Vision-Guided Agricultural Tasks With Diffusion-Based Motion Learning Using a Human–Robot Collaborative System

Hung Hon Cheng, *Member, IEEE*, Cheng Pan, *Graduate Student Member, IEEE*, Alessio Desogus, Qinghua Guan, *Member, IEEE*, Louis Sutter, and Josie Hughes, *Member, IEEE*

***Abstract*—We present a vision-guided agricultural automation framework with human-in-the-loop learning for harvesting, inspection, and occlusion removal. The system integrates lightweight 7-DoF manipulators with interchangeable soft grippers and dual-camera perception, mounted on cable/rail infrastructures for large-area operation. Three motion-generation strategies are implemented: YOLO-based visual servoing, diffusion-based policy, and a combined YOLO–diffusion policy. A teleoperation interface supports task-space demonstration, data correction, and motion intervention during deployment. The system is validated on raspberry harvesting and further demonstrated on bimanual coordination and human–robot collaboration tasks. The resulting framework integrates perception, human-in-the-loop control, and learned motion for large-scale agricultural manipulation.**

***Note to Practitioners*—Many agricultural robots are designed for a single task and rely on heavy platforms, expensive industrial arms, or task-specific control pipelines. In practice, farms also face changing crops, occlusions, uneven terrain, and situations that require human intervention. This paper presents a robotic framework that combines lightweight rail- or cable-mounted robotic arms, human-in-the-loop operation, and vision-guided motion learning. Operators can demonstrate and correct robot motions in real time, while learned policies handle repetitive or visually complex actions. The dual-arm teleoperation interface supports bimanual control and real-time collection of human motion data during operation. Different vision and control methods, such as object detection, learned motion policies, or their combination, can be selected based on the task. The system uses low-cost arms and modular hardware with minimal setup requirements. Demonstrations collected during operation can be used for policy training without expert programming or large datasets. The system was tested on harvesting, inspection, occlusion removal, and human–robot collaboration tasks. Current limitations include camera-based perception and the need for human intervention in unfamiliar situations. Additional sensors and improved perception methods can be integrated into the same framework. This allows the level of automation to increase gradually while retaining human supervision when needed.**

Received 15 August 2025; revised 19 December 2025, 4 May 2026, and 14 August 2026; accepted 17 October 2026. Date of publication 2 November 2026; date of current version 10 November 2026. This article was recommended for publication by Associate Editor J. Liang and Editor Z. Li upon evaluation of the reviewers' comments. This work was supported in part by BRIDGE, Swiss National Science Foundation (SNSF), under Grant 40B1-0_242617/1, in part by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie under Grant 945363, and in part by SNSF under Grant 1000157. *(Hung Hon Cheng and Cheng Pan are co-first authors.) (Corresponding author: Hung Hon Cheng.)*

Hung Hon Cheng, Cheng Pan, Alessio Desogus, Qinghua Guan, and Josie Hughes are with the CREATE Laboratory, School of Engineering (STI), EPFL, 1015 Lausanne, Switzerland (e-mail: chenghunghon@link.cuhk.edu.hk).

Louis Sutter is with Agroscope, 1964 Conthey, Switzerland.

This article has supplementary downloadable material available at https://doi.org/10.1109/TASE.2026.3749033, provided by the authors.

Digital Object Identifier 10.1109/TASE.2026.3749033

***Index Terms*—Agricultural robots, human-robot interaction, imitation learning, telerobotics.**

I. Introduction

AGRICULTURAL manipulation tasks such as planting, harvesting [1], [2], [3], [4], pruning [5], and inspection [6] require dexterous robot behavior in diverse environments including orchards [7], greenhouses, and vertical farms [8]. Most existing systems use commercial manipulators mounted on mobile bases with vision modules [2], [3], [4], [9] and typically focus on a single specific task [10], [11]. Large-scale deployments remain limited, not only because of environmental factors such as uneven terrain, occlusion, inconsistent lighting, and sensor limitations, but also because of system-level constraints including high costs, limited platform generality that impedes scalability, and the lack of a unified framework for adapting manipulation strategies to new tasks or leveraging human-in-the-loop collaboration, as explored in recent works [12], [13], [14], [15], [16].

These studies also highlight three major challenges: first, developing a multi-task robotic platform and framework capable of large-scale deployment for multiple tasks. Second, enabling flexible integration of different vision-based models for motion planning and generation. Third, establishing a teleoperation system that enables efficient collection of human demonstrations and real-time user intervention.

The first challenge can be addressed using high-degree-of-freedom (DoF) robotic arms equipped with exchangeable end-effectors, mobility, and sensing. Standard 6-DoF arms such as UR and Franka are designed to handle heavier objects rather than lightweight fruits, resulting in unnecessary weight, cost, and power demand. Currently, most researchers concentrate on end-effector design, with limited consideration given to practical engineering challenges for real-world applications, such as system architecture, multi-arm collaboration and cost per unit.

To address the second challenge of vision-based manipulation in agriculture, an emerging direction in agricultural robotics focuses on multi-level perception and motion generation. Recent studies have proposed vision-based and teleoperation systems that enable robots to collaborate with human workers in fruit-picking operations and support field logistics [17], [18]. In [7], the researchers used intervention-based imitation learning to train a UAV to mimic a human pilot in orchards. Similarly, several studies [2], [19], [20], [21], [22] have developed learning frameworks for precise crop picking and cutting of apples, lychees, tomatoes, pumpkins

© 2026 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

and pineapple using different computer vision approaches such as 3D reconstruction. From a motion-planning perspective, vision-based methods remain the most common [23]. Deep learning models such as YOLO [24], [25], [26], and SSD [27], integrated with 3D sensors like LiDAR or depth cameras [28], [29], are widely applied in harvesting and inspection tasks. In addition, few-shot learning, deep learning, and diffusion-based probabilistic approaches are emerging to enhance crop and weed detection [30], [31], [32], [33]. However, motion planning remains a significant challenge. Recent studies have explored imitation learning for agricultural manipulation [1], [13], [15], [16]. For example, bimanual harvesting was learned from 20-minute demonstrations [1], while diffusion policies trained on 300 RGB demonstrations achieved a 28.95% success rate in autonomous pepper harvesting [16].

Beyond perception and learning, the lack of intuitive control remains a critical limitation in current agricultural robotic systems. Most platforms lack intuitive user interfaces and rely on low-dimensional point-to-point controls such as stepwise single-direction arm motions [1], [14], [34]. Such control methods limit users' ability to control, teach, and intervene in robot motion in an intuitive and arm-like manner, making tasks such as clearing leaves or adjusting approach angles particularly challenging. Recent research has begun addressing these challenges through human-in-the-loop frameworks that integrate real-time teleoperation with adaptive autonomy. For instance, systems employing formal logic for safe multi-robot coordination and cooperative learning frameworks for shared human–robot decision-making have demonstrated improved robustness and flexibility in agricultural environments [12], [14], [35]. Teleoperation is another key component of practical human–robot collaboration, enabling real-time correction, human-guided adaptation, and data-efficient learning. In particular, imitation learning from teleoperated demonstrations allows robots to learn task-specific skills directly from human operators and adapt them to unstructured field environments, eliminating the need for detailed environment reconstruction or manual motion programming.

This paper presents a human-in-the-loop robotic system, as shown in Figure 1, that integrates CAFEs (Collaborative Agricultural Floating End-Effectors) [36] with a teleoperation controller [37]. The proposed system architecture enables real-time human operation and demonstration data collection for training across diverse agricultural tasks in row-based agricultural setups. The system consists of multiple 7-DoF vision-guided arms with interchangeable grippers that can be easily mounted on low-cost, long infrastructures such as cables or rails. This design eliminates the need for a heavy and expensive mobile base on uneven terrain by using a shared and expandable structure, allowing the system to operate efficiently above uneven terrain without expending energy to maintain its position. Each arm is equipped with onboard computation and built-in mobility, supporting both policy control and human teleoperation for diverse task scenarios. We validate the system on soft-fruit harvesting and inspection tasks using models trained from limited demonstrations. The same architecture also supports different vision-based control methods.

- A scalable multi-arm agricultural robotic system with interchangeable end-effectors and unified control architecture, enabling deployment across multiple manipulation tasks within the same hardware framework.
- An intuitive 7-Dof teleoperation framework that enables real-time, dual-arm control directly in Cartesian space, allowing users of any skill level to demonstrate and correct manipulation motions for action model generation in agricultural tasks.
- Experimental validation across multiple representative agricultural tasks, including raspberry harvesting, leaf inspection, and occlusion removal, demonstrating the effectiveness of integrating vision-based perception, diffusion-based motion learning, and human-in-the-loop supervision within a unified system.

Based on this system architecture, we conduct task-wise field experiments in the raspberry field and the indoor hydroponic setting. Demonstrations are collected via teleoperation and used to train task-specific models, which are then evaluated under randomized field conditions including daylight, nighttime operation, and varying indoor lighting. Approximately 40 demonstrations were collected per task. In total, more than 50 preliminary harvesting tests, 43 approach trials in the field, 20 leaf-turning trials, 50 follow-up harvesting tests, and 150 occlusion-clearing tests were conducted in an indoor hydroponic system.

This paper is structured as follows. Section II introduces the multi-task robotic system, including the hardware architecture, control framework, and human-in-the-loop operation. Section III presents example implementations of vision-guided autonomous control methods deployed within the proposed framework, covering both YOLO-based visual servoing and diffusion-based motion generation. Section IV describes the experimental setup, and Section V shows results across harvesting, inspection, occlusion clearing, and policy learning. Finally, Section VI concludes with key findings, discusses current limitations, and outlines potential future extensions.

II. Multi-Task Robotic System

This section introduces the framework of the proposed robotic system. To support common agricultural tasks and allow an operator to demonstrate tasks, the proposed system consists of three major elements: (a) hardware components such as the robotic arm and an intuitive controller; (b) a software framework for algorithms that leverage recent techniques in object detection and motion generation; (c) human-in-the-loop teleoperation that can control the robot in real time using hand and arm manipulation for data collection rather than a traditional motion planner.

A. Hardware Component

1) Robotic Arm Design: Building upon the concept introduced in [36], where the CAFEs system comprises multiple modular, linear-traversing robotic arms, this work further develops the 7-DoF robot [38] with a customized gripper and

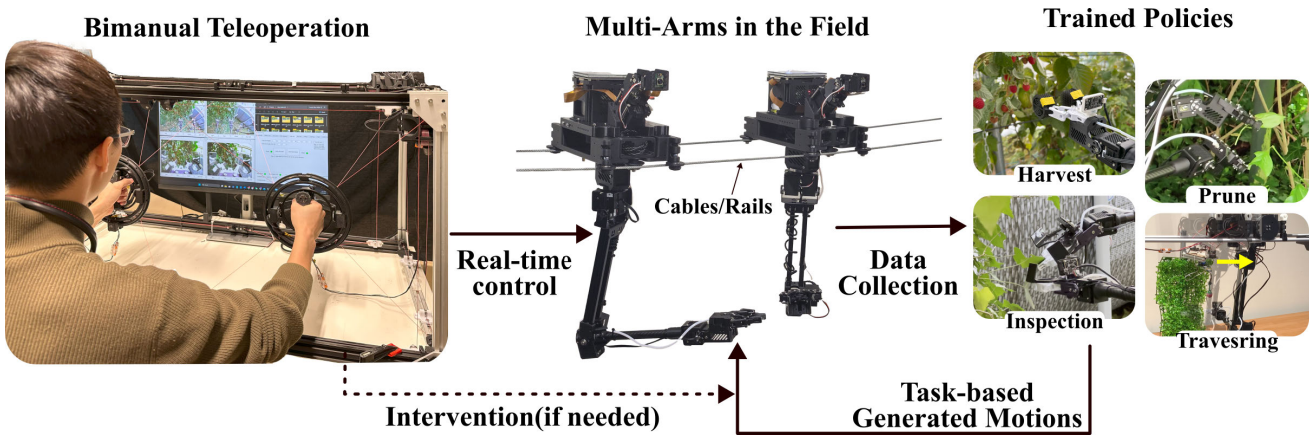


Fig. 1. Overview of the proposed system: the leader controller sends task space commands to the robot, while all data are collected for training task-specific policy models. These models can be retrained as needed, with real-time human intervention used to correct or guide generated motions.

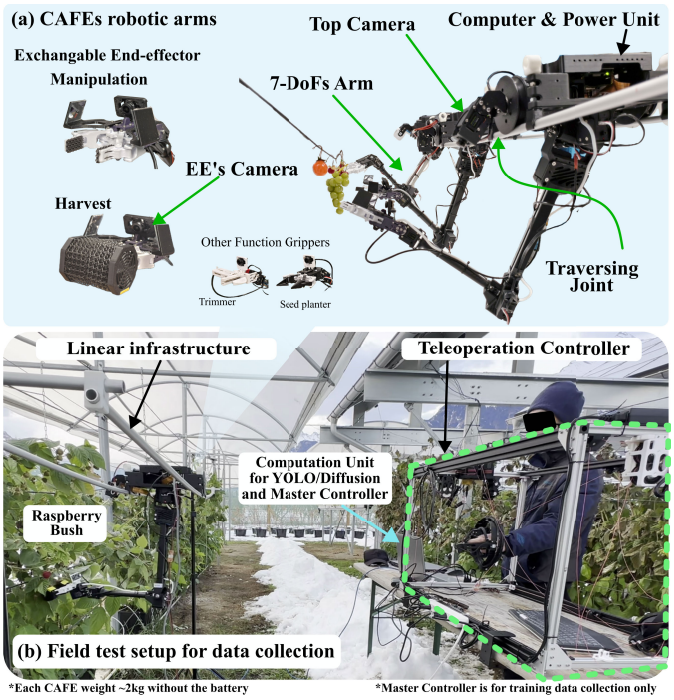


Fig. 2. (a) Hardware design showing the multi-arm structure, exchangeable end-effectors, and onboard computing units. (b) Field test setup for raspberry harvesting, demonstrating the system in a real-world agricultural environment.

control system. As shown in Figure 2(a), each CAFEs unit is a lightweight manipulator deployed on fixed overhead cables or rails. This design supports efficient coverage of large crop fields while minimizing ground interference and infrastructure complexity. The linear mobility along the cable/rail offers low energy consumption and high manipulability compared to UAV platforms, especially in row-structured environments.

Each CAFEs unit integrates 12 servo motors, a mobile base, a Raspberry Pi 5, a servo control board, a display monitor, a cooling fan, two cameras, and several 3D-printed structural components. The total weight without the battery is about 2 kg. The arm supports a 0.5-1.5 kg payload (depending on the motor version). The previous design is described in [38] for reference, and the arm design in this work follows the same

TABLE I
ARM DH-TABLE AND PARAMETERS

Joint i	θ_i	α_i	a_i	d_i (mm)
1	$-90°$	$90°$	0	0
2	$90°$	$-90°$	0	0
3	$90°$	$90°$	0	-325.7
4	0	$90°$	0	0
5	0	$90°$	0	-236.52
6	$90°$	0	-102.72	0

kinematic design but with greater emphasis on the gripper, mobile base, and low-level control refinement.

The robotic arm uses a humanoid configuration to simplify motion mapping from the operator, and its DH parameters are listed in Table I. The closed-form inverse kinematics algorithm follows the method presented in [39], with an additional mobile joint enabling traversal along the linear infrastructure. The linear traversal joint (with encoder) is used only for coarse base alignment and is locked during manipulation. Therefore, the inverse kinematics is solved relative to the local mobile-base coordinate frame, and the arm alone provides sufficient 6-DoF pose control. This removes kinematic redundancy and simplifies teleoperation and data collection by maintaining a consistent camera viewpoint and reducing training-data complexity. The inverse kinematics computation completes within milliseconds, while the ROS2 node regulates the Python solver frequency at 100 Hz, accounting for other parallel image processing and servo control computations.

2) Gripper Design: As shown in Figure 2(a), the end-effector is designed to be exchangeable, with two interchangeable gripper options for this work. The dimensions of the exchangeable part for both designs are presented in Figure 3(a). The first design is intended for more precise dexterous manipulation, such as grasping stems, and therefore uses a smaller and compliant contact area. The second design provides a larger contact area with a softer interface to better accommodate delicate fruits.

Both grippers use a conductive lattice structure with tunable stiffness through design parameters. Details of the gripper compliance and passive morphing behavior are provided in

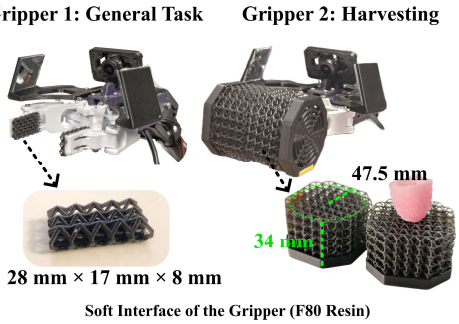


Fig. 3. Gripper Design: Lattice design for soft and deformable contact.

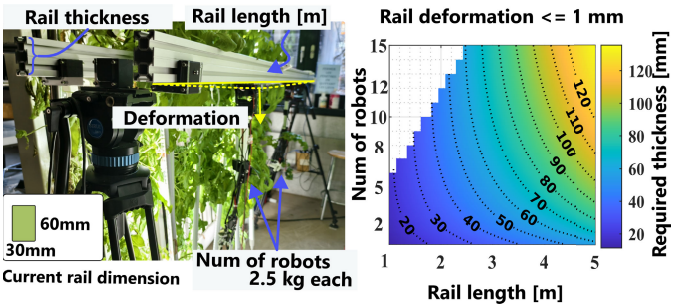


Fig. 4. Required rail thickness as a function of rail length L and number of robots to maintain a maximum beam deflection of $\sigma \leq 1mm$. The contour lines indicate regions of constant required thickness. Increasing the number of robots or the rail length results in a higher required rail thickness to satisfy the deflection constraint.

[40]. In addition, the conductive structure can also function as a tactile sensor when connected to sensing circuitry [41] but it is not yet implemented in this work. The soft gripper (Gripper 2) is specifically designed to compensate for position tracking errors in the control algorithm and robot movement. Its dimensions were finalized based on the target raspberry size range of 22–28 mm in diameter, with the corresponding size estimation described in Section III-A For raspberries significantly outside this range, the gripper dimensions would need to be adjusted accordingly. The main design advantages of the gripper are summarized as follows:

- Passive deformation to adapt the target's shape: The gripper adapts to the object's geometry. As a result, a secure and gentle grasp can be achieved rather than a pinch grip.
- Mechanical compensation for vision tracking errors: Its large and compliant structure tolerates positional uncertainty without requiring highly precise positioning of the target within the gripper.
- Built-in passive force regulation: The compliant structure intrinsically limits the gripping force/pressure acting on the target fruit, eliminating the need for force sensors and reducing system cost. This also reduces the risk of fruit damage.

Experimental validation is presented in the results section.

3) Camera Design: Two cameras are mounted on the system: one on top of the robots base and the other on the wrist of the robotic arm. They can stream images at a maximum rate of 30 Hz because of hardware limitations. This setup is chosen to provide wide visual coverage and to facilitate more effective target alignment and control. The RGB cameras can be replaced with depth cameras if additional sensing capability is required. The wrist-mounted (gripper) camera serves two purposes: (1) providing a close-range view for the policy or other vision-based models and (2) ensuring that the object remains within the grasping area.

An optional mirror can be mounted above the gripper. Its purpose is to provide implicit stereo and additional pixel cues for the user. Since the video streaming is only 2D, the mirror helps the operator reduce uncertain motions during data collection when using the smaller gripper (Gripper 1) for fine movement. Otherwise, unnecessary corrective motions by the operator could be incorporated into the demonstration data. Gripper 2 tolerates larger positioning errors for grasping the object and the mirror may not be necessary. This idea was introduced in [42] and has been used for policy training data collection. Instead of adding extra cameras, including depth cameras, the mirror offers a simple, lightweight, and low-cost method to indicate that the object is within the gripping region, as shown in Figure 3(b).

The top camera, positioned and centered above the mobile base, serves two key functions: (1) observing the arms motion for policy model control, and (2) enabling alignment between the arm and object. As described later, a simple PD controller adjusts the pan-tilt unit such that, when the pan angle approaches zero, the robotic arm is effectively aligned with the target. This significantly reduces the complexity of approach motions. In particular, for policy model training, it allows users to provide fewer demonstrations from a single initial pose, thereby reducing the amount of data required to cover various arm configurations.

4) Scalability Analysis of the Supporting Structure: Deformation and oscillation in thin cables or rails can affect the arm's position and motion. To mitigate these effects, we analyze the structural parameters based on the classical Euler-Bernoulli beam theory. The maximum vertical deflection per supporting rail is given by

$$\delta_{\max} = C\,\frac{k\,m\,g\,L^3}{E\,I_{\text{rect}}}, \quad I_{\text{rect}} = \frac{b\,h^3}{12}, \tag{1}$$

where k is the number of robots acting on the span, m is the mass of a single robot, g is the gravity term, L is the rail length, E is the Young's modulus of the rail material, and I is the second moment of area of the cross-section, while b and h denote the rail width and height, respectively. The constant C depends on the boundary condition and load distribution ($C = 1/96$ for a simply supported beam and $C = 1/384$ for a fixed–fixed beam). In this analysis, two parallel aluminum rails are assumed to share the total load equally, and the deflection is evaluated for different rail geometries and robot configurations.

For example, based on Figure 4, maintaining a deformation below 1 mm over a 4 m span with two robots requires a minimum rail thickness of 55–60 mm. This stiffness reduces arm oscillation during motion and improves overall system stability. This structural configuration was not used in the

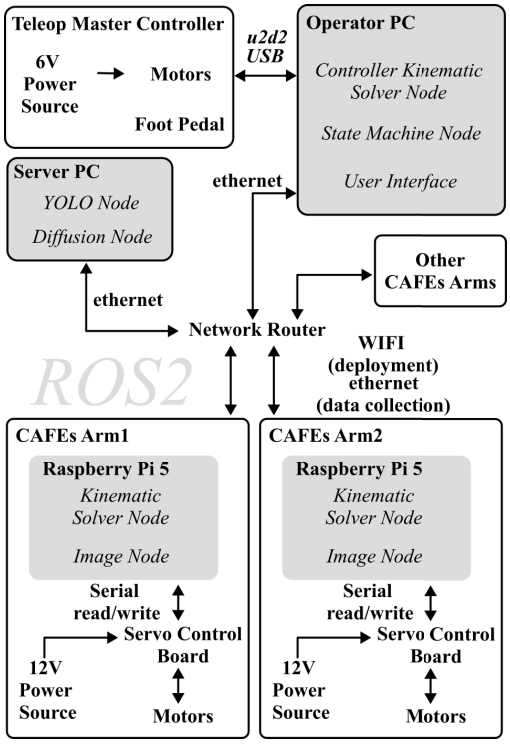


Fig. 5. Hardware connection and the major ROS2 node in the system.

initial field tests but was introduced during later stages of system development. Other materials, such as steel, can achieve the same stiffness with even smaller thickness.

5) Teleoperation Device: The teleoperation controller consists of two 8-DoF cable-driven parallel robots running at 200 Hz. The detailed design and algorithm are described in [37]. It continuously computes task-space commands, including Cartesian position (x, y, z), orientation, redundant-joint commands, and gripper commands based on the operator's hand movements, and provides real-time control and haptic feedback for both the gripper and mobile joints. This setup allows the operator to intuitively move their arms to generate natural, spatially continuous trajectories, which are directly mapped to the teleoperated CAFE units.

The system supports simultaneous control of one or two CAFE arms, enabling flexible operation in multi-robot settings. By abstracting low-level joint commands and bypassing conventional motion planning, the leader controller enables responsive manipulation without robotics expertise. Field workers can perform complex tasks with minimal training. Other haptic teleoperation devices can provide similar functionality. However, their limited degrees of freedom might restrict control of additional DoFs, such as the linear traversing joint. In contrast, our controller is explicitly designed to operate the CAFE arms.

B. Software & Control Component

1) Control System Architecture: The overall software framework connecting the hardware components is shown in Figure 5. The system communicates through ROS2. The main data exchanged between nodes are camera images and

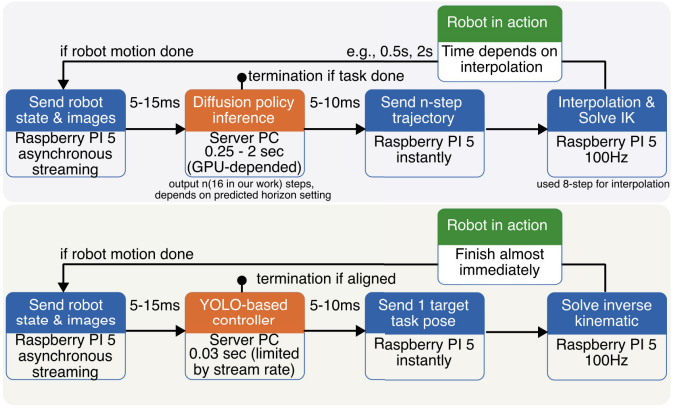


Fig. 6. Two different control loops with timing breakdown and processing units.

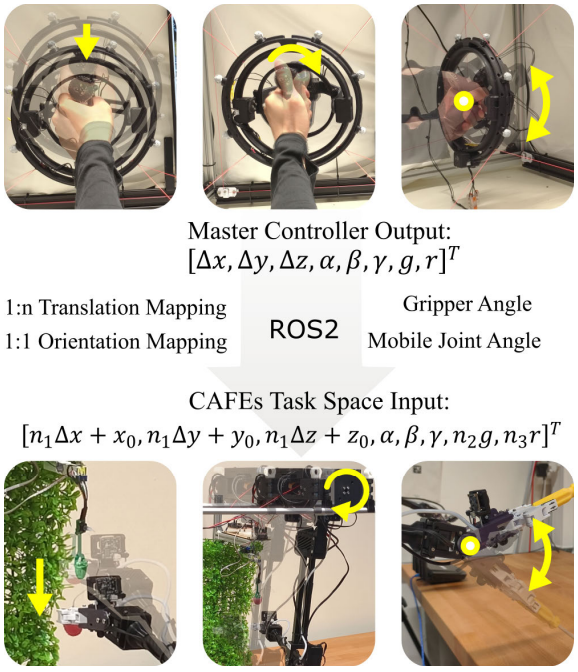


Fig. 7. Teleoperation Control between the leader and the CAFEs.

Cartesian poses. Each node handles local tasks such as inverse kinematics, image processing, and motor control. Two control methods are used: YOLO-based visual servoing, included as a baseline to show compatibility with traditional approaches, and diffusion-based imitation learning for motion generation. Their timing breakdowns are illustrated in Figure 6. Communication latency depends primarily on the network router. For the diffusion policy, the prediction horizon and output motion duration are determined by the sampling rate and prediction length [43]. The execution speed can be adjusted at deployment via interpolation, and the achievable smoothness depends on the robot hardware and controller.

2) Teleoperation Command: As shown in Figure 7, control inputs from the leader teleoperation device are transmitted to the CAFE's computer unit, with the teleoperated pose $\mathbf{X}_s$ defined as:

$$\mathbf{X}_s = [\mathbf{x}_0 + n_1[\Delta x, \Delta y, \Delta z]^T]^T, \alpha, \beta, \gamma, n_2 g, n_3 r]^T \in \mathbb{R}^8 \quad (2)$$

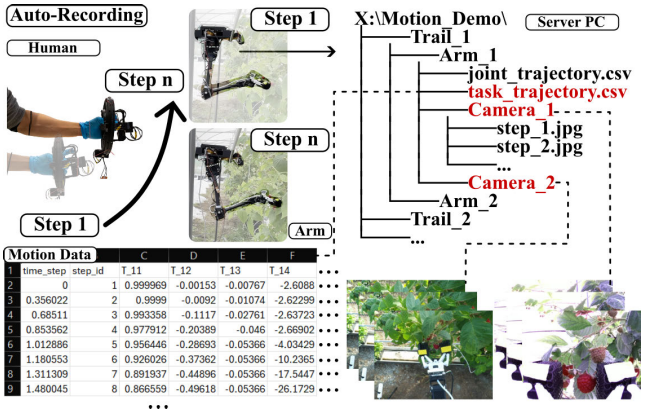


Fig. 8. Dataset preparation during operator demo.

Here, $\mathbf{x}_0 = [x_0, y_0, z_0]^T$ represents the initial position of the robotic arm. The terms $\Delta x, \Delta y, \Delta z, \alpha, \beta, \gamma, g$, and r correspond to the positional changes, orientation angles, and the gripper and rotator angles generated by the leader controller. These inputs are scaled by user-defined mapping ratios n_1, n_2, and n_3.

The system also allows the user to intervene in the robot's motion at any time during deployment of the trained policy. For example,

$$\mathbf{X}_s = (1-u)\mathbf{x}_D + uv\mathbf{x}_m \tag{3}$$

where $\mathbf{x}_D \in \mathbb{R}^8$ and $\mathbf{x}_m \in \mathbb{R}^8$ are task space commands generated by the diffusion model and the leader controller, respectively. The parameter $0 \le u \le 1$ is the supervision ratio that determines the relative contribution of autonomous and human control. For example, when $u = 1$, the operator has full control. The parameter $0 \le v \le 1$ is a decaying variable that ensures smooth motion transitions between autonomous control and operator intervention.

3) Data Collection Method: In this work, three types of data are collected during teleoperation: (1) joint angle trajectories, (2) task-space trajectories, and (3) images from all cameras. The data are organized in a structured folder, with images saved in JPEG format and trajectories stored as CSV files for subsequent downsampling and training, as shown in Figure 8. The same structure extends to any additional data (in CSV) collected in future tasks. Currently, the data are captured at 30 Hz (maximum sampling rate is 100Hz) to limit storage requirements. It is worth noting that when the operator activates the operator controller, all data are recorded automatically.

This embedded data collection approach enables automatic dataset generation for both perception and control. The captured data are used to train YOLO and diffusion policy models, eliminating the need for separate image collection and manual labeling. Because data is gathered from the robot's viewpoint during real tasks, the resulting models benefit from consistent visual alignment between training and deployment, improving generalization and detection accuracy compared to lab-based datasets.

C. Human Role in Proposed Framework

Human involvement is an important component of the proposed framework. As shown in Figure 1, the human roles can be summarized as follows:

- Motion data collection (Section II-B3): the operator uses the teleoperation device to demonstrate motion data for different tasks.
- Intervention actions: the operator can control any arm for motion correction [44] or collaborative manipulation with other automated arm(s).
- High-level control: the operator monitors and assigns tasks or actions to different robotic arms via the user interface.

In addition to intervention, the policy model outputs a normalized task progress value between 0 and 1, with smooth, continuous motion during normal execution and stalled or discontinuous behavior indicated by a prolonged low value. We focused on enabling intuitive control of the target arm without requiring programming expertise from the operators. Unlike existing systems that rely on discrete keyboard or joystick commands, our system maps the operators hand motion directly to the robots spatial motion and gripper state. This reduces operation complexity and provides simple, natural control, allowing smooth high-DoF trajectories for training.

III. Vision Guided Autonomous Control Methods

This section introduces the three motion generation methods used in our framework: visual servoing with YOLO, motion generation using the diffusion policy, and the combined YOLO–diffusion approach. Both YOLO and diffusion-policy implementations are open-source and widely adopted, but in this work they serve as modules for (1) detecting the target fruit and (2) generating motions for tasks that cannot be easily programmed. In our framework, the three methods form a hierarchical structure: visual servoing with YOLO provides the baseline capability for target tracking, diffusion policy enables higher-level motion generation, and the combined YOLO–diffusion approach integrates both sensing and motion synthesis for tasks requiring both explicit target cues and learned motion generation. A summary of the data preparation for both models is provided in Table II. Training scripts and configuration settings are provided in the supplementary materials.

A. Visual Servoing Using YOLO

We used the *YOLOv11n* models for both the top and wrist cameras on all robots. Each camera performs object detection at 30 Hz. A standard vision-based servoing pipeline is applied, as shown in Algorithm 1, with symbol definitions listed in Section VI. Both tracking stages use proportional control. To bring the target within the arm's reachable workspace, the mobile base is repositioned first. The pan-tilt camera and base joints first track the target and reposition the robot so the arm faces the target directly. This reduces the base-to-target distance and improves harvesting success.

The arm then performs an approach motion to bring the target into the gripper. Unlike the pan-tilt phase, the gripper

TABLE II
SUMMARY OF DATA PREPARATION, PREPROCESSING, AND TRAINING SETTINGS FOR YOLOV11N AND THE DIFFUSION POLICY

Aspect	YOLOv11n	Diffusion Motion Policy
Data source & amount	1 fps down-sampled images per camera from teleoperated harvesting (top and wrist).	∼3–10 image frames per second plus full robot state (position, orientation, gripper angle).
Image resolution & preprocessing	640 × 640 px; saturation ±25%, brightness ±15%, exposure ±10%, blur 0–2.5 px, noise 0–0.1%; 240–720 augmented images per camera.	320 × 240 px; random cropping to 288 × 216 px; independent augmentation per camera view.
Annotation	Semi-automated via Roboflow + Grounding DINO; manual verification.	See Figure 8, uses expert robot trajectories directly.
Training	200 epochs, batch size 8, AdamW optimizer (default Ultralytics settings).	800–1500 epochs (task dependent), batch size 64, AdamW optimizer (learning rate 1e-4, weight decay 1e-6). Image augmentation: per batch, augmentation types (brightness, contrast, saturation) and their intensities are randomly sampled.
Model-specific settings	N/A.	Diffusion steps: 100, prediction horizon: 16, observation horizon: 2, action horizon: 8.
Training time	∼6 hours.	2–10 hours (depends on dataset size).
Model output	Detected objects' IDs and 2D bounding box coordinates (max 30 Hz); two models (top and wrist camera).	16-step task/joint space motion trajectory (2–4 Hz); task-specific policy per task.

Algorithm 1 ID-Based Raspberry Approaching

Input : Tracked ID ID_t, Image center (x_c, y_c), Arm Controller
Output: `flag` indicating success
while *true* **do**
  Receive all existing IDs D
  if $ID_t \notin D$ **then**
    Exit if lost tracking too long
    continue
  end
  Extract (x_o, y_o, a_o)
  $\varepsilon_x, \varepsilon_y \leftarrow x_c - x_o, y_c - y_o$
  if *Pan-Tilt camera* **then**
    $\theta_p, \theta_t \leftarrow \theta_p + K_p \cdot \varepsilon_x,\ \theta_t - K_p \cdot \varepsilon_y$
    Send (θ_p, θ_t) to camera
    if $|\theta_p| > \theta_{tol}$ **then**
      Move mobile base using proportional control
    end
    if $|\varepsilon_x| < \varepsilon_{tol}$ *and feedback confirms* **then**
      `flag` ← true; `break`
    end
  else if *Arm-mounted camera* **then**
    if *not aligned* **then**
      Update R_T:
      $R_T \leftarrow R_T \cdot \mathrm{rotx}(K_p \cdot \varepsilon_y) \cdot \mathrm{rotz}(K_p \cdot \varepsilon_x)$
      Update T:
      $T \leftarrow \begin{bmatrix} R_T & \mathbf{p} \\ \mathbf{0}^\top & 1 \end{bmatrix}$
      Solve IK using T and send joint command
      continue
    end
    if *aligned and not yet translated* **then**
      $\mathbf{p} \leftarrow \mathbf{p} + \delta \cdot \hat{\mathbf{y}}$
      Solve IK and update pose
    end
    if $a_o \geq a_{threshold}$ *for 40 steps* **then**
      `flag` ← true; `break`
    end
  end
end

orientation is aligned with the target direction, and the arm moves along this direction. This ensures the gripper remains pointed toward the raspberry until it reaches the target within a predefined tolerance. Since the apparent object size provides

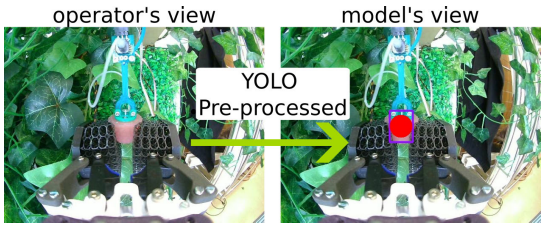


Fig. 9. Concept of integrating YOLO and the diffusion policy. YOLO provides a consistent visual marker for the target, which is used by the diffusion model during training and deployment. The operator's view will not be affected during the teleoperation.

an implicit depth cue, we use the object's pixel area as a stopping criterion for the approach motion:

$$a_o = \frac{color\ area\ of\ target\ object}{image\ size} \tag{4}$$

The threshold $a_{threshold,min}$ is calibrated from the 50 largest raspberries ($D_r \in [22, 28]$ mm from farm records) under a fixed camera-gripper configuration. The robot was positioned such that the raspberry center lay at the gripper boundary (edge case in Figure 15a), the furthest position at which the compliant gripper can still successfully grasp the fruit. The YOLO model detected the target fruit, followed by RGB-based red colour segmentation to estimate the pixel area, with 5% of boundary pixels discarded to remove noise. The average pixel area across these samples was used as the threshold. The threshold is sensitive to camera placement and requires recalibration if the mounting configuration changes, but remains consistent in our deployment as the camera is fixed to the gripper.

This threshold also informs the gripper diameter design. Under the pinhole camera model ($A_{pixel} \propto D_r^2/d^2$), when the largest raspberry ($D_{r,l} = 28$ mm) triggers the threshold at the gripper boundary (edge case in Figure 15a)), the smallest raspberry ($D_{r,s} = 22$ mm) produces the same pixel area at:

$$d_s = \frac{D_{r,s}}{D_{r,l}} \cdot d_l = \frac{22}{28}\left(L + \frac{D_g}{2}\right) \tag{5}$$

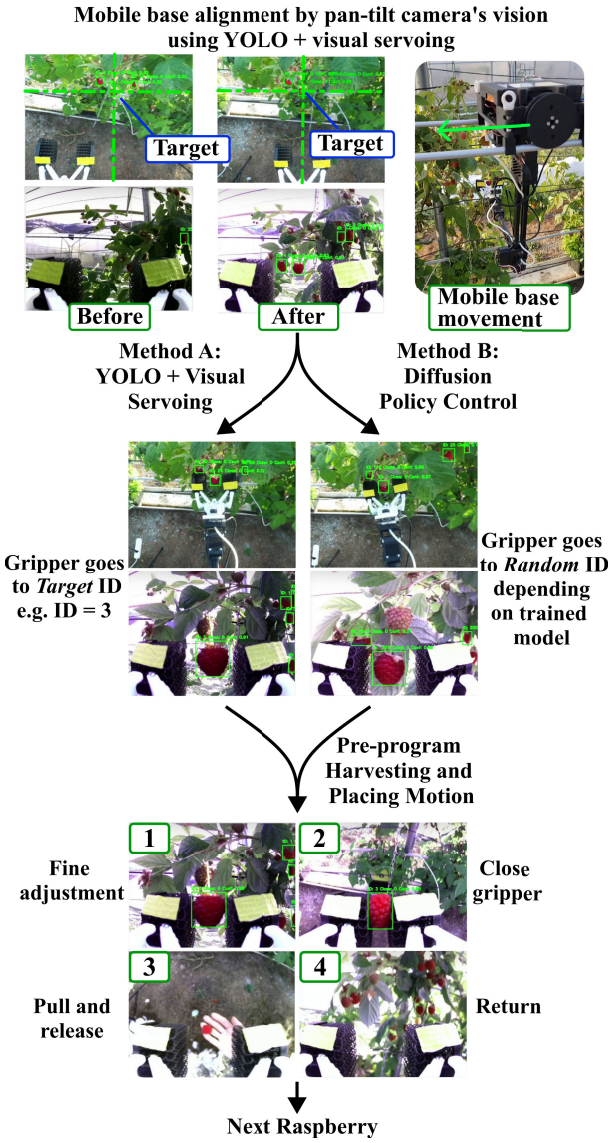


Fig. 10. Automation approaching workflow for both YOLO(left) and Diffusion(right).

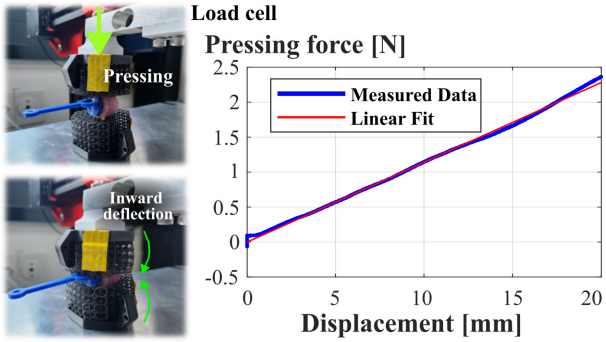


Fig. 11. Measurement of driving force required to close the gripper, showing a linear stiffness of 0.11 N/mm (11.63 g/mm).

where d_s and d_l are the distances from the camera to the raspberry center, and $L \approx 55\,\text{mm}$ is the camera-to-gripper-center distance. For the smallest raspberry's center to remain within the gripper capture region ($L - D_g/2 \le d_s \le L + D_g/2$),

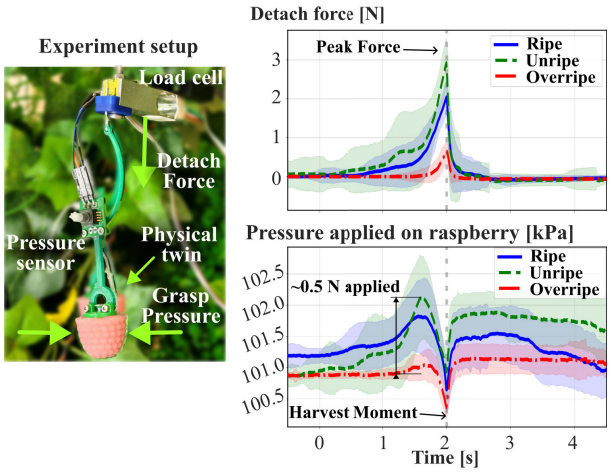


Fig. 12. Detach force profiles for raspberries of different ripeness levels and pressure applied during harvesting.

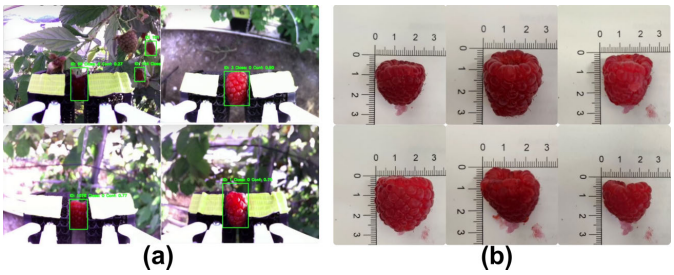


Fig. 13. (a) Gripper grasping real raspberries. (b) Harvested raspberry samples after 6 h of storage.

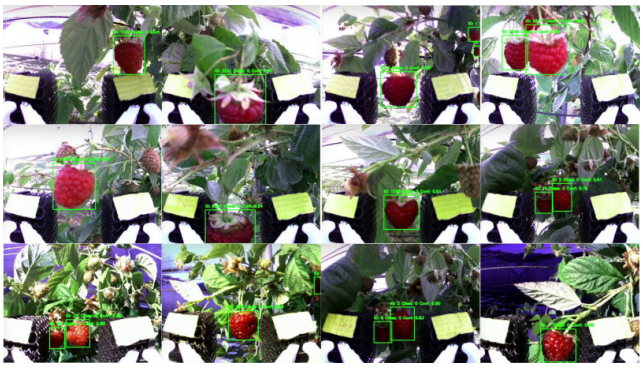


Fig. 14. Examples of gripper view in the field test.

solving yields:

$$D_g \ge \frac{6L}{25} = 13.2\,\text{mm} \tag{6}$$

The final gripper diameter of 47.5 mm provides an additional positional tolerance to accommodate tracking errors in the control algorithm and robot movement. We set $a_{threshold,min} = 3.5\%$ for a 640×480 image, with the gripper center at $(x_o, y_o) = (320, 350)$ in the image. If the detected raspberry satisfies the area threshold and remains visible for 40 consecutive steps, it is considered inside the gripper region. By using the gripper design introduced in Section II-A2, the system allows a large positional tolerance around the gripper center. The success

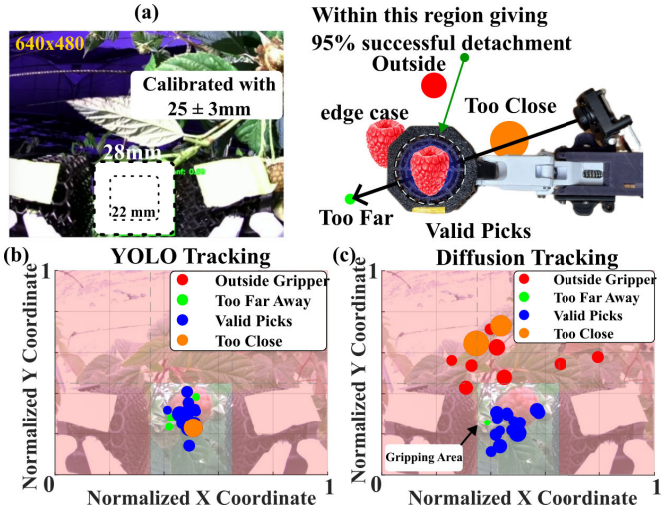


Fig. 15. (a): Illustration of raspberry position, with marker size indicating apparent raspberry size in the image. (b)-(c): Position distribution of the raspberries in the test using (b) **YOLO** model. (c) **Diffusion** model to move the arm. The larger dot size represents closer distance to the camera.

rate of this approach is 76.5%. A depth camera may further improve distance estimation and approach robustness.

B. Motion Generation Using Diffusion Policy

The diffusion policy takes the current observation and robot state as input. For example, the state includes two image frames from each camera and the current 6-DoF pose with the gripper's angle (mobile joint fixed). It then outputs a 16-step sequence, where each step consists of a 7-DOF vector representing the robots pose along with the gripper angle. However, only the first 8 steps are executed to avoid accumulated prediction errors.

For the task-specific diffusion policies, the harvesting policies used 52 demonstrations (4,732 training samples) in the laboratory and 43 demonstrations (4,643 training samples) in the field, both sampled at 3 Hz. The leaf-inspection policy used 24 demonstrations (2,793 samples) at 5 Hz, while the YDM and NDM occlusion-removal policies each used 22 demonstrations (6,452 samples) at 10 Hz. A random seed of 42 was used for training. No train/validation split was applied; all collected demonstrations were used for training, with evaluation conducted in separate deployment trials. The last training checkpoint was used for deployment.

The diffusion policy can generate longer trajectories by increasing the prediction horizon during training, but alternative settings were not explored, as they are outside the scope of this work. We follow the configuration in [43]. In our setting, the model performs inference at 0.5-4 Hz (depending on GPU), which is sufficient for our tasks. Higher frequencies can be achieved with better hardware. The output motion scale depends on the training data sampling rate: a high sampling rate produces small state changes between frames and short movements, while a low sampling rate produces larger state changes and therefore requires more interpolation to smooth the trajectory (our approach).

In addition, the diffusion model outputs an action stage value between 0 and 1 that indicates the progress of the trajectory. Although it is not a direct percentage, abnormal motion can be identified when this value remains low for a long period. Since the operator can directly monitor the robot, repeated motions in a small region or stalled behavior are easily visible from the camera view, allowing timely human intervention or correction.

C. YOLO + Diffusion Motion Generation

Finally, we integrate YOLO and the diffusion policy into a single framework. In this approach, YOLO provides explicit target information to the diffusion model, providing the policy with an explicit visual cue indicating the target during motion generation. Intuitively, YOLO gives the model a clear target indicator such as "the gripper should reach this marked region in the image", rather than requiring the model to infer the target implicitly.

To implement this, YOLO detection is run during teleoperated data collection. Along with the raw top and wrist camera images, we also record the YOLO-processed images where the target is replaced by a constant marker such as a red dot bounded by a blue box, as shown in Figure 9. This creates consistent visual cues across lighting conditions. During deployment, the policy only requires the processed view to identify the target, and the same marker can be applied to other fruits such as tomatoes.

However, YOLO detection may fail on individual frames. Therefore, we do not force all images to be YOLO-processed. Instead, the model receives a mixture of raw images and YOLO-processed images, which improves robustness to occasional detection failures and prevents the policy from relying on a single visual format. The training data are also randomized to better reflect real deployment conditions.

A remaining limitation is the need for a pretrained YOLO model before collecting the second round of training data, as well as the systems reliance on YOLO stability. This limitation is manageable because the proposed framework naturally accumulates data over time, enabling continuous refinement of the detection model.

IV. Experimental Setups

Multiple experiments were conducted in both field and indoor farming environments. Although several hardware components changed across iterations, including the linear traversing infrastructure, which evolved from a 5 mm steel cable to a 10 mm aluminum tube and later to a 30×60 mm aluminum profiles and 16×16 mm steel rails to increase system stability, the overall system framework remained unchanged. The main hardware units used throughout these experiments are summarized in Table III.

A. Task Decomposition

In this work, each task is decomposed into distinct phases to simplify motion generation and controller design. Certain actions, such as pulling the raspberry and placing it into a known location, can be pre-programmed without

TABLE III
SUMMARY OF HARDWARE COMPONENTS

Unit	Specification	Notes/Usage
Power	12V 5A supply 12V LiPo 3400 mAh	1–1.5 h runtime Full CAFEs power
CAFEs computer	Raspberry Pi 5 8GB RAM	Camera streaming Kinematics calculation Actuator control
Operator computer	Intel Ultra 9 185H 64GB RAM	Teleoperation UI handling Data storage
Server computer	RTX 4090 (lab) RTX A2000 8GB (field) i7-12850HX / i9-14900K 16GB / 32GB RAM	YOLO detection Diffusion motion Model training Model storage
Camera	RGB USB cameras FoV 135° / 95° / 70° Upgraded to 175°	Top camera Wrist camera

model training or object detection, while other parts require perception-driven control. For example, the overall harvesting workflow is shown in Figure 10 and consists of three main phases:

1) *Algorithm driven*: navigating the CAFEs platform toward the target using the mobile base and YOLO detection from the top camera;
2) *Algorithm or policy driven*: approaching the target using either YOLO-based control or diffusion-policy control;
3) *Pre-programmed*: harvesting the target and placing it into a predefined drop-off position, followed by moving the arm to a home or collection pose.

This framework was also applied to other tasks such as leaf inspection and removing an occlusion shown in this work. Similar multi-model strategies have also been adopted in other manipulation and locomotion studies.

B. Evaluation Protocol

The environment is not strictly controlled during data collection or deployment, with cameras set to auto-exposure, auto-focus, and auto-balance. Demonstrations are collected via teleoperation at randomized times, with additional LED lighting provided only when natural sunlight is unavailable during field tests. The rail structures are fixed in position. Images are randomly augmented, and policies are evaluated under randomized conditions including daylight variation, indoor lighting changes, leaf occlusions, and varying arm positions along the rail. Data are processed according to Table II. Success criteria are defined per task:

- *Raspberry approaching (Section V-B):* a trial is considered successful when the raspberry enters the gripper's target region at the end of the approaching phase. Upon success, the final grasp-and-detach motion is executed using a preprogrammed motion.
- *Leaf inspection (Section V-C):* a trial is considered successful when one arm successfully grasps the target leaf and lifts it, while the second arm approaches and captures an image of the underside of the leaf. A trial is considered unsuccessful if the leaf is not grasped or if the underside of the leaf cannot be imaged.
- *Occlusion clearing and harvesting (Section V-D):* performance is evaluated using a six-point scoring scheme, where each of the following actions is worth one point: (1) the right arm detects the raspberry, (2) the right arm moves toward the blocking leaf or stem, (3) the right arm removes the leaf or stem, (4) the left arm detects the raspberry, (5) the left arm reaches the raspberry, and (6) the left arm grasps the raspberry. If a stage fails on the first attempt but succeeds on a later attempt, half a point is awarded.
- *Human–robot collaboration (Section V-E):*performance is evaluated using a five-point scoring scheme, where each of the following actions is worth one point: (1) the left arm remains on stand-by until the human completes the leaf-removal action, (2) the detector identifies the target after leaf removal, (3) the left arm moves toward the target, (4) the left arm performs a successful grasp without a second attempt, and (5) the left arm maintains the grasp without unintended reopening due to the operator's motion.

More details on the scores can be found in the corresponding result sections.

V. EXPERIMENTAL RESULT

A. Harvesting Gripper Evaluation

The gripper is designed for raspberries with diameters under 25 mm. When the raspberry is near the gripper center, the flexible lattice structure fully wraps around it. Even with moderate off-center positioning, the compliant surface conforms to the fruit and maintains a stable and gentle grasp. Using the physical twin testing setup in [45] and the actual raspberry size dataset, the gripper diameter was optimized to 47.5 mm to balance tracking error tolerance, material softness, and end-effector size.

The experimental setup used for characterizing stiffness and harvesting behavior is shown in Figure 11 and Figure 12. As shown in Figure 11, the driving force increases approximately linearly against the deformation, requiring between 1.3 N and 2.4 N to close the gripper. These characteristics allow the gripper actuator to use simple position control, with actuator displacement approximately proportional to the driving force.

We conducted 200 grasp-and-pull trials on randomized gripper's position to evaluate gripper performance. A total of 189 successful detachments were achieved when the raspberry entered the blue capture region shown in Figure 15(a), including some edge cases.

We also examined the force acting on the raspberry surface to avoid damage. Previous studies [46], [47], [48] report typical raspberry detachment forces of up to 2 N depending on the ripeness. As shown in Figure 12, our gripper comfortably meets this requirement with a safety factor of 1.5. Pressure measurements on the physical twin show that the applied surface force on a 25 mm by 20 mm area is less than 0.5 N, which provides a gentle grasp and minimizes damage to the raspberry.

Since the gripper achieves a high grasp success rate once the fruit enters the effective capture region, the harvesting task

is separated into approaching and grasping phases. The main challenge therefore lies in accurately approaching the ready-to-harvest position.

Finally, Figure 13(a) shows how the real raspberries deformed inside the gripper. Meanwhile, Figure 13(b) shows six harvested raspberry samples after 6 h of storage at room temperature.

B. Task 1: Approach to Harvest Position

The approaching phase is the most challenging part of the harvesting workflow, as it requires perception-driven motion to align the gripper with the raspberry before the pre-programmed detachment motion can begin. Therefore, in this section, gripper alignment is considered the goal. We present results using both a YOLO-based visual servoing controller, which is commonly used in harvesting systems, and a diffusion-based policy enabled by the proposed framework. The YOLO-based controller is included only as a baseline to demonstrate that the hardware system is compatible with such commonly used methods. To prepare the data for both YOLO and diffusion training, we conducted 25 and 43 demonstrations in preliminary lab and field tests, respectively, both within one hour.

Early lab tests using artificial raspberries revealed the importance of camera selection. The wrist camera initially used had a narrow field of view of 60 degrees, causing frequent tracking loss whenever the fruit left the image. As a result, the success rate in these tests reached only 60% over 30 trials.

The field trials were conducted from daytime to evening, as shown in Fig. 14, to evaluate the policy under lighting conditions different from those encountered during data collection, which had been completed the previous morning. Due to the late harvesting season, fewer than 100 raspberries were available after the data collection: Of these, 11 were used to evaluate YOLO-based control, 25 were used to evaluate diffusion-based control, and the remainder were used for motion demonstrations and other measurements. As mentioned earlier, a trial was considered successful when the raspberry entered the grippers target region at the end of the approaching phase. Under the same hardware setting, the alignment success rate of the diffusion policy reached 56% in the field deployment. The YOLO-based visual servoing achieved a 76.5% success rate in the field tests.

To further analyze the approaching accuracy, Figure 15(b) shows the final fruit positions in the camera frame. During field testing, noticeable vibration was observed when using the diffusion policy, whereas the YOLO-based approach produced smoother motion. This difference arises from the distinct control strategies employed by the two methods, as well as the high speed and acceleration profile applied to the diffusion policy. In combination with the limited structural rigidity of the rail system, the diffusion-based approach exhibited more frequent start–stop behavior and less smooth motion. This issue was later mitigated using a more rigid rail structure. Execution time was also measured for reference, as shown in Figure 16. The characteristics of the two methods are as follows: 1. The YOLO-based controller outputs a single task-space command at each inference, resulting in an average

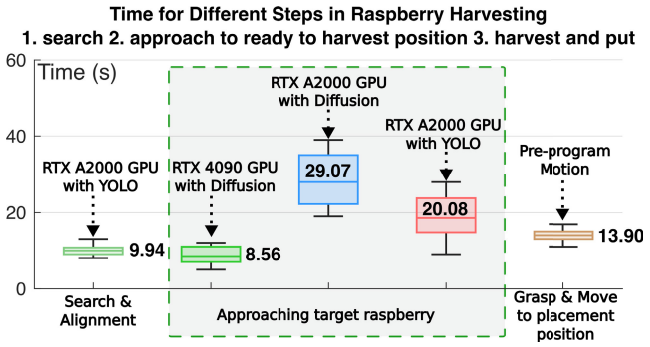


Fig. 16. Average execution time for successful stages of the harvesting task using the diffusion and YOLO-based methods.

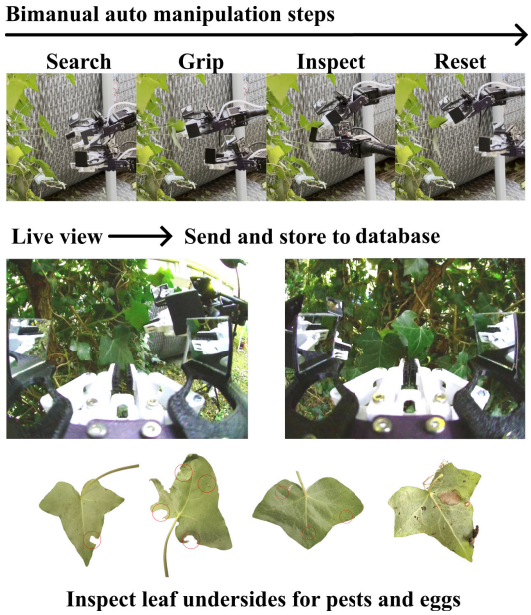


Fig. 17. Flow of bimanual leaf turning. The average time for leaf turning is 29 seconds using diffusion policy.

execution time of approximately 20 seconds. 2. The diffusion policy generates a multi-step trajectory at each inference, leading to an average execution time of approximately 29 seconds in the field tests.

The YOLO-based controller achieved a consistent operating rate of 30 Hz on both the RTX 4090 and A2000 GPUs, limited primarily by the streaming camera frame rate. With a more powerful GPU, the diffusion policy reached an inference rate of approximately 4 Hz on the RTX 4090, resulting in an average approach time of about 8.5 seconds because of the reduced inference latency. The success rate increased to 87.5% although this improvement may also be partially attributable to the more consistent hydroponic setup.

C. Task 2: Leaf Inspection (Diffusion)

To demonstrate bimanual automation, we performed a leaf inspection task in which the operator remained seated in a separate room. The demonstration was first collected through teleoperation and then used to train the diffusion policy to reproduce the bimanual motion. The teaching procedure is the

same as in the harvesting task, with the only difference being the use of both arms. As shown in Figure 17, the task has three stages: searching for the leaf, gripping it, and turning it for inspection.

The experiment demonstrated that the target leaf could be grasped and examined using the learned bimanual motion. We performed twenty trials on a leaf using different approach strategies. 13/20 trials were successful in grasping and taking a picture, while failures occurred when the target leaf was not gripped correctly. The average execution time is about 29 seconds which is slightly slower than the demonstration time (around 20 seconds on average). However, we did not quantitatively evaluate or standardize other performance of this motion. This result therefore serves only as a demonstration of a complex and implicit action that cannot be described or pre-planned easily using YOLO-based or traditional visual servoing methods.

D. Task 3: Clear Occlusion and Approach Target (YOLO-Based Diffusion)

To further demonstrate the capability of the proposed framework comparing to the single arm system, we conducted another bimanual task commonly seen in farming: removing a blocking leaf and then reaching the target fruit. As described in Section III-C, we trained the diffusion policy using a mixture of YOLO-processed images and normal images rather than exclusively raw images. All other training settings remained unchanged. We collected 22 demonstrations within 45 minutes such that the operator removed the occluding leaf and moved the gripper toward a ready-to-grip position. In the demonstration, the operator finished the task at roughly 10 to 25 seconds. This experiment evaluates the three configurations within the proposed system under varying conditions in a predefined region, rather than under identical scene settings.

For the YDM/NDM setting, all experiments were conducted in an indoor hydroponic setup with natural daylight from 7am to 5pm entering through a window toward the plant rows, supplemented by LED lighting (25W) placed at varying positions and directed toward the plants. The cameras used were an IMX335 5MP with a 120° field of view, operating with auto exposure, auto white balance, and automatic brightness settings. Human demonstrations were collected between 3-4pm with fixed LED positions and varying daylight, while deployment experiments were conducted between 10am to 7pm under natural daylight with LED positions changed. To introduce additional lighting variation, some trials were performed with LEDs turned off or intermittently flashing near sunset. To simulate realistic occlusions, blocking leaves were randomly placed during both data collection and deployment. The target fruit x–y location remained nearly unchanged with height variation of ± 5 mm, while the robotic arm base position was varied by ± 15 mm along the rail (x direction) to generate mobile-base alignment errors. These variations were applied across trials to ensure robustness to lighting changes, occlusions, and positioning variations. All trials started from the same predefined end-effector task-space pose. The three settings were evaluated in repeated blocks of ten trials in the order of NDM with raw images, YDM with raw images, and YDM with YOLO-processed images, until 50 trials were completed for each setting. The target position, mobile-base position, and blocking-leaf configuration were not matched one-to-one across settings. Therefore, the results represent comparisons under comparable environmental conditions rather than strictly identical initial scenes.

The performance is evaluated as described in Section IV-B. In the following, we denote the YOLO-assisted diffusion model as YDM and the normal diffusion model without YOLO as NDM. We trained the YDM and NDM with the two dataset where the only difference was the added marker in the images.

We compared three settings on both trained models with each of them using 50 trials (in total 150 trials): NDM with normal images input, YDM with normal images input, and YDM with YOLO-processed images input. This comparison illustrates how the use of mixture images improves the performance of a visually complex bimanual task. As shown in Figure 18, using only raw images produced limited performance. In this input-representation ablation experiment, when both policies were evaluated using raw images. The two policies share the same diffusion-policy architecture and differ mainly in their training input representations. The policy trained with a mixture of raw and YOLO-processed images achieved an average score of 4.29 out of 6, compared with 3.53 for the policy trained using raw images only. The observed difference indicates that the training input representation affects policy performance.

Among the three evaluated configurations, YDM with YOLO-processed deployment images achieved the highest average score of 4.51 out of 6. However, because both the training distribution and testing input differ among the configurations, the present comparison does not isolate the independent contribution of YOLO processing. The results should therefore be interpreted as observed performance differences among the evaluated input configurations. For example, in some frames, the YOLO fails to detect the target and mask it. But the diffusion model still has the perception of moving toward the un-masked object. The best performance was obtained when YDM used YOLO-processed images, achieving 4.51 out of 6. In this setting, the explicit marker provided by YOLO encoded the target position and size implicitly, enabling more reliable motion planning for both arms.

These results demonstrate that combining YOLO-processed images with the diffusion model improves performance in bimanual tasks that involve removing occlusion and reaching a partially hidden target. The YOLO marker helps the model interpret the scene structure and supports more robust execution of complex action sequences.

E. Task 4: Collaboration Between Human and Policy Model

We also explored the idea that uses the same trained policy model together with human collaboration to handle visually complex scenes. As shown earlier in Figure 18, the success rate of the right arm grasping a stem or leaf is low across all three approaches. We hypothesize that this is due to the dense green background, which makes the target stem or leaf difficult to distinguish with the limited number of demonstrations. Such

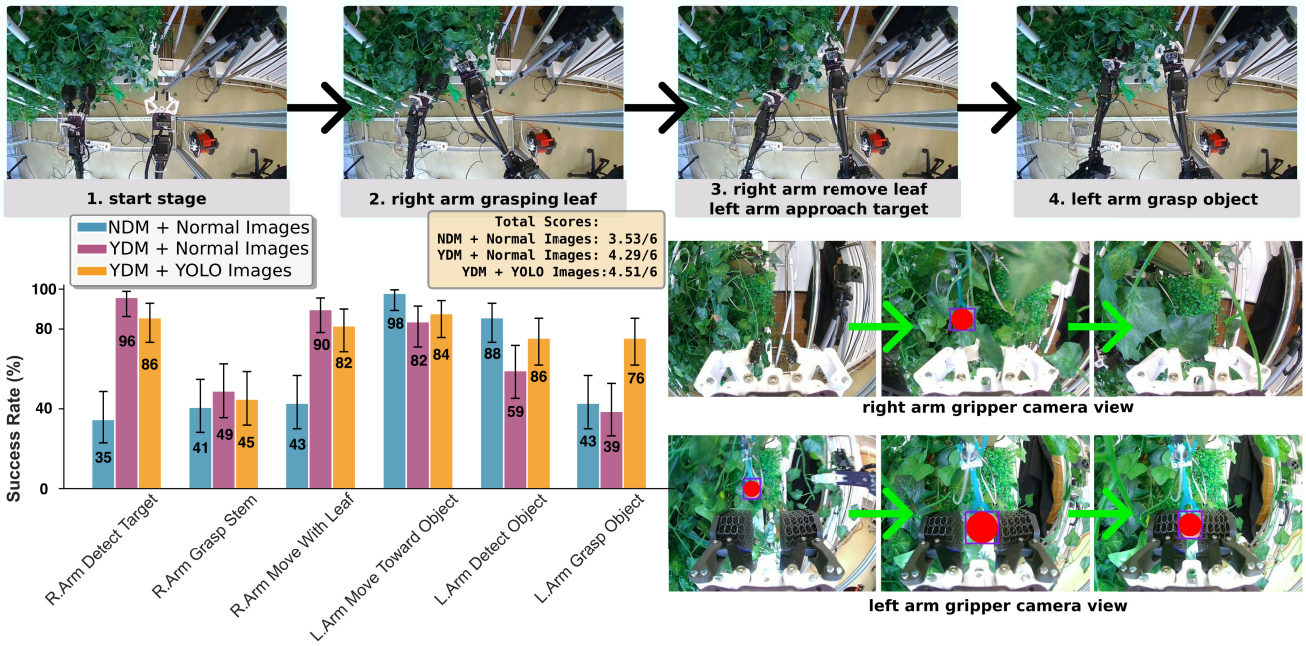


Fig. 18. Performance comparison for the bimanual leaf-removal and fruit-reaching task. Top: key stages of the task using the two arms. Bottom left: success rates with 95% confidence level for each subtask under three settings: NDM with raw images, YDM with raw images, and YDM with YOLO-processed images. Bottom right: examples of YOLO-processed images used as input for YDM.

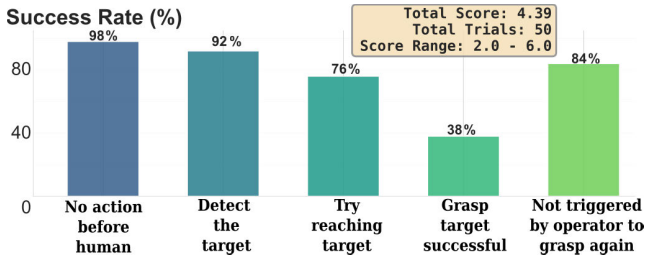


Fig. 19. Success rates of the human–robot collaborative setting. The right arm is fully teleoperated by the operator to remove the leaf, while the left arm is controlled by the trained policy. The five metrics evaluate whether the left arm waits for human action, detects the target after the leaf is removed, reaches the target, performs a successful grasp, and maintains the grasp without unintended reopening.

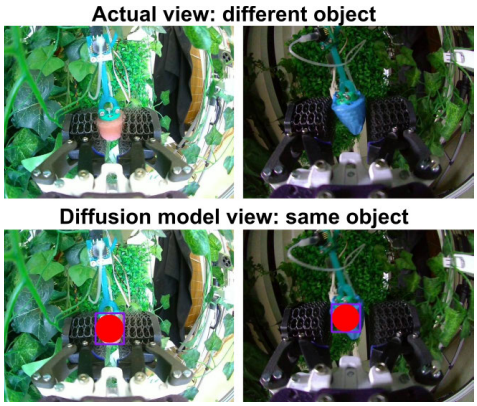


Fig. 20. Action-transfer experiment using anonymized masked images.

sensitivity in early deployment is expected when the training set is small.

However, the proposed framework allows direct operator intervention. In this experiment, the same YDM with normal images as input was reused for automatic control of the left arm, while the right arm was directly controlled by a human operator, enabling simultaneous human–policy bimanual operation. The model still received the full robot state and both camera views, while the right-arm output was bypassed for real-time human control.

We evaluated this setting using five one-point criteria: (1) the left arm remains on stand-by until the human completes the leaf-removal action, (2) the YOLO detector identifies the target after leaf removal, (3) the left arm moves toward the target, (4) the left arm performs a successful grasp without allowing a second attempt, and (5) after grasping, the left arm is not triggered to reopen the gripper due to any motion of the right arm.

As shown in Figure 19, the collaborative setup significantly improves overall task performance. The left arm successfully detects the target in 92% of trials, reaches it in 76%, and completes the grasp in 38%. Importantly, in 98% of trials, the left arm waits correctly for the human action, and in 84% of trials it maintains the grasp without being affected by the operators motion on the right arm. The successful grasping rate remains similar to the previous test in Figure 18. These results show that the framework can incorporate human assistance when occlusions or other unseen conditions occur. Online learning [44] may further improve the success rate of specific actions.

F. Transfer Action to Different Object Using Masked Images

In addition to the experiments using masked images as training data (Section III-C) and task transfer [44], we also

TABLE IV
SYMBOLS USED IN ALGORITHM 1

Symbol	Meaning
ID_t	Tracked object ID
(x_c, y_c)	Image center coordinates (pixels)
(x_o, y_o)	Tracked object's coordinates in the image (pixels)
a_o	Area of the tracked object (pixels^2)
ε_x	Horizontal image error: $x_c - x_o$
ε_y	Vertical image error: $y_c - y_o$
θ_p	Pan angle of the camera (radians)
θ_t	Tilt angle of the camera (radians)
K_p	Proportional gain for pan/tilt or rotation control
θ_{tol}	Tolerance threshold for pan angle alignment
ε_{tol}	Tolerance threshold for pixel alignment
R_T	Rotation matrix of the camera or arm end-effector
T	Full pose transformation matrix
$\mathbf{p}$	3D position vector of the end-effector
$\hat{\mathbf{y}}$	Unit vector along the camera's forward (Y) axis
δ	Step size for forward motion during approach
$a_{\text{threshold}}$	Minimum object area to stop translation
`flag`	Boolean indicating whether the object is centered

tested a version in which the object was changed in appearance, as illustrated in Figure 20. Although the target object may change, the required motion (e.g., reach and grasp) can remain similar. The transfer experiment reused the previously trained YDM without any additional diffusion-policy training or fine-tuning, under the same testing conditions as the previous YDM experiment. The new blue target was identified using HSV-based color segmentation with a fixed HSV range of $(83, 117, 187)$ to $(123, 197, 255)$. Other blue objects were removed from the background so that the segmented region corresponded to the target. The target region was then converted into the same mask representation used during the original YDM training and provided directly to the unchanged YDM. A trial was considered successful when the robot reached the target and completed a grasp.

Preliminary results show that this approach achieved a 68% success rate (in 50 trials) for reaching and grasping the masked target, without additional diffusion-policy tuning. However, this early result is still limited by the mixed composition of the training dataset, where both raw and masked images coexist. Further software and engineering refinement is required to increase the proportion of masked images or fully remove unprocessed ones.

VI. DISCUSSION AND CONCLUSION

This work has been under continuous development since January 2024. Many engineering problems and stability issues have been overcome at different stages of this work. The hardware, software, and system framework have evolved together, and the goal of this paper is to introduce a unified framework that integrates teleoperation, linear traversing robotic arms, human–robot collaboration, and multiple vision–guided motion generation methods for general agricultural tasks. At this stage, we do not focus on fine-tuning YOLO or diffusion models or address their known challenges in depth. Instead, we develop a system that enables these components to operate together as part of a flexible and extensible workflow.

As shown by the stage-wise results in Fig. 18, failures occurred at multiple stages of the manipulation process. One possible contributor is the limited coverage and diversity of the demonstration set. However, this cannot be confirmed from the current experiments because no demonstration-size or coverage ablation was conducted. Other factors, including perception errors, target misalignment, structural vibration, and variability in target and occlusion configurations, may also contribute to the observed failures. Online correction learning may help address these failures by using targeted human corrections during deployment [44]. This approach can also be integrated into our system.

The operating frequency depends on the implementation and directly affects task execution time. A faster GPU reduces training and inference time but does not change the diffusion-policy formulation. It can also support larger models and training settings when needed. In practice, policy performance also depends on the coverage and quality of the demonstration data, as reported in previous diffusion-based studies. This further suggests that diffusion performance is mainly determined by demonstration coverage and high-quality motion data, as noted in numerous diffusion-based studies.

During system development, we found that corrective demonstrations, structured inputs, and task decomposition were important for diffusion-policy training. A single long demonstration, such as searching for a raspberry and placing it into a basket, contains multiple stages with different success conditions and goals. Models often struggle to learn such long behaviors. We therefore use YOLO-based visual servoing for stages with explicit targets and the diffusion policy for motions that are difficult to pre-program. The human operator provides supervision, handles unseen scenarios, and supplies corrective demonstrations from failure cases. Over time, this multi-model system can accumulate task knowledge and continue improving through human guidance.

We only used a limited number of demonstrations across the tasks. The goal of this work is not to create a large open-source dataset for a single task, but to show that the proposed framework can rapidly adapt to different agricultural tasks with minimal data. Performance is expected to improve with more training data before eventually saturating. Prior work has shown that diffusion models can scale to large datasets and complex tasks such as folding clothes, loading dishwashers, using tools, or pouring liquids. Our aim here is to highlight how the proposed system structure supports key agricultural operations rather than optimizing one specific policy to saturation.

Future work includes extending the framework to incorporate additional learning methods such as reinforcement learning, improving hardware stability and end-effector designs, enabling multi-arm collaboration, expanding to tasks such as pruning and inspection, and integrating global planning for full autonomy. Adding a depth camera would improve distance estimation and robustness without requiring changes to the framework. Overall, this work presents a promising direction for developing scalable, adaptable, and human-centered agricultural robotic systems.

APPENDIX
SYMBOL DEFINITIONS

See Table IV.

REFERENCES

[1] J. Sun et al., "Fruit flexible collecting trajectory planning based on manual skill imitation for grape harvesting robot," *Comput. Electron. Agricult.*, vol. 225, Oct. 2024, Art. no. 109332.

[2] L. Yang, T. Noguchi, and Y. Hoshino, "Development of a pumpkin fruits pick-and-place robot using an RGB-D camera and a YOLO based object detection AI model," *Comput. Electron. Agricult.*, vol. 227, Dec. 2024, Art. no. 109625.

[3] J. Rong et al., "Decoupled motion planning method for 7-DOF manipulator and lifting joint in automated tomato harvesting," *Comput. Electron. Agricult.*, vol. 237, Oct. 2025, Art. no. 110693.

[4] P. Zhang et al., "A parallel dual-arm robotic control method of white asparagus based on moving-looking-harvesting coordination and asynchronous harvest cooperation," *Comput. Electron. Agricult.*, vol. 232, May 2025, Art. no. 110046.

[5] T. Botterill et al., "A robot system for pruning grape vines," *J. Field Robot.*, vol. 34, no. 6, pp. 1100–1122, Sep. 2017.

[6] T. Meraj, M. I. Sharif, M. Raza, A. Alabrah, S. Kadry, and A. H. Gandomi, "Computer vision-based plants phenotyping: A comprehensive survey," *iScience*, vol. 27, no. 1, Jan. 2024, Art. no. 108709.

[7] P. Wei, P. Ragbir, S. G. Vougioukas, and Z. Kong, "Vision-based navigation of unmanned aerial vehicles in orchards: An imitation learning approach," *Comput. Electron. Agricult.*, vol. 238, Nov. 2025, Art. no. 110802.

[8] D. Mahalingam, A. Patankar, K. Phi, N. Chakraborty, R. McGann, and I. Ramakrishnan, "Containerized vertical farming using cobots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 17897–17903.

[9] Z. Wang, Y. Xun, Y. Wang, and Q. Yang, "Review of smart robots for fruit and vegetable picking in agriculture," *Int. J. Agricult. Biol. Eng.*, vol. 15, no. 1, pp. 33–54, 2022.

[10] P. Gonzalez-De-Santos, R. Fernández, D. Sepúlveda, E. Navas, L. Emmi, and M. Armada, "Field robots for intelligent farms—Inhering features from industry," *Agronomy*, vol. 10, no. 11, p. 1638, Oct. 2020.

[11] M. Spagnuolo, G. Todde, M. Caria, N. Furnitto, G. Schillaci, and S. Failla, "Agricultural robotics: A technical review addressing challenges in sustainable crop production," *Robotics*, vol. 14, no. 2, p. 9, Jan. 2025.

[12] L. Benos et al., "Human–robot interaction in agriculture: A systematic review," *Sensors*, vol. 23, no. 15, p. 6776, 2023.

[13] S. Mahmoudi, A. Davar, P. Sohrabipour, R. B. Bist, Y. Tao, and D. Wang, "Leveraging imitation learning in agricultural robotics: A comprehensive survey and comparative analysis," *Frontiers Robot. AI*, vol. 11, Oct. 2024, Art. no. 1441312.

[14] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu, "Robot learning on the job: Human-in-the-loop autonomy and learning during deployment," *Int. J. Robot. Res.*, vol. 44, nos. 10–11, pp. 1727–1742, Sep. 2025.

[15] A. Porichis, M. Inglezou, N. Kegkeroglou, V. Mohan, and P. Chatzakos, "Imitation learning from a single demonstration leveraging vector quantization for robotic harvesting," *Robotics*, vol. 13, no. 7, p. 98, Jun. 2024.

[16] C. H. Kim, A. Silwal, and G. Kantor, "Autonomous robotic pepper harvesting: Imitation learning in unstructured agricultural environments," *IEEE Robot. Autom. Lett.*, vol. 10, no. 4, pp. 3406–3413, Apr. 2025.

[17] A. Pal, A. C. Leite, and P. J. From, "A novel end-to-end vision-based architecture for agricultural human–robot collaboration in fruit picking operations," *Robot. Auto. Syst.*, vol. 172, Feb. 2024, Art. no. 104567.

[18] T. Liu et al., "Immersive human-machine teleoperation framework for precision agriculture: Integrating UAV-based digital mapping and virtual reality control," *Comput. Electron. Agricult.*, vol. 226, Nov. 2024, Art. no. 109444.

[19] S. A. Deka et al., "Enhancing precision agriculture through human-in-the-loop planning and control," in *Proc. IEEE 20th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2024, pp. 78–83.

[20] H. Wang et al., "Geometry-aware 3D point cloud learning for precise cutting-point detection in unstructured field environments," *J. Field Robot.*, vol. 42, no. 7, pp. 3063–3076, Oct. 2025.

[21] J. Kim, H. Pyo, I. Jang, J. Kang, B. Ju, and K. Ko, "Tomato harvesting robotic system based on deep-ToMaToS: Deep learning network using transformation loss for 6D pose estimation of maturity classified tomatoes with side-stem," *Comput. Electron. Agricult.*, vol. 201, Oct. 2022, Art. no. 107300.

[22] F. Meng, J. Li, Y. Zhang, S. Qi, and Y. Tang, "Transforming unmanned pineapple picking with spatio-temporal convolutional neural networks," *Comput. Electron. Agricult.*, vol. 214, Nov. 2023, Art. no. 108298.

[23] M. Ariza-Sentís, S. Vélez, R. Martínez-Peña, H. Baja, and J. Valente, "Object detection and tracking in precision farming: A systematic review," *Comput. Electron. Agricult.*, vol. 219, Apr. 2024, Art. no. 108757.

[24] D. Wu et al., "Detection of camellia oleifera fruit in complex scenes by using YOLOv7 and data augmentation," *Appl. Sci.*, vol. 12, no. 22, p. 11318, Nov. 2022.

[25] H. Zhao et al., "Real-time object detection and robotic manipulation for agriculture using a YOLO-based learning approach," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2024, pp. 1–6.

[26] Y. Tang et al., "Recognition and localization methods for vision-based fruit picking robots: A review," *Frontiers Plant Sci.*, vol. 11, p. 510, May 2020.

[27] D. Agarwal and A. Bhargava, "On-tree fruit detection system using darknet-19 based SSD network," *J. Food Meas. Characterization*, vol. 18, no. 8, pp. 7067–7076, Aug. 2024.

[28] L. Fu, F. Gao, J. Wu, R. Li, M. Karkee, and Q. Zhang, "Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review," *Comput. Electron. Agricult.*, vol. 177, Oct. 2020, Art. no. 105687.

[29] Y. Ge, Y. Xiong, G. L. Tenorio, and P. J. From, "Fruit localization and environment perception for strawberry harvesting robots," *IEEE Access*, vol. 7, pp. 147642–147652, 2019.

[30] T. A. Ciarfuglia, I. M. Motoi, L. Saraceni, and D. Nardi, "Pseudo-label generation for agricultural robotics applications," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 1685–1693.

[31] F. Xiao, H. Wang, Y. Xu, and R. Zhang, "Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review," *Agronomy*, vol. 13, no. 6, p. 1625, Jun. 2023.

[32] L. Li and H. Kasaei, "Enhanced view planning for robotic harvesting: Tackling occlusions with imitation learning," 2025, *arXiv:2503.10334*.

[33] D. Chen, X. Qi, Y. Zheng, Y. Lu, Y. Huang, and Z. Li, "Synthetic data augmentation by diffusion probabilistic models to enhance weed recognition," *Comput. Electron. Agricult.*, vol. 216, Jan. 2024, Art. no. 108517.

[34] E. Wyckoff, R. Reza, and F. Moreu, "Human-in-the-loop control of dynamics and robotics using augmented reality," *J. Dyn. Disasters*, vol. 1, no. 1, Mar. 2025, Art. no. 100005.

[35] Z. Liu, R. M. Sampurno, R. M. R. D. Abeyrathna, V. M. Nakaguchi, and T. Ahamed, "Development of a collision-free path planning method for a 6-DoF orchard harvesting manipulator using RGB-D camera and Bi-RRT algorithm," *Sensors*, vol. 24, no. 24, p. 8113, Dec. 2024.

[36] H. H. Cheng and J. Hughes, "CAFEs: Cable-driven collaborative floating end-effectors for agriculture applications," 2025, *arXiv:2503.00514*.

[37] H. H. Cheng and J. Hughes, "8-DoFs cable driven parallel robots for bimanual teleportation," 2025, *arXiv:2504.01554*.

[38] H. H. Cheng and J. Hughes, "Enhancing bimanual teleoperation with variable shoulder distance: Manipulation in varying-scale of applications," *npj Robot.*, vol. 3, no. 1, p. 38, Nov. 2025.

[39] M. A. Ali, H. A. Park, and C. S. G. Lee, "Closed-form inverse kinematic joint solution for humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 704–709.

[40] Q. Guan, B. Dai, H. H. Cheng, and J. Hughes, "Lattice structure musculoskeletal robots: Harnessing programmable geometric topology and anisotropy," *Sci. Adv.*, vol. 11, no. 29, Jul. 2025, Art. no. eadu9856, doi: 10.1126/sciadv.adu9856.

[41] Q. Guan, B. Dai, H. Sonar, and J. Hughes, "Inverse co-design of mechanical and sensory properties in soft lattice foams for multifunctional wearables," *Adv. Sci.*, vol. 12, no. 44, p. 07102, Nov. 2025.

[42] C. Chi et al., "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," 2024, *arXiv:2402.10329*.

[43] C. Chi et al., "Diffusion policy: Visuomotor policy learning via action diffusion," *Int. J. Robot. Res.*, vol. 44, nos. 10–11, pp. 1684–1704, Sep. 2025.

[44] C. Pan, H. H. Cheng, and J. Hughes, "Online imitation learning for manipulation via decaying relative correction through teleoperation," 2025, *arXiv:2503.15368*.

[45] K. Junge and J. Hughes, "Soft sensorized physical twin for harvesting raspberries," in *Proc. IEEE 5th Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2022, pp. 601–606.

[46] J. Rabcewicz, P. Białkowski, and P. Konopacki, "Evaluation of the possibility of shaking off Raspberry fruits with a pulsating air stream," *J. Horticultural Res.*, vol. 25, no. 1, pp. 61–66, Jun. 2017.

[47] J. Rabcewicz and J. Danek, "Evaluation of mechanical harvest quality of primocane raspberries," *J. Fruit Ornamental Plant Res.*, vol. 18, no. 2, pp. 239–248, 2010.

[48] L. Giongo, M. Ajelli, P. Poncetta, M. Ramos-García, P. Sambo, and B. Farneti, "Raspberry texture mechanical profiling during fruit ripening and storage," *Postharvest Biol. Technol.*, vol. 149, pp. 177–186, Mar. 2019.



Hung Hon Cheng (Member, IEEE) received the B.Eng. (Hons.) and Ph.D. degrees from The Chinese University of Hong Kong in 2017 and 2022, respectively. He worked on surgical robotic systems at Cornerstone Robotics and was a Post-Doctoral Researcher at the CREATE Laboratory, EPFL, Switzerland. He was the Principal Investigator of an SNSF BRIDGE-funded project on low-cost, scalable robotic system for agricultural automation. His research interests include robotic design, modeling, control, particularly cable-driven parallel robots, robotic manipulation, teleoperation, human–robot collaboration, and learning-based control, with applications in construction and agriculture.



Cheng Pan (Graduate Student Member, IEEE) received the bachelor's degree from Shanghai University, China, and the master's degree from the Technical University of Munich, Germany. He is currently pursuing the Ph.D. degree in AI and robotics with the CREATE Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, under the supervision of Prof. Josie Hughes. His research interests include imitation learning, reinforcement learning, vision–language-action model, dexterous robotic manipulation, and tactile sensing.



Alessio Desogus received the B.Sc. and M.Sc. degrees in mechanical engineering from EPFL, Switzerland. During his studies, he joined the CREATE Laboratory as a Graduate Student Researcher, focusing on robotic manipulation, learning-based control, and autonomous robotic systems. He is currently a Robotics Engineer at Mimic Robotics, Zürich, Switzerland, focusing on robot learning, control, and simulation for dexterous robotic manipulation.



Qinghua Guan (Member, IEEE) received the B.Eng. degree in composite materials in 2015 and the M.S. and Ph.D. degrees in engineering mechanics in 2017 and 2023, respectively. He was a Visiting Student at the University of Maryland, College Park, MD, USA, in 2015, and a Visiting Ph.D. Student at EPFL, Lausanne, Switzerland, from 2022 to 2023. Since 2023, he has been with the CREATE Laboratory, EPFL, first as a Post-Doctoral Researcher and currently as a SNSF Ambizione Fellow. His research interests include soft robotics, architected materials, computational and inverse co-design, bioinspired pneumatic artificial muscles, and embodied intelligence.



Louis Sutter is currently a Research Scientist and the Head of the Berries and Medicinal Plants Research Group, Agroscope, Switzerland's centre of excellence for agricultural research. Based in Conthey, he leads applied research aimed at improving the sustainability, resilience, and productivity of horticultural production systems. His expertise encompasses integrated pest management, biological control, crop protection, and innovative berry production technologies. His research focuses on developing environmentally sustainable pest management strategies, enhancing the establishment and effectiveness of beneficial organisms, and evaluating alternative production systems, including agri-voltaics. He coordinates interdisciplinary research projects and collaborates closely with universities, international research institutions, industry partners, and growers to translate scientific findings into practical solutions. Through his work, he aims to bridge fundamental ecological understanding and agricultural practice, contributing to more sustainable and climate-resilient crop production.



Josie Hughes (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees from the Bio-Inspired Robotics Laboratory (BIRL), University of Cambridge. She is currently an Assistant Professor at EPFL, where she established the CREATE in 2021. Following this, she worked as a Post-Doctoral Associate at CSAIL, MIT, Distributed Robotics Laboratory. Her research focuses on developing novel design paradigms for designing robot structures that exploit their physicality and interactions with the environment. This includes the development of robotic hands, soft manipulators and locomoting robots, and automation systems for applications focused on sustainability and science.