *Nature Reviews Neuroscience*, no. 12, 2021.

[25] A.-S. Augurelle, A. M. Smith, T. Lejeune, and J.-L. Thonnard, "Importance of cutaneous feedback in maintaining a secure grip during manipulation of hand-held objects," *Journal of neurophysiology*, no. 2, 2003.

[26] D. A. Nowak and J. Hermsdörfer, "Selective deficits of grip force control during object manipulation in patients with reduced sensibility of the grasping digits," *Neuroscience research*, no. 1, 2003.

[27] M. Cavdan, N. Goktepe, K. Drewing, and K. Doerschner, "Assessing the representational structure of softness activated by words," *Scientific Reports*, no. 1, 2023.

[28] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[29] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman, "Tapir: Tracking any point with per-frame initialization and temporal refinement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[30] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker: It is better to track together," *arXiv preprint arXiv:2307.07635*, 2023.

[31] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *arXiv:2406.09414*, 2024.

[32] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," *arXiv preprint arXiv:2401.17270*, 2024.