A Perception-Manipulation Robotics System for Food Cutting

Xinyuan Luo[1], Wenzhen Yuan[1]

***Abstract*— In the development of cooking robots, mastering the task of cutting is crucial. A significant challenge lies in the diverse properties of food, which necessitate distinct cutting policies and even different knives for optimal processing. This paper presents a perception-manipulation framework for food-cutting tasks. Our system features a knife selection module that utilizes force data from a preliminary fixed trial cut to select the appropriate knife for the given food. This is followed by an adaptive cutting phase using reinforcement learning (RL) to balance cutting speed and energy efficiency. In our experiments, the knife selection module achieved 100% successful rate on unseen food, and we compared the performances of fixed policy, RL policy, with human operators. Our method not only achieves high performance but also demonstrates comparable results to those of human participants.**

I. Introduction

The development of cooking robots has been a key interest among researchers and industry practitioners for many years. The comprehensive cooking process includes food preparation tasks such as cutting [1], peeling [2], and stirring [3], along with various cooking techniques like flipping [4] and stir-frying [5]. Most of these sub-tasks present significant challenges due to the diverse properties and irregular shapes of food, as well as the delicate manipulation skills required.

Cutting, a critical process in food preparation, necessitates a thorough understanding of food properties such as hardness and friction coefficient. While humans have an innate sense of these properties before cutting, they also adapt their cutting techniques in real-time based on the force-torque feedback received from knife contact. This adaptation control can be achieved by learning a dynamic model for model predictive control (MPC) [1], [6], [7], employing preset control algorithms [8]–[10], and utilizing reinforcement learning (RL) [11], [12].

In this work, we aim to develop a food-cutting system that generalizes to various types of unseen foods. Similar to humans, our robot initially interacts with the food to assess its properties. During the cutting process, the robot can dynamically adjust its cutting strategy based on force feedback. Fig. 1 provides an overview of the proposed system. To expand the variety of foods it can handle, we have equipped the robot with an easy-to-switch knife mount that allows seamless transitions between a fruit knife and a serrated knife. As shown in Fig 2, the whole process begins with a fixed trial cut to collect force data, which is then used to determine the appropriate knife. Then, we have designed a reward system to measure cutting efficiency and employed an on-policy RL agent to online refine the cutting motions.

[1]Xinyuan Luo and Wenzhen Yuan are with the University of Illinois at Urbana-Champaign {xl153, yuanwz}@illinois.edu

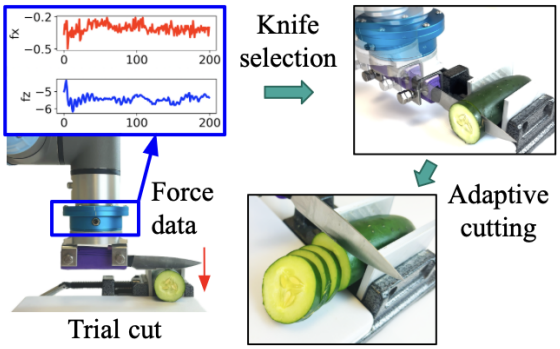

Fig. 1: We developed a food cutting system that first performs a fixed trial cut motion to the food to understand the food properties and select a proper knife. Following this, the food is cut using an RL adaptive controller.

A core challenge in food cutting stems from the diverse properties of food, such as hardness, surface friction coefficient, and juiciness. For instance, chopping is unsuitable for baguette due to its deformable nature; instead, a serrated knife and a sawing motion are required. Additionally, cutting juicy foods like tomatoes too hard may lead to damage due to their delicate structure [13], [14]. To tackle this challenge, we designed an easy-to-switch knife mount shown in Fig 4(c) that could effectively switch between a fruit knife and a serrated knife. Utilizing the information obtained from the trial cut, the system can choose a proper knife and perform sawing or chopping.

Another significant challenge is the convergence of the adaptive reinforcement learning controller. To expedite the convergence of the RL algorithm, we carefully design the state space, action space, and reward function. We then develop an initial value selector to accelerate the convergence. These approaches enable our system to achieve convergence within the cutting of just 2 to 3 slices.

In summary, the key contributions of this work are 1) A trial cut motion that selects the appropriate knife and cutting policy, achieving 100% accuracy with 4 types of unseen food. 2) An adaptive controller that is optimized via real-world reinforcement learning, capable of rapidly generalizing to new types of food and outperforming fixed policies while delivering results comparable to human performance. 3) A comprehensive robotic cutting pipeline that integrates knife and policy selection with adaptive cutting.

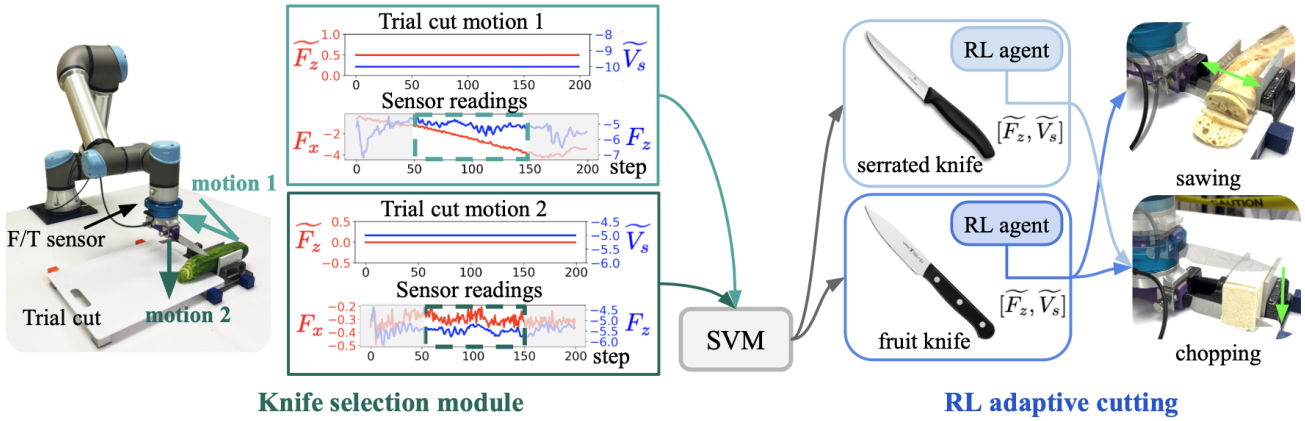

Fig. 2: Pipeline of our Perception-Manipulation Robot Cutting System for an arbitrary food item: the robot first conducts two trial cut motions (defined as fixed target vertical force $\widetilde{F_z}$ and a fixed sawing speed $\widetilde{V_s}$) and uses the measured force value F_x and F_z to decide the type of knife to use. Subsequently, the robot with the selected knife, will execute cutting tasks using the policy $[\widetilde{F_z}, \widetilde{V_s}]$ output by the pre-trained reinforcement learning (RL) agent.

II. Related Works

A. Robotics cutting

Researchers tackle the cutting task using either control-based methods [8]–[10] or learning-based methods.

Due to the significant effort required to tune control parameters for foods with varying properties, researchers have turned to model predictive control (MPC). Lenz et al. [1] introduces DeepMPC, which employs a novel deep neural network structure to learn the dynamics of various foods. Similarly, Mitsion et al. [7] utilizes an LSTM network to learn the intrinsic dynamics, and their another work [6] employs a comparable learning and control framework. However, these approaches generally require the collection of large datasets to accurately learn the dynamic models of various foods.

Dynamic motion primitives (DMP) [15] represent another approach to robotic cutting [16]. Zhang et al. [17] employs DMP as a behavior cloning method to learn from human demonstrations of cutting various foods, and also to generalize to unseen food types. Yang et al. [18] also incorporates cutting tasks into their experiments.

Reinforcement learning offers promising directions for robotic cutting. Padalkar et al. [11] employs Policy learning by Weighting Exploration with the Returns (PoWER), learning parameters for an impedance controller in downward motions combined with a fixed sawing motion to effectively cut cucumbers and bananas. Meanwhile, Beltran-Hernandez et al. [12] leverages the DiSECt simulation engine [19], specifically designed for cutting tasks. This approach models food as a spring, uses a trial cut to determine its parameters, and learns cutting skills through simulation before transferring them to a real robot. In contrast, our trial cut is not used to explicitly estimate physical parameters for simulation. Instead, we use force features from trial cuts to select the knife and initialize a real-robot cutting policy, and then use on-policy reinforcement learning to fine-tune the target vertical force and sawing speed during execution. This positions our method as a real-world perception-manipulation pipeline rather than a parameter-identification step for simulation-based training.

B. Perception of food properties

Conducting food manipulation tasks necessitates an understanding of food properties. Numerous researchers have explored this area. For example, Mu et al. [20] propose a recursive least-squares method to estimate relevant physical parameters, such as Poisson's ratio, fracture toughness, and the coefficient of friction, using only force sensor readings. Sawhney et al. [21] introduce a multimodal sensory approach that combines vision, finger vision, and audio to learn embeddings that represent food type, juiciness, hardness, slice type, and width effectively. They also use multimodel perception to classify the current state of a cutting system, for example, the knife contacting the food or scraping the cutting board [17]. Similarly, Ye et al. [2] utilizes vision, vibration, and force to determine whether food has been peeled. As a preliminary study to DeepMPC, Gemici et al. [22] investigates various actions using different tools to learn about food properties such as hardness, plasticity, elasticity, tensile strength, brittleness, and adhesiveness. Our method is related to these probing-based approaches, but it uses the trial interaction as a compact decision signal for knife selection and policy initialization, without requiring an explicit estimate of material parameters.

III. Method

To address the challenge of adaptively cutting any unseen food, we constructed a low-level controller with two parameters: the target force in the vertical direction ($\widetilde{F_z}$) and the sawing speed ($\widetilde{V_s}$). By tuning these parameters, the robot can perform chopping or sawing at various speeds, adapting to foods with different physical properties. To expand the variety of foods the robot can successfully cut, we developed a knife selection module that enables the robot to switch between a fruit knife and a serrated knife. To decide which knife to use, the robot performs a set of trial cuts with

predefined motions to collect force data. This data is then used to solve a binary classification problem to select the appropriate knife. We then implemented an RL adaptive controller to balance cutting speed and energy efficiency, with a reward function that encourages efficient cutting.

A. Low-level cutting controller

Both the trial and main cutting motions are controlled by the force ($\widetilde{F_z}$) in the z-direction (vertical) and the velocity ($\widetilde{V_s}$) in the x-direction (sawing). During the motion of cutting a single slice, the knife's movement is confined to the x-z plane. The knife is attached to a force/torque (F/T) sensor mounted at the robot's terminal. The food is fixed by a sliding vice. Fig 4(a) depicts the setup in a real-world scenario.

On the z-axis, the knife is controlled by a PID force controller to apply down force. The control output, $u(t)$, the output velocity of the robot terminal in vertical direction, is defined as: $u(t) = K_p \cdot e(t) + K_i \cdot \int e(t)\,dt + K_d \cdot \frac{d}{dt} e(t)$, where $e(t) = F_z - \widetilde{F_z}$ defines the error between the measured force F_z and the target vertical force $\widetilde{F_z}$. The parameters K_p, K_i, and K_d represent the proportional, integral, and derivative gains, respectively.

On the x-axis, the sawing motion of each back-and-forth movement is governed by a position controller that tracks the commanded sawing speed, with the displacement constrained within a range of ± 2 cm.

B. Knife selection module

Intuitively, a fruit knife, with its flatter blade, typically produces flatter slices indicative of superior cutting quality; However, we observed challenges or failure when using a fruit knife to cut hard foods like potatoes and sweet potatoes, as well as deformable and tough foods such as baguettes and other breads. In contrast, a serrated knife more easily cuts such foods. Therefore, a knife selection module is essential.

To classify food with those features and select a proper knife, we designed a trial cut to perform a pre-defined motion on food using the fruit knife. Utilizing force data obtained from the trial cut, we can decide whether to continuously use the fruit knife or switch to a serrated knife. To find the best trial cut policy, we collected force, velocity, and displacement data across 14 different foods using 6 sets of trial cut parameters. Our analysis revealed that the force series data in sawing and vertical directions are the most informative. Experiments detailed in Section V-A demonstrated that a combination of fixed sawing and chopping motions most effectively captures the properties of the food, resulting in optimal knife classification outcomes.

To get the ground truth label of knife selection, we have established the following rules: 1. Use a fruit knife by default. 2. Use a serrated knife for extremely hard foods (e.g., sweet potato), foods with a hard skin (e.g., lemon), or tough and deformable foods (e.g., bread). Based on these rules, we conducted experimental cutting using a fixed policy, with details provided in Section V-A.1.

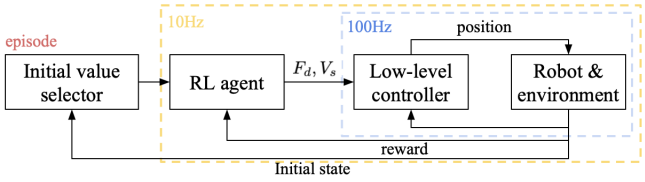

Fig. 3: The training pipeline of the RL agent. The low-level controller operates at 100 Hz. The RL agent, given the reward and state, outputs the desired downward force (F_d) and sawing speed (V_s). The initial value selector runs at the beginning of each episode.

C. Reinforcement learning adaptive cutting

1) Reward function: Our objective in optimizing cutting techniques is to achieve a relatively high cutting speed while minimizing the effort required to cut the food. Intuitively, using a higher desired vertical force $\widetilde{F_z}$ can accelerate the cutting process but also result in greater resistance. An appropriate sawing speed $\widetilde{V_s}$ facilitates smoother cutting but increase friction in the sawing direction, thus consuming more energy. To find the optimal $\widetilde{F_z}$ and $\widetilde{V_s}$ that balance between cutting speed and energy efficiency, we devised a reward function to evaluate the cutting performance:

$$r_t = -K_e(E^x_{(t-\Delta t):t} + E^z_{(t-\Delta t):t}) + K_z \Delta z_{(t-\Delta t):t}$$

where r_t represents the reward at time t. The coefficients K_e and K_z weigh the contributions of energy consumption and movement in the z-axis, respectively. The term $(E^x_{(t-\Delta t):t} + E^z_{(t-\Delta t):t})$ quantifies the total energy expended during the time interval Δt, adding an energy penalty to the reward to promote energy efficiency by the agent. The term $\Delta z_{(t-\Delta t):t}$ measures the displacement in the z-axis, incentivizing the agent to increase the cutting speed over the interval Δt. Given the use of this reward function, a reinforcement learning framework can be readily implemented to optimize the cutting process. We select an on-policy RL algorithm, Proximal Policy Optimization (PPO) [23] for its stable convergence, utilizing techniques such as Generalized Advantage Estimation (GAE) and clipping in its actor-critic framework.

In our implementation, $E^x_{(t-\Delta t):t}$ and $E^z_{(t-\Delta t):t}$ are computed from the measured force and displacement along the sawing and downward directions:

$$E^x_{(t-\Delta t):t} = \sum_{\tau=t-\Delta t}^{t} F_x(\tau)\Delta x(\tau),$$

$$E^z_{(t-\Delta t):t} = \sum_{\tau=t-\Delta t}^{t} F_z(\tau)\Delta z(\tau).$$

The reward then penalizes $|E^x_{(t-\Delta t):t}| + |E^z_{(t-\Delta t):t}|$, because resistance in either direction contributes to the effort required for cutting. The weights K_e and K_z were selected to normalize the two reward terms to comparable magnitudes during preliminary robot trials, so that the agent is encouraged to make downward progress without ignoring the energy penalty.

2) Observation space and action space: To effectively train RL algorithms on actual robots, it is crucial to design the RL settings to ensure rapid convergence. Our setup features a low-dimensional observation space and a discrete action space, complemented by an initial value selector that outputs the optimal starting value to expedite agent convergence.

The observation space is an 6-dimensional vector defined as: $O_t = [v_x, v_z, f_x, f_z, \widetilde{F_z}, \widetilde{V_s}]$. Where v_x and v_z denote the velocities of the knife in the x and z axes. f_x and f_z measure the forces exerted in the respective axes. $\widetilde{F_z}$ and $\widetilde{V_s}$ correspond to the desired force in the z-axis and the desired sawing speed at time t, respectively.

The action space is defined as a 4-dimensional discrete set, where each dimension corresponds to an increase and decrease of one unit in the desired force $\widetilde{F_z}$ and the sawing speed $\widetilde{V_s}$. Due to safety concerns and the stability of the knife mount, the target vertical force is bounded by -30 N $\leq \widetilde{F_z} \leq 0$ N, and the target sawing speed is constrained to 0 cm/s $\leq \widetilde{V_s} \leq 2$ cm/s. These bounds also reduce unsafe motions and excessive deformation of soft foods during real-world RL training.

3) Initial value selector: The initial value selector is designed to choose starting values for $\widetilde{F_z}$ and $\widetilde{V_s}$. During the initial exploration phase of training, it randomly outputs values to ensure a broad sampling of data. After collecting data for some episodes, the selector employs the critic network to choose initial values that are expected to yield the highest rewards.

4) Learning pipeline: An episode is typically defined by a single back-and-forth motion of the knife. If the duration exceeds 10 seconds, a time-out penalty is applied, terminating the episode. Additionally, the episode also concludes if the knife contacts the cutting board. Fig 3 illustrates the pipeline of the learning process. The lower-level cutting controller, described in III-A, operates at a frequency of 100 Hz. The actor-network takes the observation as input and outputs an action at a frequency of 10 Hz, providing $\widetilde{F_z}$ and $\widetilde{V_s}$ values to the lower-level controller. Thus, the commanded force and sawing speed can vary throughout a slice, although each command is tracked by the low-level controller between two consecutive policy updates. The initial value selector operates at the episode level. At the end of each episode, the history observations, actions, and rewards are stored in the memory buffer and used for policy updates.

IV. Experiment Settings

A. Experiment setup

Fig 4 shows our experimental setup. Our system uses a 6-DoF UR5e Robot arm by Universal Robotics. This arm is equipped with a wrist-mounted F/T sensor from Nordbo Robotics. Fig 4(a) shows the robot setup. The easy-to-switch knife mount shown in Fig 4(c) can hold the serrated knife and fruit knife simultaneously and switch the knife by rotating the last joint of the robot by 180 degrees without manually changing the knife. The cutting board equipped with a slide vise can hold the food tightly.

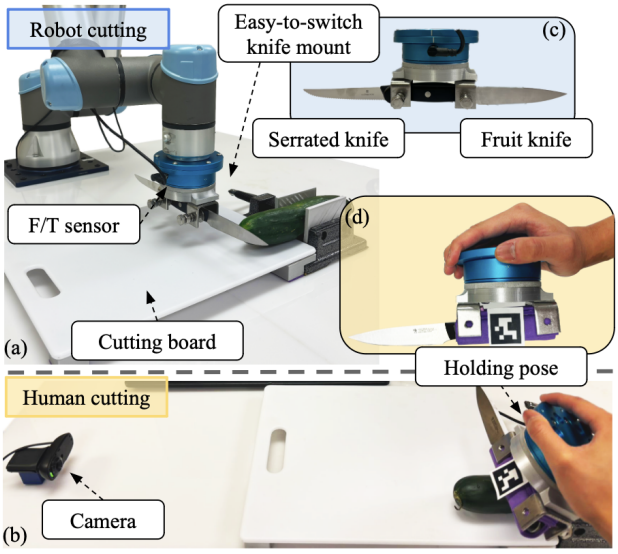

Fig. 4: (a) Experimental setup of the perception-manipulation robotics slicing system. (b) Experimental setup of the human evaluation system (c) Close-Up View: The easy-switch knife mount. (d) Close-Up View: Human holding the knife, ensuring accurate force/torque data collection.

B. Human evaluation setup

The human evaluation setup is designed to monitor force and motion during human cutting. This setup features a camera, a user interface, and a knife outfitted with a F/T sensor. The camera tracks an AprilTag [24] affixed to the knife, ensuring precise motion capture. When cutting, users should adopt the holding pose as depicted in Fig 4(d) to ensure the F/T sensor can accurately record the force and torque data. The human trials were used as a reference for comparison rather than as a statistically exhaustive human-subject study. Human participants were allowed to use their natural cutting styles and were not constrained by the robot's force and velocity limits.

The coefficient for converting pixel measurements to real-world distances needs calibration to ensure accurate real-world measurements. We attached an AprilTag to the knife and directed the robot to hold it while moving a known distance along the z and x axes. By measuring the corresponding pixel displacement, we determined the pixel-to-real-world conversion coefficient.

C. Dataset for classification and cutting experiments

Fig 5 displays the variety of food used in our experiments, ranging from the softest (tofu) to the hardest (sweet potato). The left column is labeled as cutting with a serrated knife, while the right column is labeled as cutting with a fruit knife. All these foods were utilized in the knife selection module experiments. The first 2 rows of foods served as unseen items for validation, with the remaining 10 types used for training. Additionally, 10 types of food, each with one slice shown on the right, are used in cutting experiments.

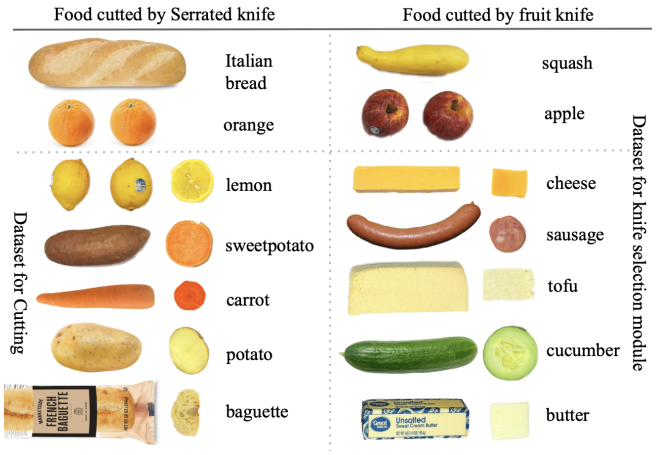

Fig. 5: Dataset Overview for Experiments: The left column lists foods cut with a serrated knife, while the right column lists those cut with a fruit knife. All foods serve as the dataset for the knife selection module's classification task. Additionally, ten types of food, each with one slice shown on the right, are used in cutting experiments.

D. Neural network and RL agent implementation

The Actor network of the RL agent consists of 4 fully connected layers, where the first 3 layers have 128 units each and use ReLU activation. The final layer outputs logits for action selection. The Critic network shares the structure of the Actor network but ended with a single output unit that estimates the state value. The network is updated after each episode. It is trained over 10 epochs each episode using the Adam optimizer at a learning rate of 10^{-3}, which decays by a factor of 0.3 every 50 epochs. The batch size is set at 64. The clipping parameter is fixed at 0.2, while the γ and λ parameters for the GAE are set at 0.9 and 0.95.

V. Experiment Results

In this section, we first evaluate the knife selection module. Next, we train our RL adaptive controller from scratch using eight different types of food to analyze the converged policies. Based on these, we propose two cutting frameworks: one with a fixed policy and the other using a pre-trained RL adaptive controller. We then compare the performance of both frameworks against human behavior.

A. Knife selection module

1) Ground truth labelling: To label which knife to use with each food, we conduct the robot to perform fixed policy cutting with desired downward force as -15 N and sawing speed as 1 cm/s, which takes the middle value of our parameter range. If the fruit knife is stuck with this fixed policy, it is very likely that the fruit knife is not proper for cutting this kind of food and a serrated knife is needed, also, the learning process is likely to stuck. The labeling results are shown in Fig 5, where the foods in the left column require a serrated knife, while those in the right column can be cut with a fruit knife.

2) Trial cut parameter exploration: By performing a trial cut on different foods, the knife selection module can select the proper knife for this certain food. A trial cut is conducted using a fixed target vertical force $\widetilde{F_z}$ and a sawing speed $\widetilde{V_s}$. During this process, force data are collected. The trial cut is halted once 200 data points have been recorded. We performed multiple trial cuts with different $\widetilde{F_z}$ and $\widetilde{V_s}$ to identify the optimal policy, then combined it with a second policy to improve performance.

To optimize the parameters for the trial cut, we selected target vertical forces of -10 N and -5 N to ensure the force was not excessive for cutting through the food. For the sawing speed $\widetilde{V_s}$, we chose 0.0, 0.5 cm/s, and 1.0 cm/s. We conducted 5 trial cuts for each parameter set across 14 different types of food, totaling 420 trial cuts. Then we split the data into a training set and a validation set in a 3 : 7 ratio and used an SVM to perform binary classification. F_x and F_z series are selected from the dataset to characterize the food properties, as they indicate the resistance in the vertical and sawing directions, respectively. We then trim the first and last 50 data points and calculated the average F_x and F_z values from the middle 100 data points as features representing the trial cut. The classification accuracy for each parameter set are displayed in Table I.

TABLE I: Classification accuracy results of exploring different trial cut parameters.

$\widetilde{V_s}$ \ $\widetilde{F_z}$	-5 N	-10 N
0.0 cm/s	0.76	0.71
0.5 cm/s	0.81	**0.95**
1.0 cm/s	0.90	0.81

The results indicate that a target vertical force $\widetilde{F_z}$ of -10 N and a sawing speed $\widetilde{V_s}$ of 0.5 cm/s most accurately reflect the food properties. To improve performance, we tried to combine it with a second set of trial cut parameters and found that of -5 N and 0.0 cm/s, achieving a 100% accuracy. Additionally, we tested this trial cut combination by training the SVM with 10 types of food and testing on the remaining 4 as an unseen validation set, achieving a success rate of 100% in correctly selecting the appropriate knife.

B. Learn the optimal cutting parameters

In this section, we train our RL adaptive controller from scratch using 8 different types of food, with 5 (butter, cheese, tofu, sausage, cucumber) cut using a fruit knife and 3 (baguette, carrot, potato) using a serrated knife. This training was conducted to evaluate the convergence capabilities of our RL controller and to determine the final policy it adopts.

1) Training process: We train the RL agent on the real robot 5 times from scratch for each type of food. To make the RL agent broadly explore the space, we set the initial value of $\widetilde{F_z}$ and $\widetilde{V_s}$ randomly at the beginning of the first 5 episodes. To ensure policy convergence, we monitor the loss curves of the actor and critic networks, the reward curve, and the entropy of the policy. Convergence is indicated

TABLE II: Experiment Results of Cutting. The abbreviations RL and FP represent cutting performed with the RL adaptive controller and a fixed policy, respectively; R, CE, and CR denote the evaluation metrics: Reward, Cutting Efficiency, and Cutting Rate. The first row of foods, indicated as FK at the top left, is cut using a fruit knife, while the second row, indicated as SK, is cut with a serrated knife. Best results are highlighted in bold. For all 3 metrics, higher values are preferable. Standard deviations are provided for results with variance from RL and human results, while they are omitted for the fixed policy results due to their relative stability.

FK	cucumber			tofu			butter			cheese			sausage		
	RL	FP	Human	RL	FP	Human	RL	FP	Human	RL	FP	Human	RL	FP	Human
R	**2.4** $\pm$ 0.0	1.8	1.0 $\pm$ 0.1	**2.3** $\pm$ 0.1	1.5	0.9 $\pm$ 0.2	**2.3** $\pm$ 0.1	1.7	0.7 $\pm$ 0.2	**2.2** $\pm$ 0.1	1.4	0.3 $\pm$ 0.2	**2.3** $\pm$ 0.2	1.4	0.9 $\pm$ 0.3
CE	3.3 $\pm$ 0.1	2.3	**3.5** $\pm$ 0.3	2.9 $\pm$ 0.1	3.2	**4.5** $\pm$ 0.6	3.3 $\pm$ 0.1	**3.8**	3.4 $\pm$ 0.5	3.3 $\pm$ 0.1	**3.6**	2.1 $\pm$ 0.5	**3.8** $\pm$ 0.3	3.5	3.7 $\pm$ 0.6
CR	1.5 $\pm$ 0.0	1.5	**2.6** $\pm$ 0.3	1.4 $\pm$ 0.2	1.0	**2.0** $\pm$ 0.7	1.4 $\pm$ 0.1	1.1	**1.7** $\pm$ 0.6	1.0 $\pm$ 0.4	0.9	**1.2** $\pm$ 0.5	1.4 $\pm$ 0.0	0.9	**1.7** $\pm$ 0.2
SK	baguette			potato			lemon			carrot			sweetpotato		
	RL	FP	Human	RL	FP	Human	RL	FP	Human	RL	FP	Human	RL	FP	Human
R	**2.0** $\pm$ 0.1	1.2	0.5 $\pm$ 0.1	**2.0** $\pm$ 0.0	0.9	1.0 $\pm$ 0.1	**2.0** $\pm$ 0.1	1.4	1.3 $\pm$ 0.3	**1.8** $\pm$ 0.2	**1.8**	0.8 $\pm$ 0.1	0.3 $\pm$ 0.1	0.2	**1.0** $\pm$ 0.2
CE	1.9 $\pm$ 0.1	**3.6**	2.0 $\pm$ 0.4	**4.1** $\pm$ 0.3	2.5	3.5 $\pm$ 0.3	**3.6** $\pm$ 0.0	**3.6**	3.0 $\pm$ 0.2	2.4 $\pm$ 0.1	2.2	**3.1** $\pm$ 0.3	1.3 $\pm$ 0.1	1.4	**2.3** $\pm$ 0.1
CR	1.3 $\pm$ 0.2	0.7	**1.7** $\pm$ 0.0	0.8 $\pm$ 0.0	0.7	**2.2** $\pm$ 0.2	1.0 $\pm$ 0.0	0.9	**3.3** $\pm$ 1.0	1.1 $\pm$ 0.2	1.1	**1.5** $\pm$ 0.3	0.7 $\pm$ 0.3	0.4	**1.8** $\pm$ 0.4

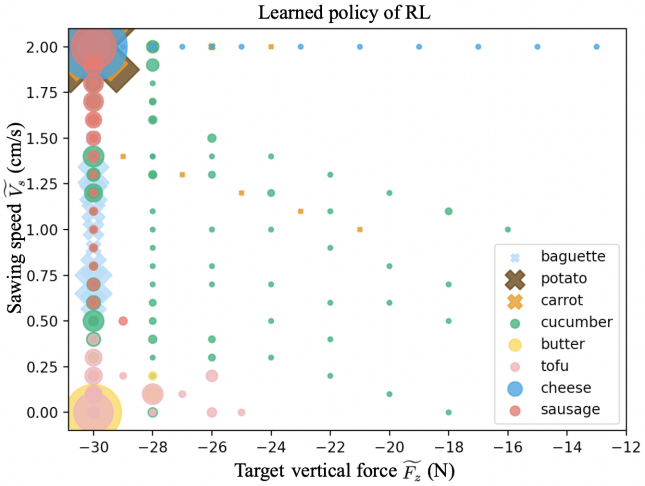

Fig. 6: Learned policy across different foods: the x-axis represents the desired vertical force ($\widetilde{F_z}$), and the y-axis indicates the sawing speed ($\widetilde{V_s}$) output by the actor network. Foods cut with a fruit knife are marked by circles, while those cut with a serrated knife are shown as crosses. The size of each symbol reflects the frequency with which the same policy was applied during the cutting process.

when the loss curves stabilize, and the entropy reaches 0. After convergence, we let the robot cut 3 more slices with the learned policy. Typically, it takes about 20 episodes, involving the cutting of 10 slices, for the RL agent to converge. However, this can vary depending on the type of food and the initial conditions.

2) Learned policy discussion: Figure 6 shows the output policy for the last 3 slices of each food type after policy convergence. Each point on the graph represents an output policy, plotted with $\widetilde{F_z}$ on the x-axis and $\widetilde{V_s}$ on the y-axis. Circles indicate foods cut with a fruit knife, while crosses represent those cut with a serrated knife. The symbol size reflects the frequency of corresponding $\widetilde{F_z}$ and $\widetilde{V_s}$ values. Most points cluster around policies of $\widetilde{V_s} = 2, \widetilde{F_z} = -30$ and $\widetilde{V_s} = 0.0, \widetilde{F_z} = -30$, indicating that, for example, the controller tends to chop tofu and butter with maximum force. In contrast, it opts for a maximum-speed sawing motion for cheese. For cucumber, however, the exact sawing speed is less crucial, as convergence occurs at a moderate velocity. This behavior suggests that, under the current reward and hardware constraints, the learned policy often discovers a small number of dominant cutting modes rather than a widely varying continuous policy over the full parameter space. We therefore interpret the RL controller as an adaptive parameter-selection mechanism that can automatically choose and refine chopping/sawing behaviors from force feedback, instead of claiming that it learns a fully general continuous cutting controller.

Why are they not converging to a value in the middle stably? Our reward function is designed to encourage finding a balance between cutting speed and energy efficiency. However, this pattern is not as distinct as expected, largely due to significant noise in the data. Another potential issue is the limitation on the desired force. While -30 N is typically sufficient, the agent consistently converges to the maximum force, suggesting it is inadequate. In contrast, the sawing speed does not always reach its maximum. However, due to safety concerns and the limitations of the robot's structure, we cannot increase the force range further. The boundary-seeking behavior also highlights a limitation of the current reward design: when faster downward progress dominates the marginal energy penalty, the optimal behavior can become biased toward the maximum allowed force.

Based on these findings, it appears that the food-cutting task can be effectively addressed using just a few fixed policies. Therefore, in the following section, we will evaluate both the RL controller and the fixed policy controller, comparing their performance to human behavior.

C. Cutting result evaluation

We evaluated the cutting result on pre-trained RL adaptive controller, fixed policy, which are baseline chopping and baseline sawing, as well as human cutting.

*1) Rl adaptive controller (**RL**):* When deploying the RL adaptive controller for actual cutting tasks, training a policy from scratch could lead to food waste. To address this, we initially trained separate policies for fruit knives and

serrated knives. The fruit knife model was trained using cucumbers, while the serrated knife model was trained on baguettes. Utilizing these pre-trained models, the RL agent can effectively cut each slice and typically converges within 1 to 2 slices.

*2) Fixed policy (**FP**):* We use fixed policies as our baselines, with parameters derived from the results in Section V-B. We choose the chopping policy $\widetilde{F_z} = -30$ N, $\widetilde{V_s} = 0.0$ cm/s for the fruit knife, and sawing policy $\widetilde{F_z} = -30$ N, $\widetilde{V_s} = 2$ cm/s for the serrated knife.

*3) Human cutting results (**Human**):* Leveraging the human evaluation setup described in Section IV-B, we asked 3 humans to cut the food using their cutting styles. Because the number of participants is small and the human trials were unconstrained, these results are intended to provide a practical reference rather than a statistically comprehensive comparison with human skill.

4) Evaluation metrics: To evaluate the cutting performance of the policies, we employ three metrics:

- Reward (**R**): the average reward value from the last 3 slices, as detailed in Section III-C.
- Cutting Efficiency (**CE**): calculated as the average vertical cutting speed divided by the average energy consumed.
- Cutting Rate (**CR**): determined by dividing the food's diameter by the time taken to cut one slice.

Higher values of the three metrics indicate better performance for all metrics. Tables II shows the result. The highest values in each evaluation category are highlighted in bold. Due to the variance in RL and human results, we present both the average and the standard deviation of the last 3 slices. For fixed policies, which exhibit less variance, only the average values are reported. The analysis shows that the RL method consistently achieves the highest reward, which is expected since it optimizes for maximum reward. RL also demonstrates comparable cutting efficiency to fixed policies and human performance, outperforming them in four instances.

5) Results discussion: Fixed policies generally perform worse than RL across the three matrices, likely because RL effectively combines chopping and sawing techniques; for example, a small sawing motion can enhance chopping efficiency. However, all fixed policy cuttings were successful and produced good slices, suggesting that a correct choice of chopping or sawing policy is sufficient for basic cutting tasks, given that other experiments using a serrated knife with a chopping policy resulted in failures when cutting sweet potatoes, lemons, and carrots. This observation also explains why fixed policies are competitive in several cases: many foods can be cut by one of a few dominant modes once the knife is selected correctly. The benefit of the RL controller is that it can select and refine these modes automatically from interaction feedback, rather than relying entirely on manually chosen parameters. In contrast, humans displayed comparable cutting efficiency but did not excel in maximizing rewards. Humans consistently achieved the highest cutting rates, likely because they could use greater force and speed and were not constrained by the robot's force and velocity bounds.

VI. Conclusion and Future Work

This paper presents a perception-manipulation robotics system for slicing food. Initially, we conduct a trial cut, analyzing displacement, velocity, and force in the downward direction to determine whether to use a serrated or fruit knife. Subsequently, an RL-based adaptive controller resolves the cutting task. This controller quickly converges to an optimal policy using a pre-trained model. During training, we observed that the policy typically converges to a cutting policy utilizing either maximum downward force and zero velocity (chopping) or maximum velocity (sawing), suggesting that a fixed policy could be sufficient for the task. We assessed both the fixed policy controller (baseline) and the adaptive RL controller, comparing their performance to human cutting. The results suggest that RL is most useful in this setting as an automatic parameter-selection and refinement mechanism, while the final behaviors often correspond to interpretable chopping or sawing modes.

In the future, with our developed system for evaluating human cutting, we can easily collect data and apply imitation learning to enhance the cutting task. Another important direction is to improve the reward function. The current reward mainly combines estimated mechanical work and downward progress, and therefore does not explicitly model cut quality, peak force, food deformation, viscous friction, or non-Newtonian effects that may appear in wet or soft foods such as tofu and butter. Future reward designs could include slice quality, deformation, peak-force constraints, and friction-aware terms to better capture these aspects. Moreover, we observed that the position of the holding hand is crucial for slicing, especially for deformable foods. Positioning the holding hand closer to the cutting site tends to improve cut quality. Further research is needed to develop an effective dual-arm cutting system.

References

[1] I. Lenz, R. A. Knepper, and A. Saxena, "Deepmpc: Learning deep latent features for model predictive control," in *Robotics: Science and Systems*, 2015.

[2] R. Ye, Y. Hu, Yuhan, Bian, L. Kulm, and T. Bhattacharjee, "Morpheus: a multimodal one-armed robot-assisted peeling system with human users in-the-loop," 2024.

[3] X. Luo, S. Jin, H.-J. Huang, and W. Yuan, "An intelligent robotic system for perceptive pancake batter stirring and precise pouring," 2024.

[4] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth, "Robotic roommates making pancakes," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pp. 529–536, 2011.

[5] 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," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5159–5166, 2022.

[6] I. Mitsioni, Y. Karayiannidis, J. A. Stork, and D. Kragic, "Data-driven model predictive control for the contact-rich task of food cutting," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pp. 244–250, 2019.

[7] I. Mitsioni, Y. Karayiannidis, and D. Kragic, "Modelling and learning dynamics for robotic food-cutting," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 1194–1200, 2021.

[8] G. Zeng and A. Hemami, "An adaptive control strategy for robotic cutting," in *Proceedings of International Conference on Robotics and Automation*, vol. 1, pp. 22–27 vol.1, 1997.

[9] X. Mu, Y. Xue, and Y.-B. Jia, "Robotic cutting: Mechanics and control of knife motion," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3066–3072, 2019.

[10] X. Mu, Y. Xue, and Y.-B. Jia, "Dexterous robotic cutting based on fracture mechanics and force control," *IEEE Transactions on Automation Science and Engineering*, pp. 1–18, 2023.

[11] A. Padalkar, M. Nieuwenhuisen, S. Schneider, and D. Schulz, "Learning to close the gap: Combining task frame formalism and reinforcement learning for compliant vegetable cutting.," in *ICINCO* (O. Gusikhin, K. Madani, and J. Zaytoon, eds.), pp. 221–231, ScitePress, 2020.

[12] C. C. Beltran-Hernandez, N. Erbetti, and M. Hamaya, "Sliceit! – a dual simulator framework for learning robot food slicing," 2024.

[13] Z. Xu and Y. She, "Letac-mpc: Learning model predictive control for tactile-reactive grasping," 2024.

[14] R. Ishikawa, M. Hamaya, F. Von Drigalski, K. Tanaka, and A. Hashimoto, "Learning by breaking: Food fracture anticipation for robotic food manipulation," *IEEE Access*, vol. 10, pp. 99321–99329, 2022.

[15] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert vol. 15, pp. 561–572, 01 2003.

[16] A. Straižys, M. Burke, and S. Ramamoorthy, "Surfing on an uncertain edge: Precision cutting of soft tissue using torque-based medium classification," 2019.

[17] K. Zhang, M. Sharma, M. Veloso, and O. Kroemer, "Leveraging multimodal haptic sensory data for robust cutting," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pp. 409–416, 2019.

[18] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, "A dmps-based framework for robot learning and generalization of humanlike variable impedance skills," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1193–1203, 2018.

[19] E. Heiden, M. Macklin, Y. Narang, D. Fox, A. Garg, and F. Ramos, "Disect: A differentiable simulation engine for autonomous robotic cutting," 2021.

[20] X. Mu and Y.-B. Jia, "Physical property estimation and knife trajectory optimization during robotic cutting," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2700–2706, 2022.

[21] A. Sawhney, S. Lee, K. Zhang, M. Veloso, and O. Kroemer, "Playing with food: Learning food item representations through interactive exploration," 2021.

[22] M. C. Gemici and A. Saxena, "Learning haptic representation for manipulating deformable food objects," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 638–645, 2014.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[24] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3400–3407, IEEE, May 2011.