

# Learning to Double Guess: An Active Perception Approach for Estimating the Center of Mass of Arbitrary Objects

Shengmiao Jin, Yuchen Mo, Wenzhen Yuan<sup>1</sup>

**Abstract**—Manipulating arbitrary objects in unstructured environments is a significant challenge in robotics, primarily due to difficulties in determining an object’s center of mass. This paper introduces U-GRAPH: Uncertainty-Guided Rotational Active Perception with Haptics, a novel framework to enhance the center of mass estimation using active perception. Traditional methods often rely on single interaction and are limited by the inherent inaccuracies of Force-Torque (F/T) sensors. Our approach circumvents these limitations by integrating a Bayesian Neural Network (BNN) to quantify uncertainty and guide the robotic system through multiple, information-rich interactions via grid search and a neural network that scores each action. We demonstrate the remarkable generalizability and transferability of our method with training on a small dataset with limited variation yet still perform well on unseen complex real-world objects.

## I. INTRODUCTION

With the growing interest in robotics manipulation in the wild, researchers have been investigating ways for robots to interact with different objects. A key factor in achieving a secure grasp is the proximity of the grasp point to the object’s center of mass (CoM). Yet, not much research has been trying to address a generalizable approach to estimate the CoM of arbitrary objects. Much recent research on grasping has been focused solely on grasping onto the object [1] [2], often overlooking how the choice of grasping points and the physical properties of objects affect successful manipulation. It has been noted in works like [3] and [4] that improper grasping points or poses on objects with certain properties can lead to failures, typically due to rotational and incidental slips. Knowing the CoM of the object, on the other hand, can help decide a more stable and robust grasping strategy, and ensure safety during manipulation.

This work provides a framework to perceive an arbitrary object’s CoM. The most intuitive solution is using the Force-Torque (F/T) sensor’s reading and solving CoM analytically. However, the physical limitations of F/T sensors provide inaccurate readings caused by the measurement noise and in-hand slips, therefore analytical solutions often fail. Additionally, analytical solutions are unable to provide uncertainty measurement, making it hard to estimate an informed action to improve the measurement. Instead, we turn to a data-driven solution to learn the correlation between the F/T Reading and the CoM locations with neural networks. Moreover, a vertical grasping pose prevents obtaining the location of CoM on the z-axis. Hence, we cannot predict precisely enough with a single F/T reading. Our strategy incorporates active

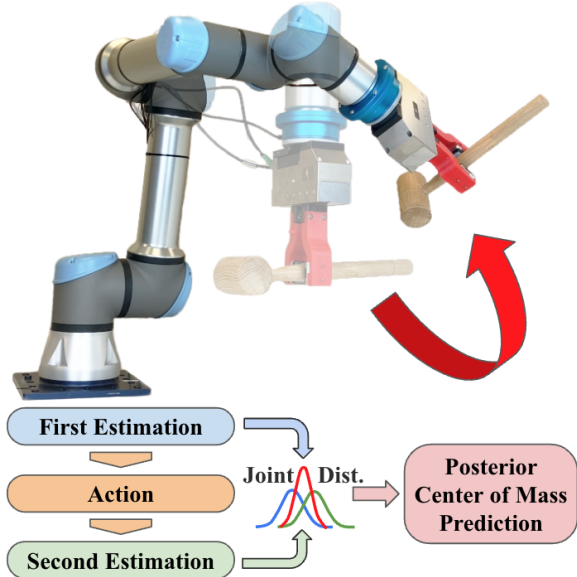


Fig. 1: We design an active perception algorithm to estimate the center of mass of arbitrary objects. Our algorithm uses the first estimation from the F/T reading to infer a new rotational orientation that improves the estimation, then executes the action and estimates again with a second F/T reading.

perception techniques to enhance accuracy, which identifies the orientation that maximizes information gained from the initial grasp. Our methodology involves actively guiding the robot to adjust its orientation which will provide a more accurate prediction.

Therefore, we propose **U-GRAPH**: Uncertainty-Guided Rotational Active Perception with Haptics to address the difficulty found in the CoM Estimation problems as illustrated in Fig. 1. We construct a Bayesian Neural Network (BNN) [5] for uncertainty quantification, obtaining a standard measurement deviation aligned with physical intuition. First, the robot picks up the object with a fixed orientation, providing a prior estimation of the CoM. With the estimation’s mean and standard deviation, we employ ActiveNet to infer the orientation that can provide the greatest information gain. From the subsequent orientation, the BNN can provide a new estimation of CoM from the latest F/T reading.

Our system is trained on 204 grasp trials each with 100 rotations with 20 weight distributions. Despite being trained only on two customized simple data collection objects, we demonstrate its ability to generalize to any arbitrary rigid object in real life. It shows an average of 1.47-centimeter error with a 7.6% error on unseen complex real-world objects in a zero-shot transfer manner. Our system shows

<sup>1</sup> University of Illinois Urbana-Champaign  
{jin45, yuchenm7, yuanwz}@illinois.edu

generalizability to all kinds of objects with different contact geometry, surface friction, and overall shapes.

## II. RELATED WORKS

### A. Physical Property Estimation

A critical premise for the successful manipulation of different objects is an understanding of their physical properties. For example, liquid properties were well-studied by previous research [6] [7] [8]. Moreover, many papers also showed that with perception, the precision and accuracy of manipulation can be increased [9] [10] [11]. Rigid body properties are also important for manipulation. Zeng et al. used residual physics to help decide a better-tossing policy for a wide range of objects [12]. Wang et al. demonstrated their algorithm can learn implicit properties and improves their swing policy with tactile explorations [13]. Murooka et al. embedded physical reasoning into manipulation skills [14]. Most previous works only consider physical property as a vague or distilled representation, while we focus on estimating the explicit measurement of the center of mass.

### B. Center of Mass Estimation

A more related set of works targeted directly at the center of mass estimation. Hyland et al. utilized iterative pushing to find a 2-dimensional CoM [15]. McGovern et al. pointed out that with reinforcement learning of stacking random shape objects in a simulator, they can estimate the CoM [16]. McGovern and Xiao also proposed a Reinforcement Learning pipeline in the real world to estimate the CoM of utensils with torque-sensing [17]. However, to our knowledge, none of the methods are generalizable to find the 3-dimensional CoM for arbitrary objects in real life, which is the problem we are trying to solve.

### C. Active Perception

Humans naturally possess the ability to explore an object actively with touch [18]. Inspired by this, many works have studied active perception in robotics with haptic or tactile sensing. Xu et al. employed active tactile perception to classify objects and showed improvement in both accuracy and efficiency [19]. Uttayopas et al. utilized active haptic sensing to classify objects with different properties [20]. Kuzliak et al. designed a framework to interactively learn the physical properties of an object with informed action selection [21]. The works mentioned above only concern a discrete action space, but in our problem setting, we have a continuous action space. Yuan et al. used active perception to decide the next best grasping location to help increase accuracy for clothes material classification [22]. Ketchum used active exploration on a scene to understand its property and shows that with active planning the exploration can be performed better [23]. In our work, on the other hand, instead of classification, we are interested in regression tasks with continuous action space and aim to solve the CoM estimation with only two actions.

## III. METHOD

Targeting a generalized and robust CoM estimation framework, we propose U-GRAPH: Uncertainty-Guided Rotational Active Perception with Haptics. This system incorporates a BNN that processes 6-dimensional force-torque readings and 2-dimensional orientation data to yield a 3-dimensional CoM estimation. U-GRAPH also features ActiveNet, which utilizes the output from the BNN to determine the next best action. Assuming the robot has already grasped the object, we perform two measurements at different orientations to accurately estimate its CoM. The BNN supplies both prior predictions and quantifies uncertainty through the standard deviation. The ActiveNet takes in prior estimation distribution and uses grid search to calculate a score for each action to determine the best one. Specifically, the action space is the 2-dimensional orientation of the grasping pose. We define the action executed as changing the pose. In the subsequent subsections, we discuss an intuitive physics model, introduce the individual modules of this framework, and present the implementation of online inference.

### A. Intuitive Model of Arbitrary Object's Center Of Mass

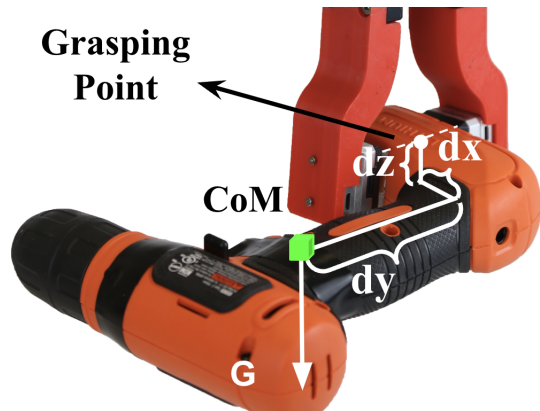


Fig. 2: Illustration of the simplified model of CoM on a real-world object. In our setup, we try to estimate the  $dx$ ,  $dy$ , and  $dz$  which are the displacement of CoM away from the grasping point.

After grasping the object, we define its CoM by some displacement  $dx$ ,  $dy$ , and  $dz$  away from the grasping point. These axes are defined in the world coordinates, as illustrated in Fig. III-A. Ideally, we could directly employ an analytical solution using the 6-dimensional F/T reading from an F/T sensor to accurately determine the CoM. However, real-world complications such as measurement noise and potential in-hand slipping of the object complicate this process. To counteract these issues, a second measurement is necessary. Our method aims to reduce the effect of real-world challenges towards a more robust and accurate prediction.

### B. Bayesian Neural Network for Uncertainty Qualification

The purpose of using BNN is to get a standard deviation for its output value. Instead of training to specify the exact weight of each network node, in the BNN framework, we

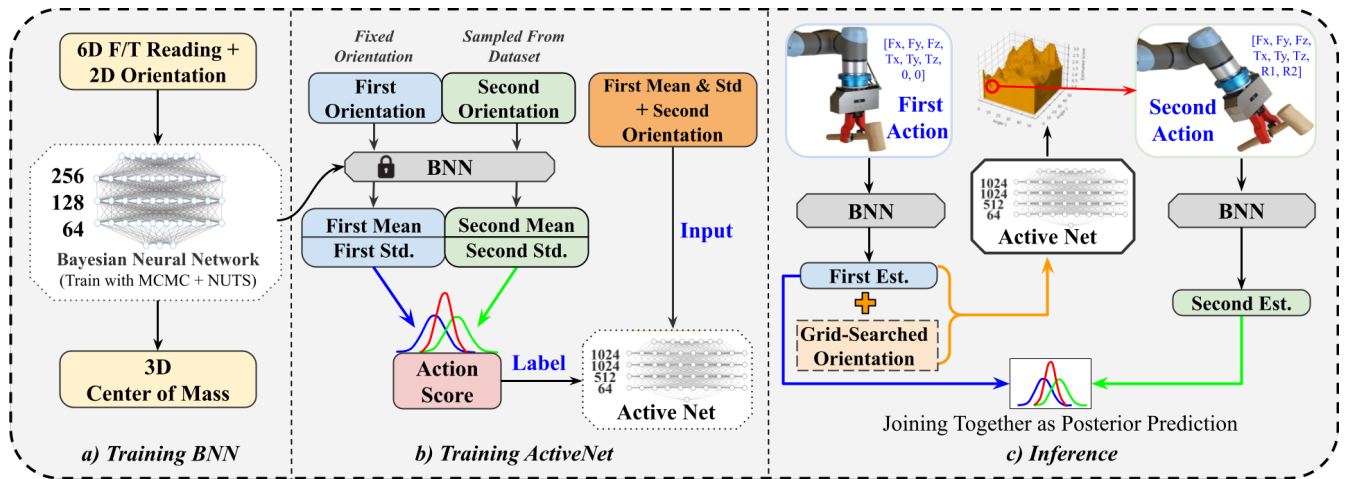


Fig. 3: a) Flowchart for training Bayesian Neural Network. We train BNN with Markov Chain Monte Carlo and No U-Turn Sampler iteratively. b) Flowchart for training an active perception module. We calculate the score from two orientations as the supervised label of the ActiveNet. We use the first prediction’s mean and standard deviation along with the second angle as the input to the network. c) Flowchart for inferencing with U-GRAPH. The robot first grasps with a fixed orientation, then passes the F/T reading with (0, 0) as orientations into the BNN. We use ActiveNet and grid search to find the second action. We pass the second F/T reading with the orientation through BNN to get a secondary prediction and join that with the first prediction to form the posterior prediction.

want to learn a posterior distribution  $p(w|D)$  given the input dataset  $D$ . Each node in our BNN will have a distribution instead of a deterministic value. Given this distribution, we can obtain the estimated distribution of unseen data  $P(\hat{y}|\hat{x})$  by getting the expectation of the predictive distribution:

$$P(\hat{y}|\hat{x}) = \mathbb{E}_{P(w|D)}[P(\hat{y}|\hat{x}, w)]$$

$w$  denotes the posterior distributions of the nodes in the BNN,  $\hat{x}$  denotes the input testing data and  $\hat{y}$  denotes the output prediction.

However, to evaluate this expectation value, we will need an infinite ensemble of networks as mentioned in [24]. To practically approximate this, Monte Carlo sampling methods, particularly Markov Chain Monte Carlo (MCMC), are employed to reduce training and inference costs. MCMC provides unbiased samples from the posterior, facilitating effective posterior inference and backpropagation. Further improving this approach, the Hamiltonian Monte Carlo (HMC) and No U-Turn Sampler (NUTS) are incorporated to avoid the inefficient random walk behavior and dynamically determine the optimal number of steps in the HMC. This automatically adjusts the BNN parameters after each sample to enhance convergence and accuracy [25] [26].

To implement the BNN and MCMC with NUTS, we used Pyro [27] to construct the network, train on our dataset, and evaluate its predictive function. This method gives us reliable uncertainty of the regression prediction of our MLP for active perception. The illustration of the BNN training process can be found in Fig. 3 a).

### C. ActiveNet: Action Selection Network

As mentioned before, our actions have 2 degrees of freedom, the last two joints on the robot are free to move,

while all other joints are fixed during perception. We always keep the orientation [0,0] as the first orientation. This is the orientation where the gripper points straight down, as shown in Fig. 3 c). To find the best second orientation that improves the prediction result, we consequently design ActiveNet and use grid search to find such orientation. The most intuitive way to generate a new action is to directly estimate from the prediction of the BNN and train the network to predict the best subsequent orientation. In our case, there are usually multiple orientations that the robot can take to minimize the error of CoM prediction. A simple regression model explicitly predicts a single “best” action, but it can overlook other “good” actions, especially if these are localized away from the highest peak.

We therefore try to perform a grid search through the action space and estimate a score to determine how good each action is. For simplicity, we define this score as the error of estimation obtained by the BNN after we perform a specific rotation that results in the orientation  $a$ . As a result, the input of our ActiveNet as illustrated in Fig. 3 b) has three parts, the estimation from BNN, the standard deviation from BNN, and a new proposed action that be scored on. The output of the ActiveNet is a score of this new proposed action.

### D. Inference

Our inference pipeline is illustrated in Fig. 3 c). We first use the fixed orientation to generate a prior estimation of the CoM location. Then ActiveNet performs a grid search over the entire action space and calculates the score for each action with prior estimation as input. It uses the action with the minimum action score to proceed. After we obtain the new F/T reading from the second orientation, we then predict



the CoM again with the same BNN. Finally, we treat each orientation as an independently observed measurement of CoM. Since our network can provide a quantified uncertainty, we assume the two measurements are Gaussian. Therefore we can obtain the posterior estimation with:

$$\mu_{final} = \frac{\left(\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2}\right)}{\left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)}$$

#### IV. EXPERIMENT SETUP

In this section, we discuss the hardware setup of the CoM estimation problem. We also explain how to set up the hardware, collect training data, and implement models.

##### A. Hardware Setup

The hardware system features a 6-DoF UR5e robot arm. Attached to the robot’s wrist is a 6-axis NRS-6050-D80 F/T sensor from Nordbo Robotics with a sampling rate of 1000 Hz. The arm is also equipped with a WSG-50 2-fingered gripper from Weiss Robotics with customized PLA 3D-printed fingers. The system is shown in Fig. 4 a).

For data collection, we designed and 3D-printed two objects with dimensions of 15cm × 15cm × 8cm, each including two holders sized 4cm × 4cm for placing AprilTags [28]. The plate object weighs 127.36 grams and allows grasp onto the center. The box object weighs 185.36 grams and is designated to be grasped on the side. We utilize standard laboratory weights for the experiments, specifically two 100-gram weights and one 200-gram weight. Fig. 4 b) shows the printed version of these objects, as well as weights that are randomly placed on them.

##### B. Data Collection

As mentioned in Sec. IV-A, we only collect data from the two customized objects for CoM estimation. Our model is based on the premise that the CoM of any grasped object can be fundamentally described by the offsets  $dx$ ,  $dy$ ,  $dz$ , and the gravitational force  $G$  acting on the object. During each trial, we randomly select 0 to 2 weights to be fixed onto one of the objects. On the software side, our data collection algorithm first uses the overhead camera to detect AprilTag and calculate the CoM while figuring out the graspable zone on the object. It randomly generates a valid grasping point from the graspable zone and calculates the  $dx$ ,  $dy$ , and  $dz$  from the CoM. Finally, the robot moves to the location and grasps the object to start a trial of data collection. This data collection algorithm saves the trouble of intensive human labor and allows us to do a larger scale of data collection.

After securing a grasp, the robot rotates the object to 100 different orientations (excluding the default [0,0]), recording the F/T readings at each position. We use the difference between the F/T reading with the object gripped and the F/T reading with nothing between fingers. Then, we loop through the entire 100 different orientations to calculate the action score for each of them. In total, we spent about 150 hours on data collection to generate a dataset from 204 different grasps, comprising 18,893 F/T readings.

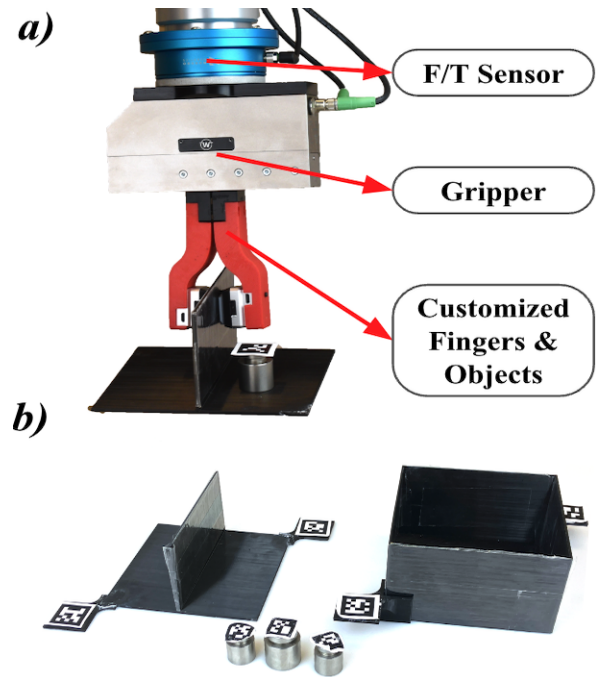


Fig. 4: a) Example of a data collection robot grasping with the location of F/T Sensor and gripper. b) Printed data collection objects in the real world, and standard lab weights for training data collection. AprilTags are placed on all of the objects. We refer to the object on the left as Plate and the object on the right as Box.

##### C. Model Implementation

The BNN has a backbone with a hidden size of [256, 128, 64]. To speed up the training process, we first train with PyTorch for a deterministic MLP with RADam [29] optimizer and a learning rate of 0.001 for 500 epochs. Then we use the pre-trained weights as mean and standard deviation of 0.5 as the initialization for our BNN. We train the BNN using Pyro with 1000 samples and 200 warmup steps.

The ActiveNet is a 5-layer MLP with a hidden size of [1024, 1024, 512, 64]. We train the ActiveNet with RADam optimizer with a learning rate of 0.0001 for 500 epochs.

#### V. RESULTS AND DISCUSSION

In this section, we present the result obtained from the CoM estimation with the U-GRAPH pipeline. We evaluate the performance of our model on the customized data collection setup with unseen weight distributions. Additionally, we experiment on real-world objects with known CoM to validate the effectiveness of our proposed framework.

##### A. Baseline Methods

In contrast to our proposed method, we implement the following 3 different baseline methods:

**Analytical Solution:** The analytical solution assumes a perfect world scenario with no noise in the F/T measurement and no in-hand slip. We can easily obtain the CoM of any object using the following formula:  $\vec{r}_{CoM} = \frac{\vec{\tau} \times \vec{F}}{|\vec{F}|^2}$ , Where  $\vec{F}$  denotes the force reading and  $\vec{\tau}$  denotes the torque readings.



Since we are only using the  $[0, 0]$  orientation, no torque should be caused by the offset of the  $Z$ -axis. However, real-world measurement noise introduces randomness into this calculation. Additionally, the F/T sensor produce inefficiently accurate torque measurement when the sensor is not placed vertically. Therefore, in this paper, we use analytical solutions only in the default pose as the baseline method.

**One Grasp:** The One Grasp method only uses the first part of our proposed pipeline which is the model that takes in one F/T measurement and tries to infer the CoM. Unlike the analytical solution, this method also uses a neural network for estimation. The MLP used is the same one we used for active perception inference.

The previous two baselines have a fundamental flaw that based on one grasp it is impossible to evaluate the offset on the  $Z$ -axis.

**Random Rotate:** The Random Rotate method uses two measurements similar to our proposed method. However, instead of doing an informed action, this method uses a random action selected from the continuous action space to do measurement again. After getting the new reading, we will use the same BNN and joint distribution methods as our proposed method to estimate the CoM.

### B. CoM Estimation on Customized Training Objects

In our first experiment, we use the same plate and box setup but vary the weight configurations. We test five different weight configurations across both objects: no weight, a single 100-gram weight, two 100-gram weights placed together, two 100-gram weights placed separately, and two separate weights that weigh 300 grams in total. For each configuration, we conducted five randomly selected grasps. The data for this experiment is captured using the same overhead camera and AprilTags setup as training. The results of these experiments are detailed in Figure 5, where we present a comprehensive analysis including the mean error and mean standard deviation for each method applied.

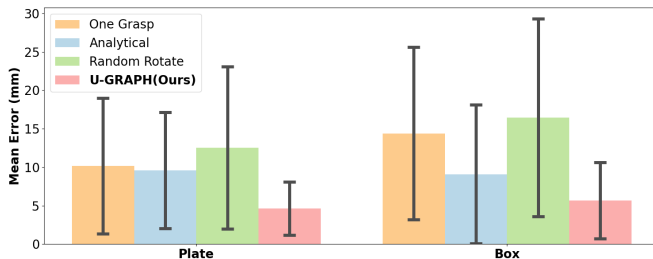


Fig. 5: Mean error and mean standard deviation (shown with the error bar) of the estimated center of mass for customized objects obtained from different methods.

### C. CoM Estimation on Unseen Real-World Objects

We also perform experiments on a set of 12 real-world objects that are commonly seen in daily life. We predefined the grasping point and found the ground truth CoM by balancing the object on each axis with a gripper. The objects have weights ranging from 43.4 grams to 613.2 grams

with maximum dimensions from 56 mm to 285 mm. We attempt to create variations in the  $X$ ,  $Y$ , and  $Z$  axis of the measurement to assess the robustness of the methods. We present the result on the error of each axis for each object in Tab. I along with each object’s dimension and weight. We will give a more comprehensive discussion and analysis in Sec. V-E.

### D. Additional Study on the Effect of Weights of the Object

The performance of our algorithm is observed to decline when the object weight falls outside the range of our initial data collection, as highlighted in Tab. I. We have set up a focused experiment using the Mustard Bottle (Object 11 in Tab. I) as our primary test subject to further investigate. For this experiment, we supplemented the Mustard Bottle with three different sets of weights, bringing the total weights to 244.6 grams, 446.2 grams, and 648.1 grams, respectively. We tape sets of standard laboratory weights on the side of the mustard bottle around the measured CoM location. The first two weights fall within the weight range of our collected dataset, while the last weight surpasses the upper limit of our previous data collection. We maintain a consistent grasping position across all weight variations to isolate the effect of weight on our CoM estimation accuracy. The specific grasping locations and the errors in the CoM predictions made using our method for each weight configuration are documented in Tab. II.

### E. Limitation and Discussion

Our approach aims to minimize the total error in CoM estimation, which occasionally compensates across different axes. As evidenced in Tab. I, while our method might yield slightly poorer results on one axis, it significantly enhances performance on others. Nonetheless, our method demonstrates superior accuracy in 10 out of the 12 test objects when compared to other techniques. This demonstrates the effectiveness of taking a second measurement, as highlighted by our comparison with the baseline method, **One Grasp**, and the importance of informed active perception noted against the **Random Rotate** method, which lacks the informed approach of our second measurements. We also show that with a second action, we can always improve the estimation along the  $z$ -axis. This aligns with the intuition that a new orientation will introduce new information about the offset of the  $z$ -axis.

To further expand this work, predictions could benefit from multiple actions and continuous updates to the estimated center of mass. In this paper, our objective is to demonstrate that this predictive framework can enhance CoM estimation, rather than to claim a complete solution to the problem. Future research should investigate the optimal number of rotations and explore whether a series of actions and predictions can, when combined, converge to the true CoM.

Sec. V-D illustrates how the objects’ weight range, spanning 127.36 to 585.36 grams, affects our algorithm’s performance. However, in practice, heavier objects are often

Objects	Image						
	Dimensions (mm)	285 × 50 × 50	185 × 45 × 40	244 × 42 × 39	51 × 152 × 99	140 × 65 × 25	105 × 105 × 25
	Weight (g)	260.7	325.1	136.9	307.7g	144.8	122.7
Prediction Error (mm)	Axis	X Y Z	X Y Z	X Y Z	X Y Z	X Y Z	X Y Z
	One Grasp Only	11.4 14.2 14.0	17.8 16.1 6.2	12.0 5.9 8.4	16.8 16.9 19.8	8.1 16.7 11.5	2.6 9.1 18.4
	Analytical Solution	6.9 5.5 8.4	6.3 <b>4.7</b> 17.9	14.0 3.1 9.3	6.1 3.8 10.7	9.9 5.5 19.4	2.3 <b>3.2</b> 21.2
	Random Rotate	10.1 22.2 13.7	13.6 18.8 <b>3.3</b>	12.2 4.9 13.4	<b>5.2</b> 14.8 17.1	13.3 11.2 9.2	2.4 9.4 18.5
	<b>U-GRAPH (Ours)</b>	<b>2.7 4.0 5.6</b>	<b>5.0</b> 5.6 4.3	<b>3.9 2.2 6.3</b>	<b>5.2 3.0 9.3</b>	<b>3.3 3.0 5.0</b>	<b>2.2</b> 6.8 <b>7.9</b>
Objects	Image						
	Dimensions (mm)	218 × 3 × 3	156 × 96 × 64	135 × 109 × 94	175 × 51 × 188	92 × 56 × 192	56 × 56 × 56
	Weight (g)	197.4	172.6	400.0	613.2 (OOD)	43.4 (OOD)	76.8 (OOD)
Prediction Error (mm)	Axis	X Y Z	X Y Z	X Y Z	X Y Z	X Y Z	X Y Z
	One Grasp Only	14.4 17.5 9.4	31.0 9.0 16.2	13.7 <b>6.1</b> 21.1	18.4 25.3 26.4	25.7 15.3 <b>8.1</b>	17.2 8.4 8.0
	Analytical Solution	4.6 <b>3.0</b> 13.7	27.9 4.8 9.3	6.8 6.5 17.3	13.5 <b>9.2</b> 17.9	<b>7.3</b> 8.2 11.4	<b>1.8</b> 4.9 7.5
	Random Rotate	15.4 23.2 18.0	26.2 3.8 20.3	11.9 20.9 14.4	<b>11.0</b> 20.1 25.3	25.9 8.5 12.1	15.9 <b>4.3</b> 4.2
	<b>U-GRAPH (Ours)</b>	<b>4.5 7.7 8.7</b>	<b>21.6 3.6 8.5</b>	<b>6.7 6.5 10.9</b>	11.8 15.8 <b>15.6</b>	12.1 <b>7.4</b> 10.1	13.5 9.3 <b>3.2</b>

TABLE I: The table shows the mean error of each axis of all 12 real-world objects. We performed 5 different grasp configurations on each object and tried to maximize the variations of  $dx$ ,  $dy$ , and  $dz$  for each grasp. We also show the results of the baseline methods and bold the best-performing estimation for each axis of each object. The  $X$ ,  $Y$ , and  $Z$  axes are defined by the world frame. The OOD label in the last three objects’ weight means that their weight is out of our collected data distribution.


	Weights (g)	Mean Error (mm)		
		X	Y	Z
	43.4 (OOD)	12.1	7.4	10.1
	244.6	5.0	6.1	6.4
	446.2	6.1	4.4	5.6
648.1 (OOD)	15.6	12.7	13.2	

TABLE II: The left image shows the grasping locations on the mustard bottle. The right table shows the mean prediction error on each axis for different weights. The  $X$ ,  $Y$ , and  $Z$  axes are defined by the world coordinate. The OOD label means the weight is out of our training data distribution.

inherently unstable for grasping, while extremely light objects fail to generate sufficient F/T signals to overcome noise, making it difficult to expand the dataset in these regions. Despite these challenges, our method demonstrates robust performance across diverse real-world objects that differ significantly in contact geometry, surface friction, and density from the training set, confirming its strong generalizability.

Moreover, our system encounters difficulties with large slips, especially with heavier objects or when the CoM is significantly offset from the grasp points. This is a common challenge in achieving stable grasps and accurate CoM estimations. To address these difficulties, we plan to

integrate fingertip GelSight sensors [30] into our system in the future. These sensors will enable precise measurement of slips during manipulation, allowing us to gather critical data to refine our algorithm further. By enhancing our ability to detect and adjust for slips, we aim to improve both the stability of grasps and the robustness of CoM estimations.

## VI. CONCLUSIONS

This paper presents U-GRAPH, a novel approach to the center of mass estimation with active perception. We design a pipeline that contains two main components: a Bayesian Neural Network that can provide prediction and its associated uncertainty, and an ActiveNet that produces an informed rotation based on the prior estimation. This approach reduces the need for repetitive grasping by replacing it with an efficient and effective rotation. Our experiments validate the effectiveness of U-GRAPH, which consistently outperforms traditional methods and adapts well to real-world scenarios.

## ACKNOWLEDGMENT

The authors would like to thank Mohammad Amin Mirzaee for his help with photos and figures, Hung-Jui (Joe) Huang for thoughtful discussions, and Harsh Gupta for revising the paper. We also thank Xiping Sun and Yilong Niu for helping make the video.

## REFERENCES

- [1] Z. Wang, X. Wu, S. Huang, L. Dong, W. Wang, S. Ma, and F. Wei, "Kosmos-e : Learning to follow instruction for robotic grasping," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 9510–9517.
- [2] Y. Mo, H. Zhang, and T. Kong, "Towards open-world interactive disambiguation for robotic grasping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8061–8067.
- [3] S. Kanitkar, H. Jiang, and W. Yuan, "Poseit: A visual-tactile dataset of holding poses for grasp stability analysis," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 71–78.
- [4] Z. Si, Z. Zhu, A. Agarwal, S. Anderson, and W. Yuan, "Grasp stability prediction with sim-to-real transfer from tactile sensing," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 7809–7816.
- [5] D. J. C. MacKay, "A Practical Bayesian Framework for Backpropagation Networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 05 1992.
- [6] H.-J. Huang, X. Guo, and W. Yuan, "Understanding Dynamic Tactile Sensing for Liquid Property Estimation," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.
- [7] X. Guo, H.-J. Huang, and W. Yuan, "Estimating properties of solid particles inside container using touch sensing," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 8985–8992.
- [8] C. Matl, R. Matthew, and R. Bajcsy, "Haptic perception of liquids enclosed in containers," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7142–7149.
- [9] 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)*, 2020, pp. 5351–5357.
- [10] H.-J. Huang, J. Xiang, and W. Yuan, "Kitchen artist: Precise control of liquid dispensing for gourmet plating," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13933–13939.
- [11] X. Luo, S. Jin, H.-J. Huang, and W. Yuan, "An intelligent robotic system for perceptive pancake batter stirring and precise pouring," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 5970–5977.
- [12] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," in *Proceedings of Robotics: Science and Systems (RSS)*, 2019.
- [13] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, "Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5633–5640.
- [14] M. Murooka, S. Nozawa, M. Bando, I. Yanokura, K. Okada, and M. Inaba, "Simultaneous planning and estimation based on physics reasoning in robot manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3137–3144.
- [15] S. M. Hyland, J. Xiao, and C. D. Onal, "Predicting center of mass by iterative pushing for object transportation and manipulation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1615–1620.
- [16] S. McGovern, H. Mao, and J. Xiao, "Learning to estimate centers of mass of arbitrary objects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1848–1853.
- [17] S. McGovern and J. Xiao, "Learning and predicting center of mass through manipulation and torque sensing," in *2022 8th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, 2022, pp. 60–66.
- [18] J. J. Gibson, "Observations on active touch." *Psychological Review*, vol. 69, pp. 477–491, 1962.
- [19] J. Xu, H. Lin, S. Song, and M. Ciocarlie, "Tandem3d: Active tactile exploration for 3d object recognition," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10401–10407.
- [20] P. Uttayopas, X. Cheng, and E. Burdet, "Active haptic exploration based on dual-stage perception for object recognition," in *2023 IEEE World Haptics Conference (WHC)*, 2023, pp. 347–353.
- [21] A. Kruzliak, J. Hartvich, S. P. Patni, L. Rustler, J. K. Behrens, F. J. Abu-Dakka, K. Mikolajczyk, V. Kyrki, and M. Hoffmann, "Interactive learning of physical object properties through robot manipulation and database of object measurements," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 7596–7603.
- [22] W. Yuan, Y. Mo, S. Wang, and E. H. Adelson, "Active clothing material perception using tactile sensing and deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4842–4849.
- [23] J. Ketchum, A. Prabhakar, and T. D. Murphey, "Active exploration for real-time haptic training," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 9689–9695.
- [24] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ser. ICML'15, 2015, p. 1613–1622.
- [25] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo," 2011. [Online]. Available: <https://arxiv.org/abs/1111.4246>
- [26] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, May 2011. [Online]. Available: <http://dx.doi.org/10.1201/b10905>
- [27] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: deep universal probabilistic programming," *J. Mach. Learn. Res.*, vol. 20, no. 1, p. 973–978, jan 2019.
- [28] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.
- [29] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, April 2020.
- [30] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, 2017. [Online]. Available: <https://www.mdpi.com/1424-8220/17/12/2762>