SCONE: A Food Scooping Robot Learning Framework with Active Perception

Anonymous Author(s)
Affiliation
Address
email

Abstract: The ability to successfully scoop up food items presents a significant challenge for existing robot systems due to the complex states and physical properties of food. To overcome this challenge, we believe it is crucial to encode food items into task-related and meaningful representations. However, the distinctive properties of food items, including deformability, fragility, fluidity, or granularity, pose significant challenges for existing representations. In this paper, we investigate the potential of active perception for learning meaningful food representations in an implicit manner. To this end, we present SCONE, a food scooping robot learning framework that leverages the representations gained from active perception to provide the food-scooping model. SCONE consists of two essential encoding modules: the interactive encoder and the state retrieval module. The encoding process allows the model to capture the characteristics of food items and essential features of states. In the real-world food scooping experiments, SCONE achieves 71% task success rate on 6 unseen food items under 3 levels of difficulty, outperforming other baselines. Additionally, it exhibits higher stability, as evidenced by the task success rate of each food item surpassing 50%.

Keywords: Food Manipulation, Robot Scooping, Active Perception

1 Introduction

The potential of robot scooping skills in food manipulation holds great promise for enhancing the quality of human life. These skills find applications in diverse areas such as food preparation and assistive feeding [1, 2, 3, 4, 5, 6, 7], benefiting environments ranging from homes and hospitals to restaurants. Enabling the robot to successfully scoop up food items presents a significant challenge due to the wide range of appearances and physical properties exhibited by different foods. These characteristics cannot be easily defined or comprehensively accounted for in advance during preparation. In practical applications, designing a framework that enables robots to acquire the necessary scooping skills for manipulating unseen food items remains a challenging problem.

Current products and frameworks [1, 2, 3, 7, 8, 9] often use hard-coded primitive actions to solve real-world food manipulation problems. However, these methods often depend on predefined trajectories or specialize in specific food categories, limiting their ability to generalize to new target items. Several data-driven learning frameworks for assistive feeding [4, 10, 11, 12, 13] assume that food items with similar visual appearance will exhibit similar success rates when adopting the same policy. However, due to the diverse states of food encountered in the scooping task, the generalization of these frameworks is still limited when dealing with fragile, liquid, or granular items. The complex nature of these food states presents challenges in accurately defining segmenting masks or bounding boxes for detection, thus making the formulation of food representations a challenging task.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

Figure 1: Real World Robot Food Scooping Task

The dynamic feedback obtained through the interaction process could effectively represent foods in an implicit manner. The implicit representations enable the model to capture essential features and characteristics of various states of foods without relying on explicit definitions or predefined features. As a result, learning frameworks with active perception [14, 15, 16], by recognizing objects' characteristics through interaction, exhibits considerable potential in addressing the generalization challenges encountered in real-world food manipulation [17, 18, 19, 20, 21, 22]. These frameworks consist of two stages. The first is the *Interacting* stage, where the robot interacts with the target objects to receive the dynamical sensing feedback. Meaningful messages about the target items can be extracted through supervised or self-supervised training. In the following *Manipulation* stage, the policy model can exhibit higher efficacy by leveraging the additional information acquired from the *interacting* stage. The pipeline has been widely used in food classification, policy selections [20, 22, 23], and dynamics inference [24, 17, 18]. Through active perception, we can implicitly encode food items into representations that capture their properties or task-related information.

To this end, we propose SCONE: a food SCOoping robot learNing framEwork with active perception. In our framework, we utilize active perception to improve the target manipulation tasks in a coarse *plus* fine manner: in addition to the overall dynamics feedback, which represents the general food properties, we also allow the model to attend to local state information at each time frame in the *Interaction* stage and aggregate them for predicting the manipulation action. Building upon this intuition, our method encodes active perception data with two separate modules: one is the interactive encoder, which encodes the overall sequential interactive observations into latent representations. The other is the state retrieval module, capturing the critical information of states from interaction based on the current state. Finally, a task-related embedding is obtained by fusing these two encoding modules. This embedding, along with the current observations, is then used to guide the closed-loop policy.

We evaluate our method on a real-world robotic food scooping platform and demonstrate its superior performance over the baselines. Overall, our method achieves a 71% task success rate when tested with six unseen food categories under three different levels of difficulty. Furthermore, the results demonstrate both the stability and generalization of the approach, with task success rates consistently exceeding 50%. Moreover, we visualize the learned latent embeddings from the interactive encoder and the attention scores from the state retrieval module. We show that the physical properties of food can be extracted and the task-related information can be retrieved at each time step. Our main contributions are as follows:

- We propose SCONE: a food scooping robot learning framework with active perception, which utilizes *interaction* data with global encoding and state retrieval module, to improve the generalization in real-world food scooping tasks.

- The visualization of latent embeddings indicates that meaningful knowledge can be extracted, and manifestly presents how the active perception facilitate the policy learning.
- In real-world experiments, the framework can succeed at manipulating unseen food items under different settings, outperforming the baselines.

2 Related Work

Robotic Food Scooping. Recent works have explored quite a number of robot food scooping frameworks for assistive feeding and food preparation tasks. Several commercial meal-assistance robots [8, 9] rely on pre-defined trajectories and need human-assigned controls, which are more stable but with limited autonomy. Ohshima et al. [1] introduce an autonomous robot system that can scoop up non-rigid solid foods with the support of Laser Range Finder, by tackling the recognition problem of remnants of fragile foods. However, it is still limited to the geometry and deformability of food items. Park et al. [2] presents the active feeding framework, in which users select the tool-use actions and the robot scoops up the foods. Without considering the properties of each food, this may encounter limitations in scalability. Grannen et al. [4] proposed a bimanual scooping framework that utilizes closed-loop visual feedback to identify states where food breakage is likely to occur. This approach helps prevent food breakage and ensures the successful scooping of food items. However, the visual detection mechanism with masks in their framework still limits the range of food states. In this work, we explore the potential of active perception to represent foods in an implicit manner, offering an alternative approach for generalizing to diverse states of food.

Active Perception in Closed-loop Policy Learning. Prior works have utilized pre-interacting data for predicting object physical properties, reconstructing the object geometries, and executing open-loop control for manipulation tasks [19, 20, 21, 22, 25, 23]. However, it remains largely unexplored how the pre-programmed interaction can improve closed-loop policy learning in an implicit manner. Saito et al. [17] propose a sensorimotor dynamical system model for the pouring task. The interaction and manipulation sequences are concatenated as a series of actions. A recurrent neural network is used to learn the dynamical transition. In the testing stage, robots observe how the liquid behaves with pre-programmed shaking actions, and generate the corresponding pouring motions. In their follow-up work, Saito et al. [18] apply a similar architecture for tool selection and tool-use tasks. To improve the capability of the dynamical model, they adopt the multiple timescale recurrent neural network with updatable initial parameters, which can be explored during interaction in the testing stage. When dealing with unseen objects, tool–object–action relations can be captured through the matching process. In contrast, our work explores an alternative approach to improve the connection and utilization of interaction data in the manipulation process. We introduce the state retrieval module, which is designed to extract the most crucial information from each step in the closed-loop process and enable a more efficient and direct utilization of the interaction data.

3 Method

Our goal is to design a learning framework that enables real-world robot scooping. We develop our method based on active perception and focus on how to utilize the interaction data to improve task success under dynamic situations. Moreover, we adopt the learning from demonstration (LfD) paradigm to enable the robot to learn using a spoon in a faster and more reliable manner. We assume a demonstration dataset is available for model training. Figure 2 shows an overview of our proposed framework.

3.1 Problem Formulation

Task Definition. In our work, the food scooping task is divided into three distinct stages: the interacting stage, the scooping stage, and the transferring stage. During the interacting stage, the robot employs a spoon to stir the contents of the container while observing the dynamics of the food items. In the scooping stage, the robot's objective is to scoop up the food items successfully. Finally,

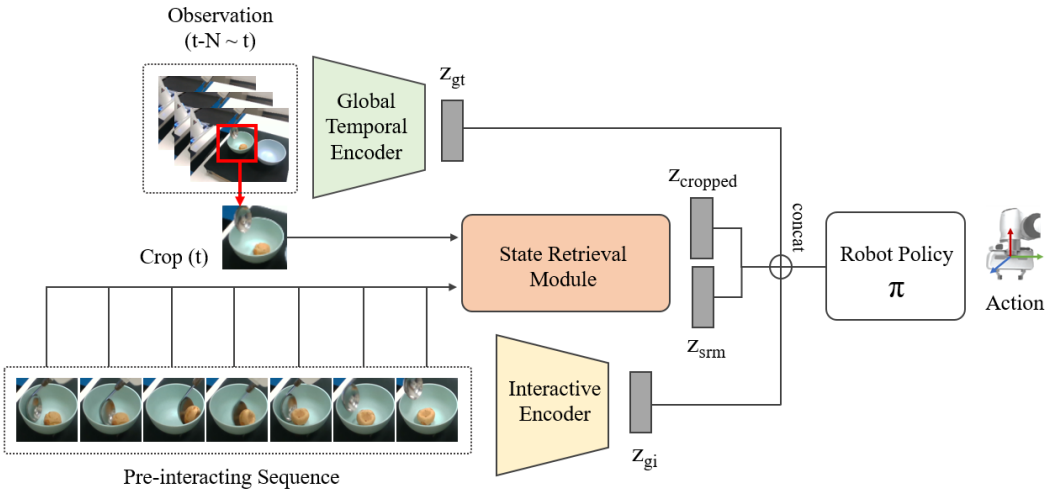

Figure 2: **SCONE** is a real-world robotic food scooping learning framework. With active perception, the generalization capability of the model can be enhanced. SCONE consists of three main feature extraction modules: (a) **Global temporal encoder**, which is able to capture the stage information from the current observations. (b) **Interactive encoder**, which effectively gathers information and integrates it to form a holistic understanding of the physical properties of the food items. (c) **State retrieval module**, which explores and extracts the task-related state information based on current state to facilitate manipulation.

in the transferring stage, the robot transfers the scooped food to another container. In this work, we focus on learning the challenging scooping process exclusively, while keeping the interacting and transferring stages fixed with pre-defined motions. Task success is determined by specific criteria: the food items on the spoon should cover over a one-third area or more than one entity without spillage and damage. The spoon must remain at a position higher than the container after scooping motions are performed.

Pre-processing of Interaction Data. To extract the information of food items with greater precision and efficiency, we sample K frames (K = 7 in our case) of observed data from sequences received in the interacting stage $D^{\texttt{origin}}_{\texttt{interact}} = \{(o_i, e_i)|i = 1, \ldots, N\}$. We also crop the viewing area around the container. The sampling starts at the time step when the spoon reaches the lowest position and acquires the data every 0.5 seconds. The pre-processed data denotes $D_{\texttt{interact}} = \{(o_i, e_i)|i = 1, \ldots, K\}$, which has sensor readings $o_i = (I_{\texttt{RGB}}, I_{\texttt{Depth}})$ and the robot end-effector poses e_i.

SCONE Architecture (Fig. 2). At each time step, we have the interaction data $D_{\texttt{interact}}$ and the N-length history of observation data $O_t = \{(o_i, e_i)|i = t - N, \ldots, t\}$. They are the inputs to the three encoding modules: global temporal encoder $f_{\texttt{gt}}$, interactive encoder $f_{\texttt{gi}}$, and the state retrieval module $f_{\texttt{srm}}$ to obtain latent vectors $z_{\texttt{gt}}$, $z_{\texttt{cropped}}$, $z_{\texttt{gi}}$ and $z_{\texttt{srm}}$. We aim to learn the policy $\pi(a_i|z_{\texttt{gt}}, z_{\texttt{cropped}}, z_{\texttt{gi}}, z_{\texttt{srm}})$ that minimizes the standard policy learning loss. In this work, we only control the 3D position and the y-axis pitch rotation of the end-effector, and the action $a_i \in R^4$ output from the policy at each time step is formulated as the delta between the current and next end-effector pose, i.e., e_i and e_{i+1}.

3.2 Global Temporal Encoder

The aim of the global temporal encoder is to capture the progressive information from the observations, guiding the policy to be aware of the current phase in the task progress. We use two designed convolution neural networks to extract feature vectors of a certain dimension from the overall observations of $I_{\texttt{RGB}}$ and $I_{\texttt{Depth}}$. To obtain the temporal information, we encode the past N frames of observations ($N = 10$ in our case), so at each time step, the inputs to the observation encoding module are $o_{\texttt{observe}} = (I_{\texttt{RGB}} \in R^{N\times W\times H\times 3}, I_{\texttt{Depth}} \in R^{N\times W\times H\times 1})$. These two sequences of

different modalities pass through the corresponding encoders $f_{\texttt{RGB}}$ and $f_{\texttt{Depth}}$ to obtain two sets of feature embeddings. We concatenate them on the last dimension. Then, we flatten and downsample the feature vectors through one fully-connected layer, obtaining global temporal features of current observation $z_{\texttt{gt}}$. This would be the overall guidance for the policy model to output actions.

3.3 Interactive Encoder

To extract a representation of food items through pre-programmed interaction, we introduce the interactive encoder f_{gi}, which consists of an RGB encoder, a depth encoder, and a self-attention module. The sequential data of two modalities $o_{\texttt{interact}} = (I_{\texttt{RGB}} \in R^{K\times W\times H\times 3}, I_{\texttt{Depth}} \in R^{K\times W\times H\times 1})$ are first passed over the convolution encoder separately, and then concatenated into latent vectors $z_{\texttt{interact}} \in R^{K\times D}$. Afterward, the concatenated embeddings are fed into the self-attention module, implementing cross-modality and cross-time attention to catch the critical information, outputting final flattened latent vectors z_{gi}. The z_{gi} contains the characteristics of food items in the latent space that can be treated as guidance for the policy model to recognize the similarity between seen and unseen objects.

3.4 State Retrieval Module

We assume that the state of each frame in the interacting stage has a different importance to the state of each time step in the manipulating stage, so the objective of the state retrieval module (SRM) is to retrieve meaningful embeddings from the sequences of interaction data queried on current observations. To obtain meaningful features for guidance, the state feature extraction layers and the cross-attention mechanism are applied.

To extract the features more precisely, we crop the current observation around the container to obtain $o_t^{\texttt{crop}}$, which is the same as the pre-processing of the interacting data. First, the RGB-D images in $o_t^{\texttt{crop}}$ are fed into designed convolution layers to output the feature embeddings separately, and then concatenated with the corresponding end-effector poses, acquiring the state pairs $z_{\texttt{cropped}} \in R^{1\times D}$ and $z_{\texttt{interact}} \in R^{7\times D}$. Next, the attention mechanism is applied to generate an attention map between $z_{\texttt{cropped}}$ and $z_{\texttt{interact}}$. The final feature maps are the result of matrix multiplication of the attention map and each frame features in $z_{\texttt{interact}}$. Furthermore, a fully connected layer is used to downsample the output latent vectors into lower dimensions. The output of the state retrieval module $z_{\texttt{srm}}$ extracts task-related state information from the pre-interacting data, and implicitly guides the policy model to adapt the policy considering the target item and the interaction experience. Concurrently, the $z_{\texttt{cropped}}$ are treated as another input to the policy model.

3.5 Policy Learning

To learn the policy $\pi(a_i|z_{\texttt{gt}}, z_{\texttt{cropped}}, z_{gi}, z_{\texttt{srm}})$, we adopt Behavior Cloning (BC) to train action prediction from the observation in a supervised manner. The model takes the latent vectors of the global observation, the local observation, the overall food representation, and the retrieved cue under the current state as inputs passes over a three-layer Multi-Layer Perceptron, and then outputs the corresponding actions. We use the mean square loss between the ground truth action $\hat{a}$ and predicted action a for training the network.

4 Evaluation

4.1 Food Scooping Dataset

Hardware setting. The data collection platform consists of a 7DoF Franka Emika Panda robot with the default gripper, a D435 RealSense camera fixed around the platform, and two containers placed in the center of the table. During the data collection, we receive sensory data of RGB-D images and the robot proprioception simultaneously. The robot controller runs at a 10 Hz frequency.

Food Categories. In the training stage, we select six food items (i.e., brown rice, mung bean, soybean, chocolate ball, dried jujube, and cheese ball) that encompass different particle sizes, shapes, and weights. Additionally, we evaluate on six unseen food items, i.e., orange, red bean, sago, macadamia, penne, and fruit candy. To assess the generalization ability of our proposed method, we design three difficulty levels. The *basic* configuration closely mimics the training data, with the container being filled with two-thirds full of food items. These primarily comprise small particles, accompanied by a smaller quantity of larger particle foods. The *extended* setting reduces the number of small particles and increases the number of large particles. The *peculiar* setting includes foods with significantly different appearances and weights compared to those in the training set. These difficulty levels are designed to have a comprehensive assessment of the model's capability to handle a wide range of scenarios.

Data Collection. To effectively capture the intricate and complex scooping operations, we utilize the hand-guiding mode, where a human kinesthetically controls the robot to perform the task while simultaneously recording the trajectory of the end-effector. We collect 10 demonstrations for each food category. Each demonstration consists of 6 seconds of interaction, 15 seconds of scooping motions, and 10 seconds of transferring actions. The recorded interacting motions are subsequently replayed to collect RGB-D images without humans in the scene. Each trial is meticulously reviewed to verify the viability of the recorded trajectory. A trial is considered a qualified demonstration if the robot successfully scoops up the food items. In total, we collect 60 real-world demonstrations involving six distinct food items.

4.2 Baselines

We conduct a comparison with seven different approaches, which encompass various BC-based methods and template-based methods. The template-based approaches employ diverse strategies for incorporating interaction data. Please see Supplementary Material for the details of each baseline.

BC-based Methods: The methods include the standard BC trained on both scooping and transferring data (**one stage**), standard BC trained w/o interaction data, BC with interaction, and the BC conditioned on food categories represented by a one-hot vector.

Template-based Methods: The method treats the learning process as a template selection problem. The category-level templates are obtained through the mean of all trajectories that belong to the same category. The experiments proceed with selecting templates **randomly** or **pre-defined** with the trained classifier.

MTRNN [18]: The baseline strings the interacting and the scooping stage as one action sequence. The model trains a multi-timescale recurrent neural network to learn the corresponding system.

4.3 Real-world Food Scooping Experiment

The experimental results are presented in Table 1. Overall, our proposed method achieves a task success rate of 71% across three difficulty levels. Additionally, our framework exhibits consistent stability and generalization across all unseen food items.

Basic Setting. BC baselines (**one stage** and **w/o interact**) perform unsatisfactory in this setting. The performance is enhanced by incorporating additional information extracted from the interacting data, which improves the success rate by more than 10%, compared to the basic BC baselines. Compared to BC-based methods, template-based methods demonstrate enhanced stability. However, in certain cases, the predictions from the classifier may lead to the selection of an unsuitable template, resulting in failure instances. For instance, when encountering the food item **Sago**, the model predicts the wrong answers, and the foods are spilled out. The MTRNN and the SCONE achieve over 80% success rate in each category, while there are failure cases caused by unstable robot arm movements or other sources of uncertainties.

Extended Setting. The overall task success rate declines as a result of disparities between the testing and training environments, posing challenges in accurately assessing the states of the target

	Basic				Extended				Peculiar		
	Orange	Red Bean	Sago	Macadamia	Orange	Red Bean	Sago	Macadamia	Penne	Candy	**Success**
BC (one stage)	1/10	0/10	0/10	1/10	0/10	0/10	0/10	0/10	1/10	0/10	**3%**
BC (w/o interact)	3/10	0/10	2/10	0/10	5/10	0/10	3/10	2/10	2/10	4/10	**21%**
BC (w/ food id)	7/10	2/10	0/10	2/10	0/10	0/10	3/10	0/10	4/10	3/10	**21%**
BC (w/ interact)	7/10	5/10	0/10	3/10	3/10	0/10	3/10	1/10	5/10	1/10	**28%**
Rand. Template	5/10	4/10	6/10	8/10	0/10	0/10	3/10	0/10	3/10	0/10	**29%**
Classified Template	**10/10**	**10/10**	0/10	**10/10**	3/10	0/10	**10/10**	3/10	1/10	0/10	**47%**
MTRNN [18]	**10/10**	**10/10**	**10/10**	8/10	4/10	2/10	0/10	3/10	**8/10**	3/10	**58%**
SCONE (Ours)	**10/10**	8/10	9/10	9/10	**7/10**	**5/10**	6/10	**5/10**	7/10	**5/10**	**71%**

Table 1: Real World Experiment Results.

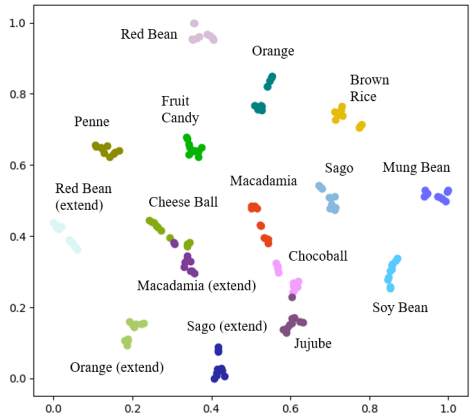

Figure 3: 2D t-SNE of Interactive Embeddings.

Global Temporal Encoder	Interactive Encoder	Local Latent	SRM	Success Rate
✓				**21%**
✓	✓			**28%**
✓	✓	✓		**48%**
✓	✓	✓	✓	**71%**

Table 2: Ablation Study.

items and leading to failed attempts at food scooping. Conversely, our proposed framework excels at leveraging the precise state information obtained during the interaction stage, resulting in improved performance. The template-based method (Classified Template) maintains stability when the appropriate template is selected.

Peculiar Setting. The task success rate of template-based methods significantly decreases in comparison to previous settings, highlighting the effectiveness and robustness of the close-loop learning-based approach. The methods with interaction outperform other baselines because of their enhanced understanding of food items. SCONE achieves the highest total success rate, which can be attributed to its ability to accurately capture task-related information and establish an effective connection between the pre-interaction state and the subgoal state during the manipulation stage.

4.4 Ablation Study

We compare four variants of our model to showcase the effectiveness of our architectural design. The results are shown in Table 2. The base model achieves a success rate of 21%. However, when interactive data is incorporated into the learning process, the performance improves to 28%. This improvement can be attributed to the increased understanding of the target food items. Moreover, incorporating the current observation, which emphasizes the model's attention on the crucial stage, results in a substantial improvement of approximately 20% in the task success rate. Lastly, the inclusion of the state retrieval module significantly enhances the task success rate, resulting in an

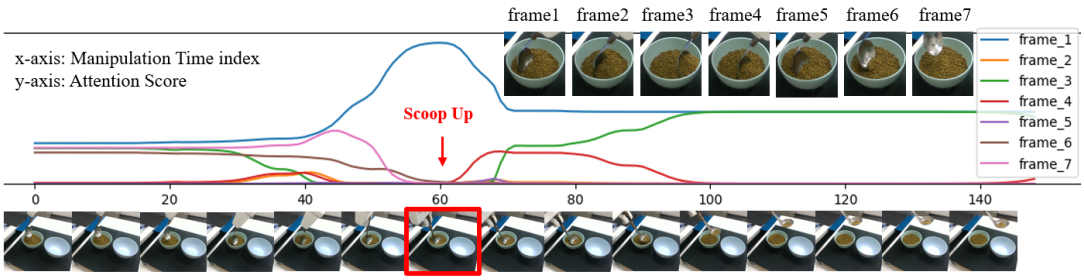

Figure 4: Visualization of Attention Scores.

impressive achievement of 71%. This notable improvement is attributed to the module's ability to accurately extract and utilize critical information from the interacting data.

4.5 Visualization

Interactive Embeddings. Given the interaction data, the interactive encoder encodes the dynamic visual data into meaningful and informative features. To interpret the behavior of the interactive encoder, we utilize t-SNE projections to visualize the interactive embeddings. In Figure 3, we show that food items belonging to the same category or possessing similar properties tend to cluster together. This result suggests that the interactive encoder effectively captures and represents the inherent similarities among the food items.

Attention Scores from State-retrieval Module. We visualize the attention scores during the manipulation in Figure 4. The x-axis represents the time index in the manipulation stage, and the y-axis is the attention scores assigned to each pair of states. The visualization shows that the model, equipped with the cross-attention mechanism, effectively identifies and captures the relative importance of states within the interacting data. The extracted features serve as cues that play a significant role in enhancing the manipulation. The visualization analysis also highlights that the attention score changes according to a consistent pattern across trials, demonstrating the model's ability to automatically capture and understand task phases. Notably, even without explicit labeling, the model demonstrated an inherent ability to capture and recognize important information related to different phases of the task. More comprehensive analysis can be found in the Supplementary Material.

5 Conclusion

We proposed SCONE, a robot food scooping learning framework with active perception, which utilizes the interacting data in a coarse plus fine manner: one is the global encoding, which captures the overall physical properties of target food items; the other is the state retrieving, which retrieves the critical state information based on the current state. Our framework possesses the ability to dynamically adapt and adjust the scooping policy based on the current observation, and by leveraging the interacting data, the model is able to adopt policy for the previously unseen food items, resulting in improved performance and generalization capabilities.

Limitations and Future Work. Our proposed method heavily relies on extracting critical information from active perception, so the design of pre-programmed interacting motions plays a crucial role in achieving successful results. In our work, we have focused on two main aspects when designing the interacting motions: (a) Ensuring that the actions performed during active perception do not break or damage the food items. (b) The motions during for active perception should provide intuitive and relevant information to the model. However, the current approach of human-hard-coded-defined actions may limit the exploration of other possibilities. In future work, we believe that there is room for further exploration in the design of motions for interacting. By making the design more explorable, we can potentially uncover more suitable interacting motions that enhance the overall performance of the system.

References

[1] Y. Ohshima, Y. Kobayashi, T. Kaneko, A. Yamashita, and H. Asama. Meal support system with spoon using laser range finder and manipulator. In *2013 IEEE workshop on robot vision (WORV)*, pages 82–87. IEEE, 2013.

[2] D. Park, Y. Hoshi, H. P. Mahajan, H. K. Kim, Z. Erickson, W. A. Rogers, and C. C. Kemp. Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned. *Robotics and Autonomous Systems*, 124:103344, 2020.

[3] L. P. Ureche and A. Billard. Constraints extraction from asymmetrical bimanual tasks and their use in coordinated behavior. *Robotics and autonomous systems*, 103:222–235, 2018.

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

[5] H. Ochi, W. Wan, Y. Yang, N. Yamanobe, J. Pan, and K. Harada. Deep learning scooping motion using bilateral teleoperations. In *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 118–123. IEEE, 2018.

[6] S. Gushi, Y. Shimabukuro, and H. Higa. A self-feeding assistive robotic arm for people with physical disabilities of the extremities. In *2020 5th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pages 61–64. IEEE, 2020.

[7] D. Park, Y. Hoshi, and C. C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3): 1544–1551, 2018.

[8] Mealtime partners. URL https://www.mealtimepartners.com/. [Online; accessed 8-June-2023].

[9] Meet obi. URL https://meetobi.com/. [Online; accessed 8-June-2023].

[10] 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*, 4(2):1485–1492, 2019.

[11] 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 *Robotics Research: The 19th International Symposium ISRR*, pages 427–442. Springer, 2022.

[12] E. K. Gordon, X. Meng, T. Bhattacharjee, M. Barnes, and S. S. Srinivasa. Adaptive robot-assisted feeding: An online learning framework for acquiring previously unseen food items. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9659–9666. IEEE, 2020.

[13] E. K. Gordon, S. Roychowdhury, T. Bhattacharjee, K. Jamieson, and S. S. Srinivasa. Leveraging post hoc context for faster learning in bandit settings with applications in robot-assisted feeding. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10528–10535. IEEE, 2021.

[14] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42:177–196, 2018.

[15] Q. V. Le, A. Saxena, and A. Y. Ng. Active perception: Interactive manipulation for improving object detection. *Standford University Journal*, 2008.

[16] D. P. Benjamin, D. Lonsdale, and D. M. Lyons. Designing a robot cognitive architecture with concurrency and active perception. In *AAAI Technical Report (5)*, pages 1–8, 2004.

[17] N. Saito, N. B. Dai, T. Ogata, H. Mori, and S. Sugano. Real-time liquid pouring motion generation: End-to-end sensorimotor coordination for unknown liquid dynamics trained with deep neural networks. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1077–1082. IEEE, 2019.

[18] N. Saito, T. Ogata, S. Funabashi, H. Mori, and S. Sugano. How to select and use tools?: Active perception of target objects using multimodal deep learning. *IEEE Robotics and Automation Letters*, 6(2):2517–2524, 2021.

[19] T. Lopez-Guevara, R. Pucci, N. K. Taylor, M. U. Gutmann, S. Ramamoorthy, and K. Suhr. Stir to pour: Efficient calibration of liquid properties for pouring actions. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5351–5357. IEEE, 2020.

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

[21] 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)*, pages 409–416. IEEE, 2019.

[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*, pages 638–645. IEEE, 2014.

[23] A. Sawhney, S. Lee, K. Zhang, M. Veloso, and O. Kroemer. Playing with food: Learning food item representations through interactive exploration. In *Experimental Robotics: The 17th International Symposium*, pages 309–322. Springer, 2021.

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

[25] S. Lu, R. Wang, Y. Miao, C. Mitash, and K. Bekris. Online object model reconstruction and reuse for lifelong improvement of robot manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1540–1546. IEEE, 2022.