2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

LAVA: Long-horizon Visual Action based Food Acquisition

Amisha Bhaskar, Rui Liu, Vishnu D. Sharma, Guangyao Shi, Pratap Tokekar

***Abstract*— Robotic Assisted Feeding (RAF) addresses the fundamental need for individuals with mobility impairments to regain autonomy in feeding themselves. The goal of RAF is to use a robot arm to acquire and transfer food to individuals from the table. Existing RAF methods primarily focus on solid foods, leaving a gap in manipulation strategies for semi-solid and deformable foods. We present Long-horizon Visual Action-based (LAVA) food acquisition of liquid, semisolid, and deformable foods. Long-horizon refers to the goal of "clearing the bowl" by sequentially acquiring the food from the bowl. LAVA is hierarchical: (1) At the highest level, we determine primitives using ScoopNet. (2) At the mid-level, LAVA finds parameters for the low-level primitives. (3) At the lowest level, LAVA carries out action execution using behavior cloning. We validate LAVA on real-world acquisition trials involving granular, liquid, semisolid, and deformable foods along with fruit chunks and soup. Across 46 bowls, LAVA acquires much more efficiently than baselines with a success rate of $89 \pm 4\%$, and generalizes across realistic plate variations such as varying positions, varieties, and amount of food in the bowl. Datasets and supplementary materials can be found on our website.**

I. INTRODUCTION

For individuals with limited mobility or disabilities, the act of feeding themselves can pose a significant challenge. This challenge has motivated the development of Robotic Assisted Feeding (RAF) [1] aiming to restore independence and enhance the quality of life for those affected, while also alleviating the caregiver burden. A key component of such an assistive feeding system is bite acquisition, i.e., the act of a robotic arm picking up morsels of food from a plate to transfer the food to a person's mouth.

Navigating the diverse array of foods—from granular cereals to semi-solid food such as yogurt and deformable food items such as tofu—without breakage or deformation presents significant challenges for RAF [2], [3]. Additionally, the dynamic positioning of food chunks within a fluid medium complicates the prediction of their exact location at the time of scooping, requiring sophisticated sensing and real-time adaptation capabilities. Traditional RAF methodologies have depended on fixed heuristics and learned primitives for food manipulation skills such as skewering [4]–[7], bite transfer [4], [8], [9], scooping [2], [10] and even end-to-end system [11], [12].

This approach, while effective for isolated actions, falls short in replicating complex, sequential behaviors exhibited by humans during feeding. Humans adeptly combine various actions, such as scooping both solid chunks and liquid from a bowl in a single motion or rearranging food items for easier acquisition. This underscores the need for long-horizon acquisition strategies capable of managing both the rigidity of solid foods and complexity of deformable items.

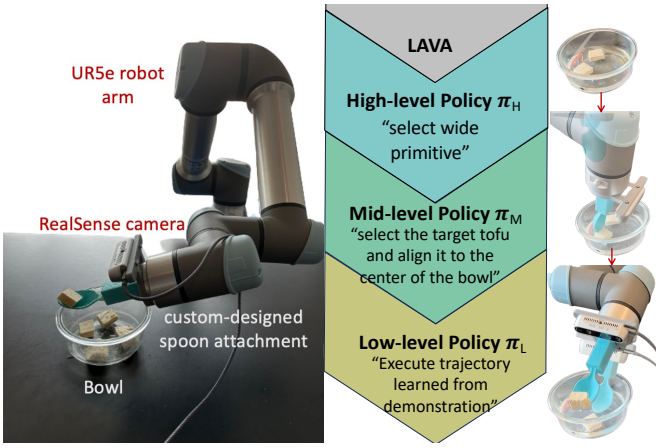


Fig. 1: LAVA System setup with an illustrative description of the proposed framework and snapshots of task execution.

Recent advancements in skill-based reinforcement learning (RL) offer promising methodologies for modeling these complex, long-horizon manipulation sequences in a hierarchical manner. This entails first learning a high-level policy for composing skills [13], and then optionally inferring the parameters of low-level skills separately [14], [15]. Such approaches have shown potential, yet they face limitations when applied to the food domain, which demands high-fidelity models for food deformation, visual recognition, and utensil interaction not fully captured in current simulations. While VAPORS [7] demonstrates effective long-horizon planning for specific food items such as noodles, it relies on learning plate dynamics in simulation, limiting its applicability broader categories of food types.

We seek to find an appropriate layer of abstraction for feeding, which can leverage the benefits of (1) hierarchical planning for long-horizon manipulation; (2) vision-based primitives for fine-grained control; and (3) flexible approach that can dynamically adapt to the wide variety of challenges presented by different food types.

We introduce **LAVA** (Long-horizon Acquisition via Visual Action), as a hierarchical policy for sequential planning of food acquisition (see Figure 1). Our approach consists of three components: a high-level policy that identifies primitives based on visual inputs; a mid-level policy to refine these primitives and parameterize the actions of the lower-level policy and a low-level policy to acquire food items and

All authors are from the University of Maryland, College Park, MD 20742 USA. Emails: {amishab, ruiliu, vishnuds, gyshi, tokekar}@umd.edu

This work is supported by the National Science Foundation under Grant No. 1943368 and an Amazon Research Award.

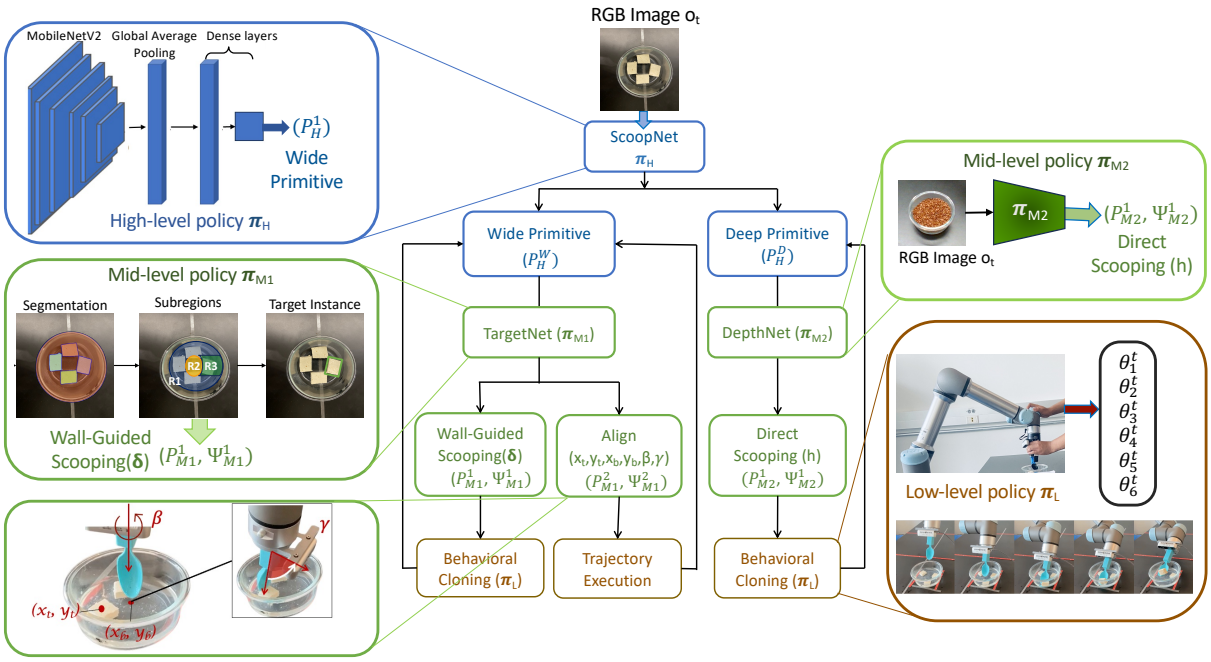


Fig. 2: **LAVA**: System Architecture of LAVA which employs a high level policy(blue) π_H to select amongst discrete high level primitives P_H^k, which gets refined by mid-level policy (green) π_M to select amongst mid-level primitivesP_M^k, low-level vision parametrized policy π_L (brown) executes trajectory learned from Behavioral cloning for long-horizon food acquisition.

sequentially clear the bowl.

The key contributions of this paper are:

- We present a hierarchical policy framework, **LAVA**, for long-horizon, visual-action-based food acquisition.
- Our method showcases adaptability and robustness across a diverse range of food types, effectively clearing bowls.
- We introduce a dataset of food items, showcasing different volumes and spatial arrangements within the bowl.
- We evaluate the learned scooping policies through real-world deployment system with UR5e and end-effector coupled with spoon attachment and D435i RealSense camera on the wrist.

II. RELATED WORK

In this section, we will discuss related work in robot-assisted feeding, learning from demonstration, and more generally long-horizon planning and control.

A. Robotic-Assisted Feeding

RAF comprises two stages: bite acquisition and bite transfer. Previous work in RAF focused on bite acquisition and transfer with the aid of robotic arms and specialized tools such as spoons and forks [2], [4], [6], [8], [16]. Models such as SPANet [5] have demonstrated proficiency in mapping images of food items to actions. However, challenges remain in handling semi-solid and deformable foods, where generalizable strategies are scarce and bimanual scooping [2] techniques have shown limited success. Market-available devices [12] offer mealtime assistance but are constrained by their reliance on teleoperation and the physical limitations of their design. While prior research has made strides in visual planning and manipulation for specific food items [7], a comprehensive approach that addresses the adaptability to a wide array of food types and real-world feeding scenarios is still needed.

B. Learning from Demonstration

Learning from Demonstration (LfD) is a methodology where robots learn new skills by observing expert demonstrations often provided by humans. LfD has been applied across various domains, including robotic assembly in manufacturing [17], path planning for complex tasks [18], and assistive technologies in rehabilitation [19]. Our research primarily utilizes kinesthetic teaching [20], where a human physically guides the robot to provide demonstrations. Within LfD's learning objectives, our focus is on developing policies for handling semi-solid and deformable food items, and optimizing their scooping trajectories.

C. Long-Horizon Planning and Control

Traditional task-and-motion planning approaches rely on extensive domain knowledge and fixed task sequences [21]–[23], but falter due to the unpredictable dynamics of food on a plate and the complexity of state estimation. Model-based planning has shown promise in tasks such as dough manipulation by using environment dynamics learned from visual inputs to plan action sequences [13], [24]. Hierarchical reinforcement Learning offers a solution by dividing decision-making into high-level strategic planning and execution by discrete, parameterized low-level primitives [25]. We build on these ideas to develop a hierarchical policy for adaptation in the real world of specialized primitives that can handle a variety of food items.

III. Problem Statement

We tackle the challenge of sequential bite acquisition to maximize the success rate and efficiency of long-horizon food acquisition to ensure efficient bowl clearance. The focus is on a variety of food types, ranging from granular items such as cereals to semi-solid foods such as yogurt, and deformable substances such as tofu, all within a bowl fixed in position and assumed to be scoopable with a spoon.

We are given bowl image observations $o \in \mathbf{R}_+^{W \times H \times C} = \mathcal{O}$ of unknown bowl states S from a wrist camera attached to the arm with a custom spoon end-effector. Here, W, H, and C denote the image dimensions. We have access to expert demonstration data for robot proprioceptive information (joint positions). Our goal is to learn a policy $\pi(\theta_t|o_t)$ that takes RGB images as input (o_t) and outputs joint angles θ_t that form sequential actions to clear the bowl.

IV. Proposed Approach

We formalize the long-horizon food acquisition setting as a hierarchical policy π. To do so we decouple π into separate high, mid, and low-level sub-policies. We have K discrete manipulation primitives P_H^k, $k \in 1, ..., K$, and learn a high-level policy π_H which selects amongst these primitives based on visual input o_t. The mid-level policy π_M further refines this selection, parameterizing the low-level policy π_L based on the chosen primitive and additional visual inputs.

This low-level policy then executes a sequence of actions θ_t^k, aimed at achieving precise food acquisition. This hierarchical arrangement is formalized as follows:

- **High-level policy:** $\pi_H(P_H^k|o_t)$ selects the manipulation primitive suitable for the current visual scene.
- **Mid-level policy:** $\pi_M(P_M^k, \psi_M^k|o_t, P_H^k)$ refines this choice by finding parameters for the low-level actions depending on the specific food item's characteristics.
- **Low-level policy:** $\pi_L(\theta_t^k|P_M^k, \psi_M^K)$ produces the joint angles that are executed on the robot.

We consider low-level actions θ_t, parameterized by the position of the tip of a spoon (x, y) and spoon roll and pitch (γ, β) in the wrist frame of reference. We describe each module in LAVA (Figure 2) in further detail.

A. High-level Policy

At the highest level of our hierarchical model, the high-level policy $\pi_H(P_H^k|o_t)$ uses visual cues to select the most suitable scooping primitive—Wide Primitive (P_H^W) and Deep Primitive (P_H^D), based on the food type present.

1) Wide Primitive (P_H^W): Wide Primitive, is a strategy developed for handling foods that lack cohesion or are deformable, such as tofu or certain types of jelly. This method involves using the bowl's wall as a guide and support mechanism for the scooping action. By gently pressing the food against the wall of the bowl, it creates a pseudo-cohesive mass that can be scooped more easily. This technique is especially useful for foods that tend to scatter or break apart, as the wall provides the necessary containment to gather and scoop the food effectively. Instance scooping requires sophisticated control over the spoon's movement, including adjusting the angle applied against the food and the bowl wall, to achieve the desired outcome without compromising the integrity of the food or missing the target. It requires identifying the target instance to not collide with other instances or break them in the process. The wide primitive is implemented with the other two mid-level primitives align and wall-guided scooping described in Section IV-B.1

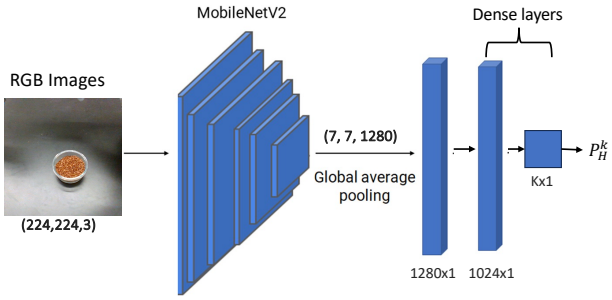


Fig. 3: ScoopNet outputs the softmax probabilities over the high-level primitive depending on the type of food items present in the image.

2) Deep Primitive (P_H^D): Deep Primitive, on the other hand, is a straightforward approach designed for foods that possess enough cohesion to be picked up directly by a spoon without requiring additional support or manipulation. This method is particularly effective for liquid and semi-solid foods such as yogurt or porridge, where the food's natural consistency allows it to adhere to the spoon when scooped directly from the top or side. The key to successful direct scooping lies in the precise control of the spoon's trajectory and depth of penetration into the food, ensuring that a sufficient quantity is acquired without disturbing the remaining contents of the bowl excessively. The deep primitive is implemented with the direct scooping mid-level primitive described in Section IV-B.2

3) ScoopNet (π_H): ScoopNet is a network designed to select between the two high level primitives based on the type of food, utilizing the MobileNetV2 architecture [26] as the base. We train on a dataset of 5316 images from a custom collection and additional sources, targeting binary classification of high-level primitives $P_H^1, ..., P_H^k$. We used data augmentation (including rotations, zooms, and flips) to increase robustness against food image variations. Our dataset is available online.

The network is initially trained on the ImageNet dataset, with a customized final layer for specific task adaptation. This configuration, along with a Global Average Pooling layer and two dense layers ending in a sigmoid activation, uses the Adam optimizer and binary cross-entropy loss for accurate classification. The detailed architecture, ScoopNet, is depicted in Figure 3. The output of ScoopNet is softmax probabilities over high-level primitives.

B. Mid-level Policy

The Mid-level Policy $\pi_M(P_M^k, \psi_M^K|o_t, P_H^k)$ serves as the intermediary layer that refines and parameterizes the chosen primitive for execution. This refinement is crucial for

bridging the gap between high-level strategy selection and low-level action execution.

1) TargetNet (π_{M1}) for Wide Primitive: We have designed TargetNet (Figure 4) that uses Mask R-CNN, tailored for the task of identifying and segmenting target food items such as tofu in a bowl, crucial for executing wide primitives. This model precisely segments food items, enabling the selection of appropriate mid-level primitives: wall-guided scooping and center align (described later in this section).

We use a custom dataset annotated for bowl, tofu, and target scooping areas, TargetNet employs transfer learning to accurately segment food items against diverse backgrounds, increasing its generalizability. In Section V-C.3, we report zero-shot generalization results for other types of food items. The model's training includes a COCOEvaluator to ensure segmentation accuracy meets COCO dataset [27] standards.

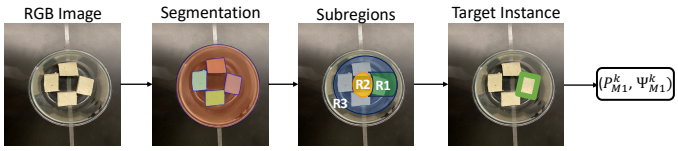


Fig. 4: TargetNet finds the next "target" item for the wide high-level primitive and the mid-level primitive that decides whether to scoop the target item or to align it first.

Post-training, TargetNet creates a binary mask for pixels that are "occupied" by instances of food items. We divide the surrounding region of interest into sub-regions. If a sub-region intersects the bowl boundary, it is considered to be "occupied." Otherwise, it is "unoccupied." A food item is classified as "R1" if it is rightmost and closest to the wall, "R2" if the food item is at the center of the bowl, and "R3" otherwise. The subsequent visualization and centroid calculation steps of detected instances help with determining its location in subregions of the bowl and its location with respect to the center of the bowl, selecting between mid-level primitives —Wall-guided Scooping(P^1_{M1}) or Align(P^2_{M1}) and predicting parameters for low-level policy.

Wall-guided Scooping (P^1_{M1}, ψ^1_{M1}) The Wall-guided Scooping strategy, parameterized by δ—the centroid distance of the target instance from the bowl's center—adapts its approach based on the target's proximity to the bowl's wall and the sub-region. For food items in subregion R1, the strategy uses the wall's structural support for a scooping action. Conversely, items in central subregion R2 require a pre-scooping alignment, tactically moving the food towards the wall to simplify the scooping motion.

Align (P^2_{M1}, ψ^2_{M1}) The alignment step is essential for orienting the spoon to the target instance and guiding its movement toward the bowl's center. This procedure takes into consideration the centroid coordinates of the tofu (x_t, y_t) and the bowl's center (x_b, y_b) as well as the spoon's roll (γ) and pitch (β). Two key parameters are computed:

- Spoon Orientation Angle: Calculated as $\gamma = \arctan(\frac{y_b - y_t}{x_b - x_t})$, this angle determines the necessary rotation of the spoon to align with the target instance, ensuring the spoon is positioned for optimal interaction and is untilted for planar push ($\beta = 0°$).
- Instance Push distance: Determined by (x_t, y_t, x_b, y_b), the instance is pushed towards the bowl's center, optimizing the positioning for the scooping action.

2) DepthNet (π_{M2}) for Deep Primitive: DepthNet (Figure 5) is designed for depth detection of food in a bowl based on visual input o_t and high-level primitive received from high-level policy.

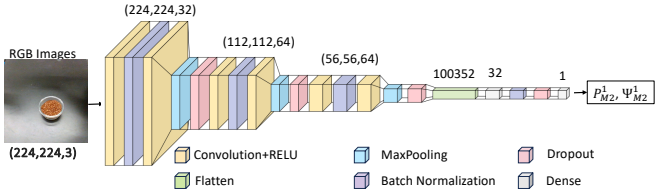


Fig. 5: DepthNet detects depth (h) of the food in the bowl.

DepthNet estimate the food volume in a bowl using a Sequential model with convolutional layers of 32, 64, and 128 filters. We use batch normalization for improved training stability and dropout layers at rates of 0.25 and 0.5 to prevent overfitting.

The model uses a flattened layer for data restructuring, followed by a dense layer with 32 neurons (using 'relu' activation) for feature processing. The architecture culminates in a final dense layer with a single neuron (using 'linear' activation) to predict the food's depth. DepthNet has been trained on a dataset of 1000 cereal images, categorized into three depth ranges in the bowl: 5.5 cm, 4 cm, and 2 cm, enabling precise depth estimation in varied food scenarios.

Direct scooping (P^1_{M2}, ψ^1_{M2}) The direct scooping strategy employs a feedback mechanism centered on a predefined scooping axis, $(\beta = 0°)$. The strategy utilizes the trained model on a dataset of correct trajectories taken by an expert human to scoop food from the bowl where the input is the position of the robotic arm relative to the bowl along with the estimated depth (h) of the food received from DepthNet and the output is the adjusted trajectory from behavioral cloning based on inputs. This real-time adjustment is critical for achieving precise interaction between the scoop and the food item, ensuring effective scooping without causing displacement or spillage and long-horizon acquisition as the level of food changes while sequential scooping. Furthermore, this strategy is enhanced by the implementation of trajectory selection from behavioral cloning.

C. Low-level policy

We use Behavioral Cloning (π_L) with kinesthetic teaching to adapt scooping actions for different food textures and consistencies at the lowest level. Various food items, with their unique requirements for scooping techniques, necessitate the modeling of distinct optimal scooping trajectories, especially for semi-solid and deformable foods. This process includes collecting demonstration data on joint positions, velocities, and timestamps to approach the scooping task as a trajectory optimization problem within the robot arm's joint space.

The objective is to minimize a cost function $J(\tau)$ over a trajectory τ, given by $J(\tau) = \int_0^T L(\text{ıı}(t), \dot{\text{ıı}}(t))dt$, where $\text{ıı}(t)$ and $\dot{\text{ıı}}(t)$ are the joint positions and velocities at time t, respectively, and $L(\cdot)$ is an instantaneous cost function penalizing deviations from the optimal trajectory. The Weiszfeld algorithm [28], [29] is used for this optimization, finding a trajectory $\hat{x}$ that minimizes the sum of Euclidean distances to demonstrated trajectories. It iteratively refines $\hat{x}$ until the adjustment falls below a small threshold ϵ.

The algorithm updates the estimate of $\hat{x}$ using $\hat{x}_{k+1} = \frac{\sum_{i=1}^{n} \frac{p_i}{|\hat{x}_k - p_i|2}}{\sum_{i=1}^{n} \frac{1}{|\hat{x}_k - p_i|2}}$, iterating until the change in $\hat{x}$ between iterations is below a predefined threshold ϵ. This method determines optimal trajectories for the robot arm's joints, enhancing the robot's scooping accuracy and effectiveness.

V. Experiments

A. Experimental Setup

The setup (Figure 1) comprises a UR5e robot arm, a custom spoon attachment, a RealSense camera, and a fixed-position bowl. The spoon is 10.0 cm long and fixed to the arm. We explore varied configurations across the amount, size, position, and depth of food as well as different food types including granular cereals, liquid water, and semi-solid yogurt in the bowl. Food position configurations encompass multiple numbers of tofu and fruit chunks placed in different instance positions across the bowl. The varied amount and food depth included cereals, water, yogurt, and jelly filled at different depth levels inside the bowl. Additionally, we conduct tests with tofu chunks inside soup as well as fruit chunks. For each food type, and depth, we conduct 10 trials of long-horizon food scooping attempts and for each position configuration in case of multiple tofu and fruit chunks, we conduct 5 trials of long-horizon food scooping attempts.

We collected data through kinesthetic teaching, which encompassed two different types of trajectory— wall-guided scooping and direct scooping, with twenty-five demonstrations recorded for each category with different parameters, focusing on RGB images and robot joint positions. This process was limited to cereals and tofu.

B. Baselines

In our study, we used two baselines, LAVA-low and Fixed Trajectory Scooping (FTS). For both baselines, the process begins with detecting the bowl in an RGB image using RetinaNet [30]. Upon identifying the bowl, we calculate its centroid and map this position to the robot's coordinate system. This allows the robot to move to the bowl's location, adjusting to a predetermined height and orientation. In the case of FTS, during tests with various food items in a stationary bowl position, wrist 2 of the robot arm is rotated by -0.6 radians to scoop along a predefined trajectory.

Conversely, LAVA-low, employs the same low-level policy π_L as **LAVA** for scooping. For deformable food and fruit chunks, we stick with the wall-guided scooping trajectory for the R1 region (Figure 4) and keep rotating the bowl every 45 degrees constantly so that the spoon can reach all the instances in the bowl near the wall and gets maximum coverage. In contrast for granular, liquid, and semi-solid foods we stick with direct scooping, adjusting its approach based on depth of the food within the bowl. This adjustment occurs once a predefined depth threshold is reached, to effectively target the lower layers of food, ensuring thorough bowl clearance.

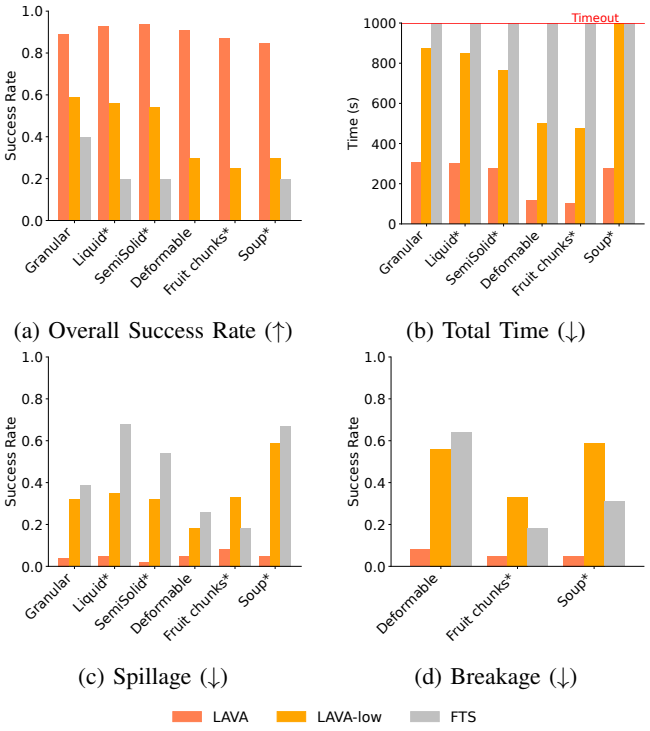
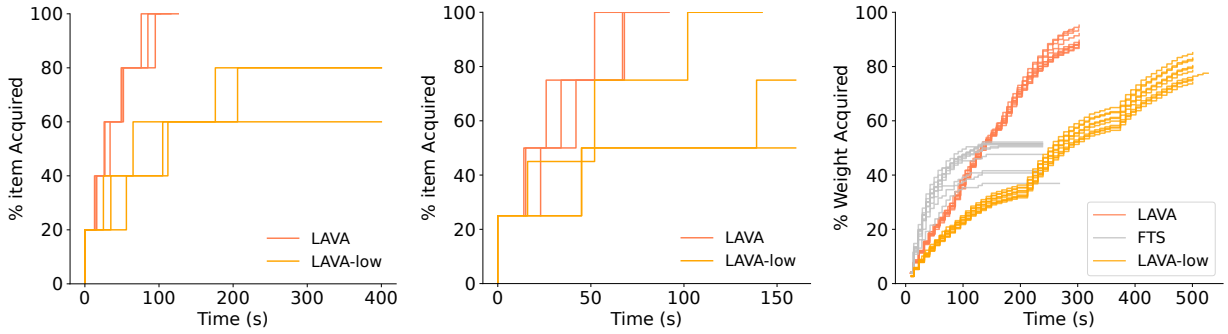


Fig. 6: Breakdown of experimental performance comparison between **LAVA**, LAVA-low, and Fixed Trajectory Scooping(FTS). $*$ represents zero-shot experiments.

C. Experimental Results

In this section, we present and analyze the experimental results. We first present the success rate of LAVA's networks. Then, following the training of the hierarchical policy, we evaluate its performance on the robot and compare it with the baseline methods. We test across varied food items and varied food configurations, including granular food cereals, liquid food water, semi-solid yogurt, deformable tofu, and multi-medium soup with tofu chunks. To assess performance, we employ the criteria of success rate, which indicates the successful scooping of food items from a bowl without spillage and breakage and successful long-horizon food acquisition by clearing the bowl efficiently. Instances where some spillage occurs are considered partial success.

*1) **LAVA**'s Network Success rates:* ScoopNet achieved 100% accuracy in choosing correct high-level primitives across 46 bowls, TargetNet accurately predicted bite targets at 87.9% over 83 instances, and DepthNet successfully determined correct spoon depths for bite sizes at 85.7% across 175 instances, demonstrating the **LAVA** networks' effectiveness in robotic-assisted feeding.



(a) 5 tofu chunks. (b) 4 tofu chunks. (c) Cereals.

Fig. 7: Individual trials comparison between **LAVA**, LAVA-low and FTS. Subfigures (a) and (b) show the comparison with different tofu configurations, and (D) show the comparison with cereals.

2) Comparison with Baselines: We evaluate the success rates of **LAVA** against two baseline models, Lava-low and FTS, across a variety of food types and scooping dynamics, as shown in Figures 6 and 7. Our evaluation focused on several key metrics: efficiency (total time taken to clear the bowl), scoop size, and spillage for granular, semisolid, and liquid foods. For deformable foods and fruit chunks, we recorded configuration, number of scoop attempts, and instances of food breakage. In particular, for complex scenarios such as soup with tofu chunks, our assessment averaged efficiency, spillage, and breakage metrics.

How do all the methods handle the challenge of scooping liquids, such as water and soup, which are prone to spillage? The analysis (Figure 6c) reveals that both baseline models struggle with the fluidity of water and soup, leading to significant spillage. The FTS model, with its fixed end-effector orientation and height, cannot adjust to the varying dynamics of liquid scooping, resulting in spillage and ineffective scooping. LAVA-low struggles as water levels decrease, showing inefficiency in maintaining adequate scoop sizes. In contrast, **LAVA** adeptly adjusts to real-time changes in food depth, achieving optimal scoop sizes and minimizing spillage for efficient bowl clearance.

What about the acquisition of more solid, yet deformable food types, such as tofu? Our findings (Figures 7a, 7b, and 6d) indicate that both baselines encounter issues with deformable foods such as tofu, often resulting in food breakage. The FTS model's rigid scooping motion damages the food, while Lava-low, despite managing to scoop, causes tofu to accumulate and break as shown by instances of food breakage in Figure 6d due to lack of strategic food prioritization based on subregions. **LAVA**, however, prioritizes tofu chunks based on their subregion, aligning them for easier access and significantly reducing breakage, mimicking human scooping strategies.

How does each method fare in preventing spillage and ensuring efficient scoop attempts with solid foods such as fruit chunks? The evaluation as visible in Figure 6c and 6a reveals that the baselines are less adept with solid, irregularly shaped foods such as fruit chunks, prone to rolling or falling off the spoon. This issue is exacerbated for fruits with curved surfaces. **LAVA**, employing an align-then-scoop strategy, ensures better alignment and significantly less spillage by adjusting to the fruit's shape for secure scooping.

We see that **LAVA** consistently outperforms the baselines, achieving higher success rates and more effective plate clearance. It surpasses FTS and Lava-low by adapting its strategy for efficient, minimal-breakage scooping across all tested food types, demonstrating the benefits of its hierarchical policy framework. As expected, FTS and Lava-low, limited by their static approaches, fail to optimize for future scooping advantages, leading to increased breakage and inefficiency, especially without considering food prioritization and arrangement strategies. **LAVA**'s comprehensive strategy ensures efficient, adaptive, and precise food acquisition, significantly improving upon the limitations of existing models.

3) Zero-shot Generalization: As detailed in Section V-A, our data collection process exclusively involved the transparent glass bowl containing granular cereals and tofu. However, we evaluated our approach to soup with tofu chunks and different food types such as liquid water and semi-solid yogurt, and solid apple chunks during testing. Remarkably, our approach demonstrates robust performance across these varied configurations, as depicted in Figure 6 and 8

Especially with soup and tofu chunks, scooping up both the solid pieces and the liquid at the same time is tricky. Our system, **LAVA**, is designed to adjust to these challenges. Despite the tofu chunks' tendency to float away from the central scooping area, **LAVA**'s adaptive strategy realigns and reorients to effectively scoop the tofu. Following the tofu acquisition, LAVA continues to adapt and clear the remaining soup, showcasing its capability to handle various food textures and types within the same meal, ensuring efficient bowl clearance.

VI. Conclusion, Limitation and Future Work

We have developed and presented a hierarchical policy framework to enhance robotic systems' capability in the acquisition of diverse food types, ranging from liquids to solids and deformable items. Our approach leverages representation learning, with sophisticated planning and execution strategies, to address the challenges associated with the

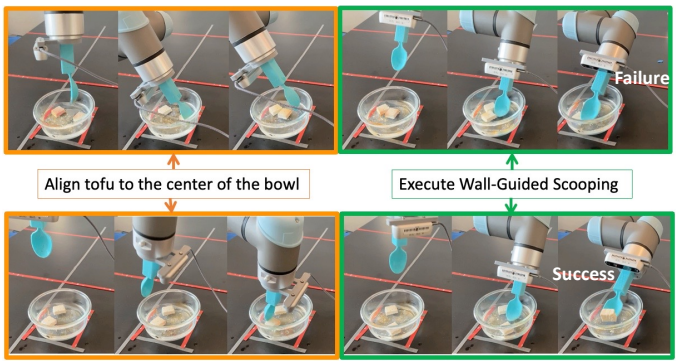


Fig. 8: Zero-shot long-horizon food acquisition: Tofu chunks in soup. Top: Spoon aligns tofu towards the bowl's center, which drifts during scooping due to fluid dynamics. Bottom: System realigns tofu to the centre and successfully scoops.

variability in food textures, sizes, and positions within the bowl. Experimental analysis shows our framework outperforms baseline models in efficiency, minimizing spillage and breakage, and adaptive food scooping, with improved success rates across various food configurations. Despite the promising result, limitations exist, particularly in handling thin, flat, or irregular foods needing specialized strategies. Future efforts will focus on broadening the action space for diverse food types and exploring efficient data acquisition methods, including using internet video resources for complex food handling.

REFERENCES

[1] S. W. Brose, D. J. Weber, B. A. Salatin, G. G. Grindle, H. Wang, J. J. Vazquez, and R. A. Cooper, "The role of assistive robotics in the lives of persons with disability," *American Journal of Physical Medicine & Rehabilitation*, vol. 89, no. 6, pp. 509–521, 2010.

[2] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh, "Learning bimanual scooping policies for food acquisition," *arXiv preprint arXiv:2211.14652*, 2022.

[3] P. Sundaresan, J. Wu, and D. Sadigh, "Learning sequential acquisition policies for robot-assisted feeding," *arXiv preprint arXiv:2309.05197*, 2023.

[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)*. IEEE, 2019, pp. 267–276.

[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 plate," in *The International Symposium of Robotics Research*. Springer, 2019, pp. 427–442.

[6] T. Bhattacharjee, G. Lee, H. Song, and S. S. Srinivasa, "Towards robotic feeding: Role of haptics in fork-based food manipulation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1485–1492, 2019.

[7] P. Sundaresan, S. Belkhale, and D. Sadigh, "Learning visuo-haptic skewering strategies for robot-assisted feeding," in *6th Annual Conference on Robot Learning*, 2022.

[8] S. Belkhale, E. K. Gordon, Y. Chen, S. Srinivasa, T. Bhattacharjee, and D. Sadigh, "Balancing efficiency and comfort in robot-assisted bite transfer," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4757–4763.

[9] R. K. Jenamani, D. Stabile, Z. Liu, A. Anwar, K. Dimitropoulou, and T. Bhattacharjee, "Feel the bite: Robot-assisted inside-mouth bite transfer using robust mouth perception and physical interaction-aware control," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 2024, pp. 313–322.

[10] Y. Niu, S. Jin, Z. Zhang, J. Zhu, D. Zhao, and L. Zhang, "Goats: Goal sampling adaptation for scooping with curriculum reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1023–1030.

[11] E. K. Gordon, R. K. Jenamani, A. Nanavati, Z. Liu, H. Bolotski, R. Karim, D. Stabile, A. Kashyap, B. H. Zhu, X. Dai *et al.*, "An adaptable, safe, and portable robot-assisted feeding system," *arXiv preprint arXiv:2403.04134*, 2024.

[12] https://meetobi.com/. (2023). [Online]. Available: https://meetobi.com/

[13] X. Lin, C. Qi, Y. Zhang, Z. Huang, K. Fragkiadaki, Y. Li, C. Gan, and D. Held, "Planning with spatial-temporal abstraction from point clouds for deformable object manipulation," *arXiv preprint arXiv:2210.15751*, 2022.

[14] M. Dalal, D. Pathak, and R. R. Salakhutdinov, "Accelerating robotic reinforcement learning via parameterized action primitives," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 847–21 859, 2021.

[15] S. Nasiriany, H. Liu, and Y. Zhu, "Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7477–7484.

[16] T. Bhattacharjee, G. Lee, H. Song, and S. S. Srinivasa, "Towards robotic feeding: Role of haptics in fork-based food manipulation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1485–1492, 2019.

[17] Z. Zhu and H. Hu, "Robot learning from demonstration in robotic assembly: A survey," *Robotics*, vol. 7, no. 2, p. 17, 2018.

[18] Z. Xie, Q. Zhang, Z. Jiang, and H. Liu, "Robot learning from demonstration for path planning: A review," *Science China Technological Sciences*, vol. 63, no. 8, pp. 1325–1334, 2020.

[19] C. Lauretti, F. Cordella, E. Guglielmelli, and L. Zollo, "Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1375–1382, 2017.

[20] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 391–398.

[21] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 639–646.

[22] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[23] R. Chitnis, D. Hadfield-Menell, A. Gupta, S. Srivastava, E. Groshev, C. Lin, and P. Abbeel, "Guided search for task and motion plans using learned heuristics," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 447–454.

[24] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu, "Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks," *The International Journal of Robotics Research*, p. 02783649231219020, 2023.

[25] S. Pateria, B. Subagdja, A.-h. Tan, and C. Quek, "Hierarchical reinforcement learning: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–35, 2021.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[28] A. Beck and S. Sabach, "Weiszfeld's method: Old and new results," *Journal of Optimization Theory and Applications*, vol. 164, pp. 1–40, 2015.

[29] E. Weiszfeld and F. Plastria, "On the point for which the sum of the distances to n given points is minimum," *Annals of Operations Research*, vol. 167, pp. 7–41, 2009.

[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.