

MORPHeus: a Multimodal One-armed Robot-assisted Peeling System with Human Users In-the-loop

Ruolin Ye, Yifei Hu, Yuhan (Anjelica) Bian, Luke Kulm, and Tapomayukh Bhattacharjee

Abstract—Meal preparation is an important instrumental activity of daily living (IADL). While existing research has explored robotic assistance in meal preparation tasks such as cutting and cooking, the crucial task of peeling has received less attention. Robot-assisted peeling, conventionally a bimanual task, is challenging to deploy in the homes of care recipients using two wheelchair-mounted robot arms due to ergonomic and transferring challenges. This paper introduces a robot-assisted peeling system utilizing a single robotic arm and an assistive cutting board, inspired by the way individuals with one functional hand prepare meals. Our system incorporates a multimodal active perception module to determine whether an area on the food is peeled, a human-in-the-loop long-horizon planner to perform task planning while catering to a user’s preference for peeling coverage, and a compliant controller to peel the food items. We demonstrate the system on 12 food items representing the extremes of different shapes, sizes, skin thickness, surface textures, skin vs flesh colors, and deformability. Check out the Morpheus project at <https://emprise.cs.cornell.edu/morpheus/>.

I. INTRODUCTION

According to a recent report, up to 24 million people aged 18 years or older need assistance with activities of daily living such as feeding, as well as instrumental activities of daily living such as meal preparation (peeling, cutting, cooking, etc.) [2]. While there is a significant amount of research involving feeding [3–11], cutting [12–15], and cooking [16–18], peeling is relatively underexplored. In this paper, we focus on robot-assisted peeling with a single-arm setup, to address the needs of people with mobility limitations who use wheelchair-mounted robot arms.

Conventionally, peeling is considered a bimanual task in which one arm stabilizes the food item while the other arm executes the peeling action [19]. Most real-world robotic caregiving systems [20] use one robot arm mounted on the wheelchair because mounting two robot arms on wheelchairs is practically challenging due to ergonomic and transfer challenges for care recipients [21]. We use an assistive cutting board inspired by how individuals with one functional hand prepare meals. An assistive cutting board [22], also known as an adaptive cutting board, is a board with multiple fixtures that can secure a food item (shown in Fig 1), so the user can peel with one arm. In addition, when users peel, they might have different preferences. For example, a user might want a half-peeled cucumber if they want to

This work was partly funded by National Science Foundation IIS #2132846, CAREER #2238792, and DARPA under Contract HR001120C0107. We thank Daniel Stabile and Mavis Lee for their help with the data collection portal. Videos and supplementary materials are available at our website [1].

All the authors are affiliated with Cornell University, Ithaca, NY, USA, {ry273, yh464, yb265, lbk73, tb557}@cornell.edu

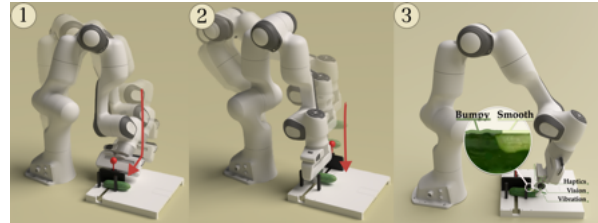


Fig. 1. Concept of robot-assisted peeling with an adaptive cutting board: (1, 2) The robot uses fixtures on the board to secure the food and interacts with the food, resulting in long-horizon manipulation. (3) The robot uses a peeler instrumented with multimodal sensors to identify whether an area is peeled.

prepare cucumber bites with some texture provided by the skin. We want to ensure that their preferences about peeling coverage (unpeeled, half-peeled, top-peeled, etc.) are taken into account.

Robot-assisted peeling presents numerous challenges due to the varying physical properties of food items, such as their shape, size, deformability, skin thickness, and skin-to-flesh color contrast. Using a single robot arm for peeling tasks with adaptive cutting boards introduces complex issues. The robot needs to reason how to use the complicated fixtures on the cutting board. Also, it needs to have a control policy that can peel a wide range of food items.

Our key insight is that multimodal perception with human-in-the-loop long-horizon planning and compliant control can enable a one-armed robot to perform the peeling task for a wide variety of food items while taking users’ preferences into consideration. Using this insight, we develop a system that includes: (1) a multimodal active perception module using visual, haptic, and vibration sensing modalities to estimate the peeling state of a food item during task execution; (2) a Large Language Model (LLM)-based interface to convert a user’s natural language commands to long-horizon plans with a human user in-the-loop; (3) a compliant controller for executing the peeling motion and adapting to a wide variety of food items with varying physical properties. Check out the MORPHeus system at <https://emprise.cs.cornell.edu/morpheus/>.

We summarize our contributions as follows:

- A one-armed robot-assisted peeling system with compliant control for peeling food items with a wide variety of physical properties and various assistive cutting boards.
- A multimodal active perception model incorporating visual, haptic, and vibration inputs for peeling.
- A natural language interface for generating long-horizon plans that cater to user preferences with human users in-the-loop to provide feedback for peeling.
- A dataset of human peeling actions on a wide variety

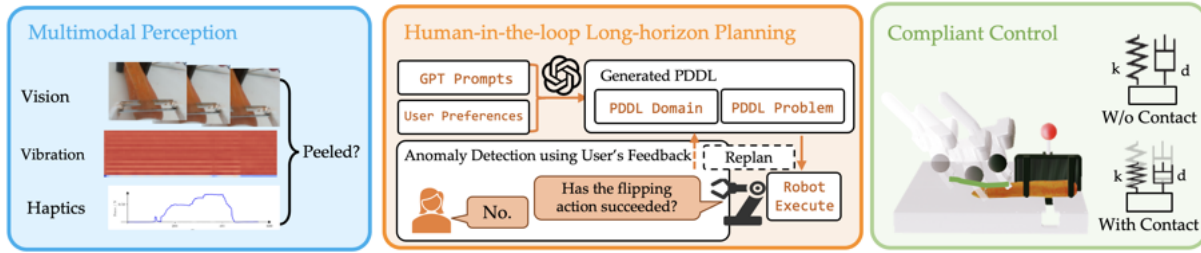


Fig. 2. Overview of the building blocks of MORPHeus: Multimodal perception (Sec. III-A) module uses visual, vibration, and haptic inputs to determine whether the food is peeled, Human-in-the-loop Long-horizon Planning (Sec. III-B) enables the robot to perform task planning while catering to a user's preference and perform anomaly detection using user's feedback, and Compliant Control (Sec. III-C) to execute peeling actions for various food items.

of food items and an open-hardware design of a peeler with sensors for visual, haptic, and vibration modalities.

II. RELATED WORK

A. Meal Preparation with Robots

Meal preparation can be broken down into multiple sub-steps, including cutting [12, 13], peeling [19], and cooking [16]. Xu et al. [12] developed a cutting system to cut food with solid cores, such as avocado. Liu et al. [16] proposed a model that learned stir-fry motions from human demonstrations. These works either assume that the food items have already been peeled or ignore the peeling problem. There is also research on peeling bananas with a dexterous hand [23], peeling grapes with a surgical robot [24], and peeling a grapefruit with a knife [25]. Watanabe et al. [26] and Dong et al. [19] used peelers for peeling, similar to our setup. However, they both used a dual-arm setup: one arm stabilizes the food items, and the other peels. Watanabe et al. [26] used 2D visual perception to determine whether the food was peeled and tested their system with a single food item. Dong et al. [19] added a point cloud to obtain 3D information and expanded the scope to 5 food items. Our work uses visual, haptic, and vibration modalities to determine whether the food item is peeled, takes a variety of assistive cutting boards into consideration, and performs experiments on 12 vegetables and fruits that cover a wide range of physical properties, with only one robot arm. Additionally, our work differs from previous work in the fact that we perform the peeling task based on user preferences about the peeling coverage.

B. Human-in-the-loop Long-horizon Planning

Language serves as a natural interface to incorporate human preferences into long-horizon planning [27, 28]. In recent years, there has been significant progress in large language models [29–36]. Given their extensive pre-training on diverse text data, LLMs exhibit impressive zero-shot generalization capabilities, making them suitable for use in robot planning [37–39]. Therefore, we use LLM to perform long-horizon planning with human-in-the-loop, taking their preferences into consideration. However, compared to directly generating task plans using LLMs, converting natural language to Planning Domain Definition Language (PDDL), and planning with a classical planner has the following advantages: (1) the PDDL generated by LLM can be quickly scrutinized; (2) executing a synthesized program is often

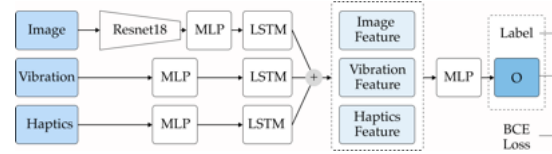


Fig. 3. Model structure for multimodal perception: The network uses image, vibration, and haptics as input, and gives a binary output for peeled/unpeeled classification (O).

quicker than repeatedly querying the LLM for each new job; (3) synthesized programs can be customized for tasks of any size; (4) using PDDL circumvents the context window size limitations inherent to LLMs [40]. We therefore use LLM integrated with PDDL for long-horizon planning. Previously, research has already focused on fine-tuning Large Language Models (LLMs) for the PDDL domain [41], as well as on translating natural language into PDDL [42–45]. These capabilities make LLMs well-suited for human-in-the-loop planning scenarios.

III. METHODOLOGY

MORPHeus has 3 building blocks: multimodal perception, long-horizon planning, and compliant control. The perception module informs the planner about the states of the peeler, the robot, the cutting board, and the food item. The planner performs human-in-the-loop long-horizon planning, integrating an LLM (GPT-4) with PDDL. For control, we use a Cartesian impedance controller. Fig.2 shows the system modules.

A. Multimodal perception

1) *Perception Model*: We introduce a binary classification model that takes an image, vibration, and force sequence as inputs. The three data modalities go through parallel feature extraction layers, including a network for images, multi-layer perceptron networks (MLPs), and LSTMs. These processed data streams are then concatenated and passed through additional dense layers to produce a binary final output for peeled/unpeeled (O). Given the model's output O and the human-labeled ground truth, we use the Binary Cross Entropy (BCE) as the loss function. We show the structure of the model in Fig. 3. Please refer to the details on our website [1].

2) *Multimodal Peeler*: We design a peeler equipped with multimodal sensors and release it as open hardware. Fig. 4(c) shows the multimodal peeler and its structural design. We use a Realsense D435 camera for visual input [46], a TE FX29K0-040B-0010-L load cell sensor [47] for force, and

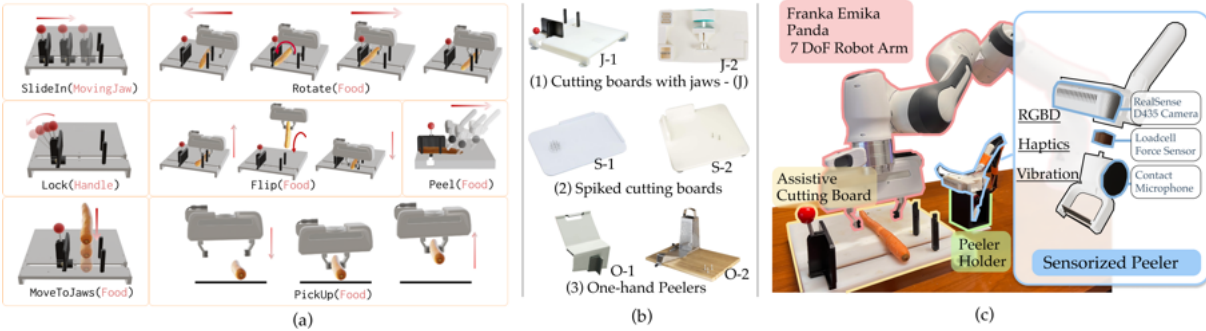


Fig. 4. (a) Actions the robot can take. For bidirectional actions including SlideIn/Out, Lock/Unlock, MoveToJaws/MoveFromJaws, we only visualize one direction here. (b) Various types of cutting boards. (c) The experiment setup includes a Franka Emika Panda 7 DoF Robot Arm, an assistive cutting board with a moving jaw, a 3D printed peeler holder, and a peeler with multimodal perception for vision, force, and vibration modalities.

a piezo contact microphone for vibration [48]. The force sensor is connected to an Arduino Uno Rev3 board [49] for readings. The contact microphone is connected to an amplifier and then to a desktop for data collection. The image signals are sampled at 30Hz, the force signals at 2.4kHz, and the vibration signals from the contact microphone at 1.6kHz. We assemble the sensors on the 3D-printed parts of the custom-designed peeler head and peeler holder, and mount a blade on the peeler similar to a standard peeler.

B. Human-in-the-loop Long-horizon Planning

We use GPT-4 as a foundation model for long-horizon human-in-the-loop planning. The role of GPT-4 is to generate the files for PDDL. In planning with PDDL, there are two critical components: a PDDL domain file and a PDDL problem file. The domain file categorizes objects in the environment into different types, specifies possible actions along with their preconditions and effects, and includes a set of binary condition predicates. The problem file outlines the initial state of the environment and the desired goal state. We introduce the planning module with necessary high-level information in this section. For the details, please refer to our website [1].

Prompt Specification: We provide GPT-4 with a structured prompt as follows. The prompt describes (a) the role of GPT, (b) the robot actions, (c) the environment, (d) the format of the output, (e) example of a customized prompt with the same structure with the generated domain PDDL, and (f) user inputs, such as peeling goals and feedback. We show an example of the prompt inside the gray box III-B-1. The prompt with these components gives GPT the context to reason for this specific peeling task, and also constraints its output syntax by giving examples of PDDL files. Prior research, such as [50, 51], has highlighted the proficiency of Large Language Models (LLMs) in utilizing human feedback to detect and correct anomalies in generated content. In our planning approach, we implement an **anomaly detection with user’s feedback** module to address potential execution errors. At critical junctures in the plan, this module will seek human feedback to detect or address potential anomalies. After completing the peeling of a certain area or a flip, the process pauses to ask the user some questions regarding whether the action has succeeded. For example, the robot may ask “Has the rotation action succeeded?” If the user identifies an anomaly, the LLM engages the user with a series of binary questions, such as “Is the food item between

the jaws?”, to update its understanding of the environment. Following this, the LLM replans to rectify the previous failure before proceeding, assume undo is possible.

Example Prompt (III-B-1): (a) Role of GPT You are an excellent interpreter of human instructions for care giving tasks... (b) Robot actions Pickup...PutDown... (c) Environment ...a robot arm with a gripper...an adaptive cutting board... to peel a food item that is on the table... (d) Output format The format of output should be a PDDL domain file... (e) Examples Below is a simple example of generating a PDDL involving non-deterministic action... (f) User inputs The user wants the potato fully peeled...

Action Specification: The peeling task involves a complex, long-horizon manipulation sequence that involves a variety of subtasks. Imagine a robot using a cutting board with a moving jaw (J-1 and J-2 in Fig. 4(b)) to peel a carrot: the robot first slides in the moving jaw and locks it to secure the food item. Then, it peels the food. After peeling, it unlocks the moving jaw, pulls out the fixture, and then rotates the food to proceed to the next peeling action. If the robot uses the spiked cutting board (S-1 and S-2 in Fig. 4(b)), it puts the carrot on the spikes, and then peels the carrot. The subtasks for peeling are different for different cutting boards. For these subtasks, we combine possible actions across three different types of cutting boards to create an action space. The action space includes a description of the cutting boards and the accompanying actions. This enables the LLM to mix and match action sets for planning when multiple types of cutting boards are available. We illustrate actions specific to the cutting board with jaws, as used in our experimental setting, in Fig.4(a). For a detailed implementation of each action and the actions for other cutting boards, please refer to our website [1].

C. Compliant Control

To ensure effective peeling without damaging the flesh of food items, the robot needs to exert the appropriate amount of force on the uneven food surfaces. We achieve this by using a Cartesian impedance controller that dynamically adjusts the applied force when peeling. Let $\mathbf{q} \in \mathbb{R}^7$ be joint positions (7 DoF robot arm), $\dot{\mathbf{q}} \in \mathbb{R}^7$ be joint velocities, $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^7$ be Coriolis torque, and $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times 7}$ be the Jacobian matrix. We define the translational stiffness as k_t and rotational stiffness

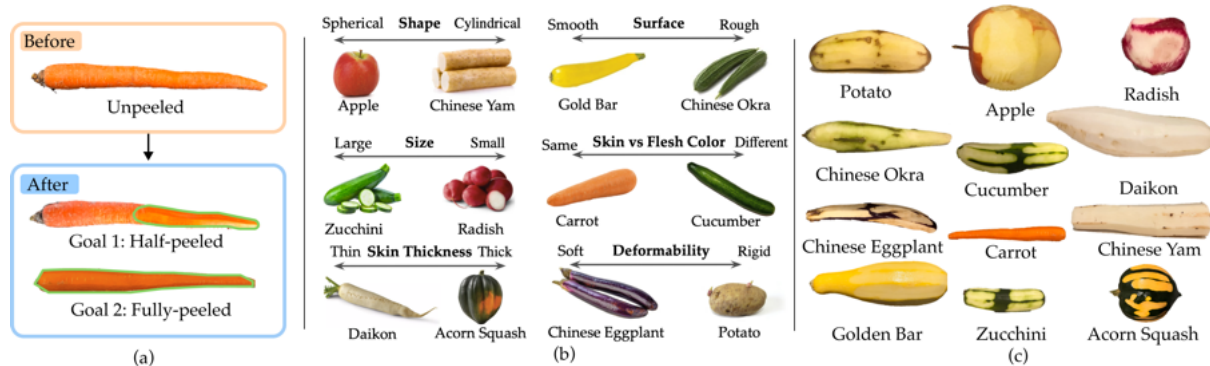


Fig. 5. (a) Users' preference: By expressing their preference for the food to be 'Half-peeled' or 'Fully-peeled', the users can get their food peeled according to their preferred peeling goal. (b) 6 axes corresponding to the characteristics of different food items and 12 food types selected to represent the extremes. (c) Food items peeled by MORPHeus.

as k_r , with which we can construct the stiffness matrix K and damping matrix D as the following assume critical damping:

$$K = \begin{bmatrix} k_l I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & k_r I_3 \end{bmatrix}, D = \begin{bmatrix} 2\sqrt{k_l} I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 2\sqrt{k_r} I_3 \end{bmatrix},$$

where I_3 is the 3×3 identity matrix.

Let $\mathbf{x} = FK(\mathbf{q}) \in \mathbb{R}^6$ be the task space pose, where $FK(\cdot)$ is the forward kinematics module, and $\mathbf{x}_d \in \mathbb{R}^6$ be the desired task space pose. \mathbf{f}_{ext} is the desired external force in the task space. With these definitions, we use the control law in Eq. 1 to obtain the desired force:

$$\mathbf{f}_{\text{ext}} = K(\mathbf{x}_d - FK(\mathbf{q})) + D(J\dot{\mathbf{q}}) \quad (1)$$

IV. SYSTEM COMPONENT EVALUATIONS

We evaluate each of the system components separately with a combination of simulations and real-world experiments.

A. Multimodal Perception

1) *Experiment setup*: We evaluate the multimodal perception module using the setup in Fig. 4(c). We use a Franka Emika Panda 7DoF robot arm and put a cutting board in front of it. The cutting board is fixed to the table using suction cups. We also design a 3D-printed holder for the peeler, and fix it to the table.

a) *Food selection*: To ensure food selection with a wide variety of physical properties, we consider the following characteristics: (1) shape, (2) surface texture, (3) size, (4) contrast of color between skin and flesh, (5) skin thickness, and (6) deformability. For each of them, we select two food items that represent the extremes, yielding a total of 12 distinct food items with significantly varying characteristics. We show the 12 food items and their different characteristics in Fig. 5(b).

b) *Data Collection Procedure*: We collect the data for multimodal perception by performing a set of haptic exploratory actions where the robot uses a set of specific *action primitives* to figure out the haptic properties of the food items. The *action primitives* include sliding over the unpeeled surface, peeling over the unpeeled surface, sliding over the peeled surface, and peeling over the peeled surface. For sliding, we turn the peeler and use the back of the blade to slide over the surface of the food item. We identify these actions as potential ways a robot can use to interact with the

food and identify whether it is peeled or not. The purpose of this experiment is to find the most suitable actions through the haptic exploratory procedure. We collect 100 samples, where 1 sample contains 4 action primitives. There are a total of $100 \times 4 \times 12 = 4800$ trials across all of the food items.

We evaluate the perception module by simulating a real-world setup with the dataset. We stream the data for each food item 10 times and calculate the classification success rate. We set the input length of the LSTM to 10 considering the speed of the robot. For initialization, we repeat the first frame for 10 times. As new frames come in, we append them to the input buffer.

Based on the experiment results in Table I, we realize that the sliding motion gives a better outcome for classifying peeled/unpeeled areas. While the visual modalities remain the same, the key difference between sliding and peeling lies in the source of vibration and force. When peeling, the front of the blade makes contact with the food item, and the blade can move freely. While sliding, the blade reaches its kinematic limit and becomes rigidly locked. As a result, less noise is generated from sliding, allowing for subtle vibrations to be more easily detected. This motivates us to use the sliding motion to tell whether the food item is peeled.

B. Human-in-the-loop Long-horizon Planning

To evaluate the human-in-the-loop long-horizon planning module, we utilize GPT-4's Chat Completion API, a context-aware text-completion API. We refer to this as 'simulation'.

The planner's goal in this simulated experiment is to find a sequence of actions based on the domain PDDL files to transition an initial state into a goal state described in the problem PDDL files. Since the planner uses both the domain and problem PDDL files, errors in either one of them might lead to errors. We evaluate the following cases: (1) GPT-generated PDDL domain files (GPT-Domain) with PDDL problem files written by human experts (Fixed-Problem), (2) GPT-generated PDDL problem files (GPT-Problem) with PDDL domain files written by human experts (Fixed-Domain), and (3) GPT-generated PDDL domain and problem files together (GPT-Domain & GPT-Problem).

We identify three common starting states for food items in the problem PDDL file: unpeeled, half-peeled, and top-peeled (where only the part facing the peeler is peeled). The

TABLE I
EVALUATING THE MULTIMODAL PERCEPTION AND CARTESIAN
IMPEDANCE CONTROL COMPONENTS

Food name	Perception		Control		
	Peel	Slide	l	m	h
Cucumber	7/10	9/10	2/10	10/10	10/10
Carrot	5/10	8/10	1/10	9/10	10/10
Chinese Yam	9/10	9/10	6/10	10/10	10/10
Chinese Okra	6/10	7/10	2/10	5/10	10/10
Chinese Eggplant	7/10	10/10	0/10	7/10	10/10
Potato	6/10	10/10	0/10	2/10	10/10
Zucchini	8/10	9/10	3/10	10/10	10/10
Gold Bar	5/10	7/10	2/10	8/10	10/10
Daikon	6/10	7/10	3/10	9/10	10/10
Apple	7/10	8/10	2/10	9/10	7/10
Radish	6/10	4/10	3/10	2/10	0/10
Acorn Squash	4/10	5/10	1/10	0/10	5/10

Peel & Slide columns: Success rates of the perception module; l , m , h columns: success rates for low, medium, and high stiffness impedance controllers respectively.

peeling goal is fully peeled. We generate 35 domain PDDL files and 35 problem PDDL files to ensure the simulation is repeated an adequate amount of times, under each starting state using the previously proposed natural language prompt template. We use a planner with non-deterministic states, PRP (Planner for Relevant Policies) [52], to find the planning policies. We run PRP for 100 rounds of simulations for each generated file, with the three different start states. During these simulations, the PRP uses the planning policy to generate plans across various simulated scenarios.

In our PDDL formulation, actions like `Peel`, `Flip`, and `Rotate` might fail in the real-world setup, therefore we treat them as non-deterministic actions. For these actions, the planner selects the subsequent states uniformly at random. For PDDL generation, we use the cutting board (J-1 and J-2). We refer to trials that can run directly without any human intervention to fix syntax errors as No Attention Given (NAG).

We show the results in Table II. The results suggest the GPT-generated domain and problem PDDL files can run robustly without any syntax errors, and are able to perform the long-horizon planning task for peeling using the cutting board with the moving jaw (J-1, J-2). For the other cutting boards (S-1, S-2, O-1, and O-2), we inform GPT of the cutting board action correspondence, allowing it to potentially reason how to use other cutting boards.

C. Compliant Control

To evaluate the compliant module, we use the same setup in the real world as the multimodal perception module. We evaluate how the peeling action works on the food items. We let the robot perform the peeling action with the specified end position, and manually rotate the food item after each peeling action. We repeat this 10 times for each food item. The trial is considered successful if the skin is separated from the flesh completely. We tune 3 controllers for the food items, namely low-stiffness compliance (l) with $k_t = 120$ and $k_r = 0$, medium-stiffness (m) compliance with $k_t = 150$ and $k_r = 5$, high-stiffness compliance (h) with $k_t = 180$ and $k_r = 10$. We show the results in Table I. The results suggest that for most of the food items, the stiffness of the controllers matters. While high-stiffness works for most food items, medium-stiffness or low-stiffness controllers work better for softer or

TABLE II
EVALUATING THE PLANNING COMPONENT

Start states	NAG ¹	Avg. Action	Max. Action	Min. Action
GPT-Domain & Fixed-Problem ²				
Unpeeled	28/35	30.96±15.23	33.62±9.81	28.70±14.31
Top Peeled	28/35	24.54±12.45	26.39±7.30	28.36±11.32
Half Peeled	28/35	22.58±8.19	26.44±7.21	20.97±8.64
Fixed-Domain & GPT-Problem ³				
Unpeeled	31/35	30.84±14.56	33.21±11.41	28.39±13.10
Top Peeled	31/35	28.47±13.22	29.57±12.96	24.73±13.29
Half Peeled	31/35	24.12±8.48	28.65±11.30	20.78±9.06
GPT-Domain & GPT-Problem ⁴				
Unpeeled	27/35	29.13±11.42	32.77±9.08	28.47±16.94
Top Peeled	27/35	26.57±10.20	28.04±13.26	22.98±13.83
Half Peeled	27/35	24.31±10.04	28.79±8.92	22.57±12.96

¹NAG (No Attention Given): Simulation runs successfully without human intervention for syntax errors. ²GPT-Domain & Fixed-Problem: GPT-generated domain PDDL, human-written problem PDDL. ³Fixed-Domain & GPT-Problem: Human-written domain PDDL, GPT-generated problem PDDL. ⁴GPT-Domain & GPT-Problem: GPT-generated domain PDDL, GPT-generated problem PDDL.

spherical food items. We select the controller with the best performance in terms of peeling success rate for each of the food items. If there are two controllers with the same performance, we select the more compliant one to avoid potential damage to the food items.

V. FULL AUTONOMOUS SYSTEM (MORPHEUS) EVALUATION

After evaluating the components individually, we evaluate the overall performance of MORPHEUS by combining them together and autonomously running the whole system. We evaluate it based on the following criteria:

- Time taken to finish the task (t): The current State-of-the-art (SOTA) method [19] takes approximately 60 minutes to peel a single food item using a dual-arm setup. We record the time taken to peel the food item in minutes and divide it by the time of the SOTA method. $t = t_a/t_s$, where t_a stands for actual time, and t_s is the SOTA time [19] in minutes.
- Peeled area ratio (p): We define p as $p = a_p/a_a$ where a_p represents the area of the peeled region, and a_a represents the area of the entire region of the food surface. For simplicity, we take 2 photos, one for the front and the other for the back of the food item, ensuring as much area as possible is visible in the image. We then calculate a_p, a_a in the pixel space by manually annotating the contour of the peeled area and the entire visible area of the food item.
- Percentage of failed subtasks (n_h): In the long-horizon sequence, we consider a subtask as failed if it fails after trying 3 times. If this happens, the robot will ask the human caregiver to give it assistance to complete the subtask. We record n_h based on the percentage of failed subtasks out of an average number of subtasks of 50.

Table III shows the results. We noticed that the cylindrical food items have a higher success rate and a peeled area ratio, and take less time to finish. The spherical ones are easier to slip out of the jaws of the cutting board due to the design of the cutting board with a moving jaw we use (J-1), resulting in a lower success rate and the peeled area ratio, also taking more time to finish.

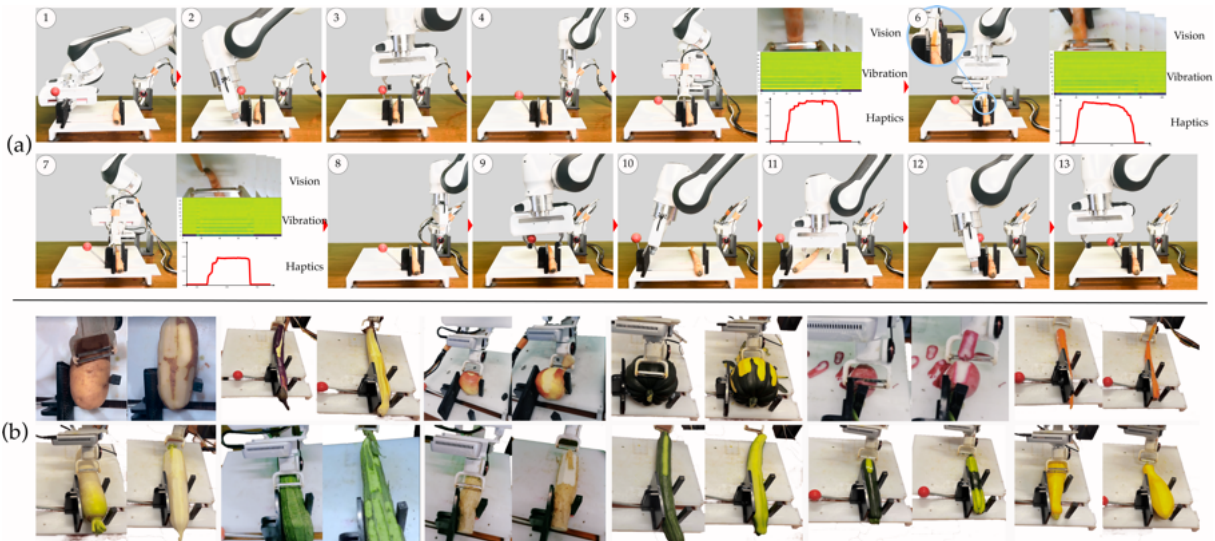


Fig. 6. (a) An example robot execution sequence for carrot peeling. (b) 12 food items at the beginning stage of peeling and at the final stage of peeling.

TABLE III
OVERALL SYSTEM EVALUATION

Food name	t	$p(\%)$	$n_f(\%)$
Cucumber	0.3	93.4	6
Carrot	0.27	90.1	4
Chinese Yam	0.33	89.7	4
Chinese Okra	0.38	88.1	2
Chinese Eggplant	0.33	87.6	6
Potato	0.57	85.7	12
Zucchini	0.3	84.2	8
Gold Bar	0.33	83.7	10
Daikon	0.27	82.7	24
Apple	0.46	76.8	24
Radish	0.58	53.2	22
Acorn Squash	0.72	46.3	33



Left: Overall system evaluation results. T column: Ratio of Time taken/SOTA time to complete the task, p column: Peeled area ratio, n_f column: Percentage of failed subtasks out of an average of around 50 subtasks. **Right:** Robot-peeled acorn squash compared with human-peeled acorn squash. The human uses only the dominant hand with the same peeler and cutting board as the robot for fair comparison.

VI. DISCUSSION

Based on the autonomous system evaluation, we identified the following points that can potentially help increase the success rate for future improvements, shown in Fig. 7.

a) Using suitable cutting boards and peelers based on the shape of food items (Fig. 7 Blue): The unique groove pattern on acorn squashes complicates the peeling task when using conventional peelers. As illustrated in Table III, both the human and the robot are not able to reach the grooves. In this setup, the human uses one arm to peel with the assistive cutting board J-1 for fair comparison. Alternative approaches, such as using a narrower peeler, pre-cutting along the grooves, or using a knife, could be effective for a more thorough peeling. The design of the cutting board also presents challenges for handling food items that are either too large or too small. Items that are too large may slip out if their widest dimension exceeds the jaw capacity, while small or spherical items risk evading the jaw's grasp altogether. We recommend using a larger assistive cutting board to accommodate larger food items, or a spiked cutting board might also work. The size can vary significantly even within the same category of food items. Items that are too

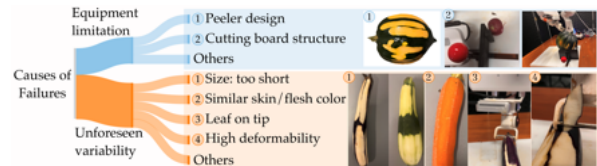


Fig. 7. Causes of failures: both equipment limitations and unforeseen variabilities may result in robot failure.

small may not be completely peeled as a result of the cutting board's dual-peg design, which could obscure the middle section before or after flipping. Using the spiked cutting board might help address this problem.

b) Accounting for more variabilities (Fig. 7 Orange): We identify that there are other variabilities in food items. The presence of thin skin strips may impact the robot's perception, which could be corrected through user feedback. However, in cases like carrots, where the skin and flesh colors are similar, even humans may not be able to tell the difference. Occasionally, the eggplant peels may not completely detach, confusing the perception algorithm. Implementing a perturbation module that shakes the food and peeler could help remove the peels. Eggplant leaves on the tip can also block the peeler. Implementing an adaptive impedance controller with online compliance updates could mitigate these issues, and we leave this open for future work.

c) Performing anomaly detection using user's feedback more intelligently: Our experiments demonstrate that this feedback allows the robot to identify and correct execution errors by providing feedback, avoiding more substantial failures later in the plan. Limiting the feedback to yes-or-no questions has the potential to lower the cognitive workload on users with mobility limitations, compared to letting users provide detailed instructions or feedback to the robot. However, other querying types with more details, such as annotating the contour of peeled/unpeeled areas, might provide the robot with more information. For future work, a query module that can intelligently decide the query timing and type has the potential to improve success rates and lower users' cognitive workload.

REFERENCES

- [1] “Morpheus: a multimodal one-armed robot-assisted peeling system with human users in-the-loop.” <https://emprise.cs.cornell.edu/morpheus/>, 2023. Accessed: 2024-03-06.
- [2] U. S. C. Bureau., “Americans with disabilities: 2014,” 2014.
- [3] E. K. Gordon, X. Meng, M. Barnes, T. Bhattacharjee, and S. S. Srinivasa, “Learning from failures in robot-assisted feeding: Using online learning to develop manipulation strategies for bite acquisition,” *CoRR*, vol. abs/1908.07088, 2019.
- [4] D. Gallenberger, T. Bhattacharjee, Y. Kim, and S. S. Srinivasa, “Transfer depends on acquisition: Analyzing manipulation strategies for robotic feeding,” in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 267–276, 2019.
- [5] R. Feng, Y. Kim, G. Lee, E. K. Gordon, M. Schmittle, S. Kumar, T. Bhattacharjee, and S. S. Srinivasa, “Robot-assisted feeding: Generalizing skewering strategies across food items on a realistic plate,” 2019.
- [6] R. Jenamani, D. Stabile, Z. Liu, A. Anwar, K. Dimitropoulou, and T. Bhattacharjee, “Robot-assisted inside-mouth bite transfer using robust mouth perception and physical interaction-aware control,” in *ACM/IEEE International Conference on Human Robot Interaction*, 2024.
- [7] R. Ye, W. Xu, H. Fu, R. K. Jenamani, V. Nguyen, C. Lu, K. Dimitropoulou, and T. Bhattacharjee, “Rcareworld: A human-centric simulation world for caregiving robots,” *IROS*, 2022.
- [8] J. Ondras, A. Anwar, T. Wu, F. Bu, M. Jung, J. J. Ortiz, and T. Bhattacharjee, “Human-robot commensality: Bite timing prediction for robot-assisted feeding in groups,” in *2022 SoCal Robotics Symposium*, 2022.
- [9] T. Xu, T. Zhao, J. G. Cruz-Garza, T. Bhattacharjee, and S. Kalantari, “Evaluating human-in-the-loop assistive feeding robots under different levels of autonomy with vr simulation and physiological sensors,” in *14th International Conference on Social Robotics*, 2022.
- [10] T. Bhattacharjee, E. Gordon, R. Scalise, M. Cabrera, A. Caspi, M. Cakmak, and S. Srinivasa, “Is more autonomy always better? exploring preferences of users with mobility impairments in robot-assisted feeding,” in *ACM/IEEE International Conference on Human-Robot Interaction*, 2020.
- [11] R. Feng, Y. Kim, G. Lee, E. Gordon, M. Schmittle, S. Kumar, T. Bhattacharjee, and S. Srinivasa, “Robot-assisted feeding: Generalizing skewering strategies across food items on a plate,” in *International Symposium on Robotics Research*, 2019.
- [12] Z. Xu, Z. Xian, X. Lin, C. Chi, Z. Huang, C. Gan, and S. Song, “Roboninja: Learning an adaptive cutting policy for multi-material objects,” 2023.
- [13] E. Heiden, M. Macklin, Y. Narang, D. Fox, A. Garg, and F. Ramos, “Disect: A differentiable simulation engine for autonomous robotic cutting,” 2021.
- [14] X. Mu, Y. Xue, and Y.-B. Jia, “Robotic cutting: Mechanics and control of knife motion,” *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3066–3072, 2019.
- [15] H. Fu, W. Xu, R. Ye, H. Xue, W. Du, Z. Yu, T. Tang, Y. Li, C. Lu, and J. Zhang, “Demonstrating rfuniverse: A multiphysics simulation platform for embodied ai,” in *Proceedings of Robotics: Science and Systems (RSS 2023)*, (Daegu, Republic of Korea), 2023.
- [16] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen, “Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects,” May 2022.
- [17] J. sup Yi, T. A. Luong, H. Chae, M. S. Ahn, D. Noh, H. N. Tran, M. Doh, E. Auh, N. Pico, F. Yumbala, D. W. Hong, and H. Moon, “An online task-planning framework using mixed integer programming for multiple cooking tasks using a dual-arm robot,” *Applied Sciences*, 2022.
- [18] K. Takata, T. Kiyokawa, I. G. Ramirez-Alpizar, N. Yamanobe, W. Wan, and K. Harada, “Efficient task/motion planning for a dual-arm robot from language instructions and cooking images,” *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12058–12065, 2022.
- [19] C. Dong, L. Yu, M. Takizawa, S. Kudoh, and T. Suehiro, “Food peeling method for dual-arm cooking robot,” in *2021 IEEE/SICE International Symposium on System Integration (SII)*, pp. 801–806, Jan. 2021.
- [20] T. Bhattacharjee, E. K. Gordon, R. Scalise, M. E. Cabrera, A. Caspi, M. Cakmak, and S. S. Srinivasa, “Is more autonomy always better? exploring preferences of users with mobility impairments in robot-assisted feeding,” in *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 181–190, 2020.
- [21] Kinova Robotics, “JACO Robotic Arm - Assistive Technologies.” <https://assistive.kinovarobotics.com/product/jaco-robotic-arm>, 2024. Accessed: 2024-03-28.
- [22] R. by USA Today, “Etac one-handed cutting board review: Customized adaptation beyond food.” <https://tinyurl.com/a-cutting-board>, Accessed on 2024-03-20.
- [23] H. Kim, Y. Ohmura, and Y. Kuniyoshi, “Goal-conditioned dual-action imitation learning for dexterous dual-arm robot manipulation,” *IEEE Transactions on Robotics*, pp. 1–20, 2024.
- [24] “Precision da vinci robotic surgery peels grape skin.” www.youtube.com/watch?v=cpPofyZbvDw, [Online; Accessed 11th January, 2024].
- [25] A. Straižys, M. Burke, and S. Ramamoorthy, “Surfing on an uncertain edge: Precision cutting of soft tissue using torque-based medium classification,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4623–4629, 2020.
- [26] Y. Watanabe, K. Nagahama, K. Yamazaki, K. Okada, and M. Inaba, “Cooking behavior with handling general cooking tools based on a system integration for a life-sized humanoid robot,” *Paladyn*, vol. 4, Jan. 2013.
- [27] Z. Zhao, W. S. Lee, and D. Hsu, “Large language models as commonsense knowledge for large-scale task planning,” 2023.
- [28] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, “Tidybot: Personalized robot assistance with large language models,” 2023.
- [29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019.
- [30] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, “Evaluating large language models trained on code,” *CoRR*, vol. abs/2107.03374, 2021.
- [31] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, “Opt: Open pre-trained transformer language models,” 2022.
- [32] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” 2020.
- [33] OpenAI, “Gpt-4 technical report,” 2023.
- [34] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, “Training language models to follow instructions with human feedback,” 2022.
- [35] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, “Llama: Open and efficient foundation language models,” 2023.
- [36] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, “Palm: Scaling language modeling with pathways,” 2022.
- [37] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents,” *CoRR*, vol. abs/2201.07207, 2022.
- [38] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay,

- D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," 2022.
- [39] S. Vemprala, R. Bonatti, A. Buckner, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," 2023.
- [40] T. Silver, S. Dan, K. Srinivas, J. Tenenbaum, L. Kaelbling, and M. Katz, "Generalized planning in PDDL domains with pretrained large language models," *arXiv preprint*, 2023.
- [41] V. Pallagani, B. Muppasani, K. Murugesan, F. Rossi, L. Horesh, B. Srivastava, F. Fabiano, and A. Loreggia, "Plansformer: Generating symbolic plans using transformers," 2022.
- [42] K. M. Collins, C. Wong, J. Feng, M. Wei, and J. B. Tenenbaum, "Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks," 2022.
- [43] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: From natural language instructions to feasible plans," 2023.
- [44] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh, "Translating natural language to planning goals with large-language models," 2023.
- [45] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "Llm+p: Empowering large language models with optimal planning proficiency," 2023.
- [46] "Realsense d435i." <https://www.intelrealsense.com/depth-camera-d435i/>, [Online; Retrieved on 16th September, 2023].
- [47] "Loadcell force sensor." <https://www.te.com/usa-en/product-20009605-16.html>, [Online; Retrieved on 16th September, 2023].
- [48] "Piezo contact microphone pickup." <https://tinyurl.com/contact-microphone>, [Online; Retrieved on 16th September, 2023].
- [49] "Arduino uno rev3." <https://docs.arduino.cc/hardware/uno-rev3>, [Online; Retrieved on 16th September, 2023].
- [50] Z. Liu, A. Bahety, and S. Song, "Reflect: Summarizing robot experiences for failure explanation and correction," 2023.
- [51] Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, N. Duan, and W. Chen, "Critic: Large language models can self-correct with tool-interactive critiquing," 2023.
- [52] C. Muise, V. Belle, and S. A. McIlraith, "Computing contingent plans via fully observable non-deterministic planning," in *The 28th AAAI Conference on Artificial Intelligence*, 2014.