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

Embodied Uncertainty-Aware Object Segmentation

Xiaolin Fang, Leslie Pack Kaelbling, Tomás Lozano-Pérez
MIT CSAIL
{xiaolinf,lpk,tlp}@csail.mit.edu

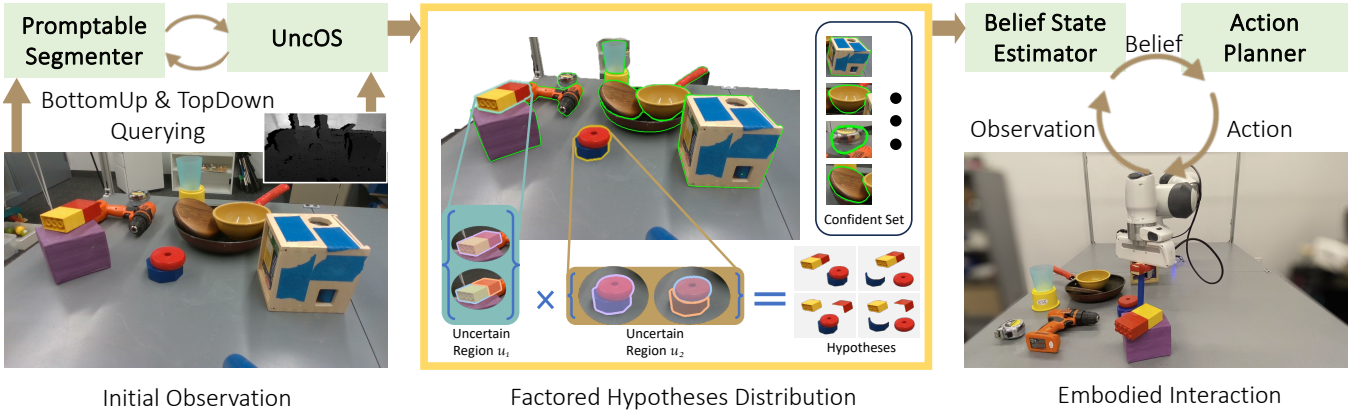

Fig. 1: Embodied segmentation with uncertainty-aware object segmentation model (UNCOS) as a basis. EOS architecture: an initial RGB-D image is repeatedly prompted by UNCOS to obtain a region-based factored segmentation hypotheses distribution. Unambiguous regions are put into the confident set (outlined in green). Alternative hypotheses are proposed for each uncertain region. A distribution over segmentation hypotheses for the whole image is constructed by taking the Cartesian product of hypothesis distributions in each region. Such factored hypothesis distribution is used to initialize a 3D belief state that forms the basis for information-gathering action planning in embodied object segmentation.

***Abstract*— We introduce *uncertainty-aware object instance segmentation* (UNCOS) and demonstrate its usefulness for embodied interactive segmentation. To deal with uncertainty in robot perception, we propose a method for generating a hypothesis distribution of object segmentation. We obtain a set of region-factored segmentation hypotheses together with confidence estimates by making multiple queries of large pre-trained models. This process can produce segmentation results that achieve state-of-the-art performance on unseen object segmentation problems. The output can also serve as input to a belief-driven process for selecting robot actions to perturb the scene to reduce ambiguity. We demonstrate the effectiveness of this method in real-robot experiments. Website: https://sites.google.com/view/embodied-uncertain-seg.**

I. INTRODUCTION

Our goal is to build long-horizon manipulation systems that can operate in environments that contain previously unknown objects. A key step in such systems is segmenting images, either RGB or RGB-D, into candidate objects to be manipulated. This step is often called "unknown object instance segmentation" (UOIS) and a number of existing deep-learning models have been developed for this task [1], [2], [3]. However, the output from these models is inevitably imperfect, due to limitations in the model, for example, limited data or limited capacity, or to challenges in the images, for example, occlusion or lighting, or to fundamental ambiguity, for example, in a stack of toy blocks. In the "embodied" manipulation setting, where we have a robot available, we can interact with the scene so as to obtain additional information, such as pushing some of the objects and tracking how they move. Furthermore, with the advent of "promptable" segmentation models [4], we can also interact with the model to obtain additional information, such as obtaining multiple segmentations from different prompts. In this paper, we explore both of these methodologies for improving segmentation results: multiple prompting of the segmentation model and active robot interaction with the objects. In particular, we construct (by multiple prompting) a characterization of the uncertainty in the segmentation and use that representation to guide the physical interaction.

Image segmentation, in its most general form, is fundamentally underconstrained. Is the bottle cap part of the bottle or a separate object? Is the shirt part of the person? In this paper, we limit ourselves to considering the segmentation of discrete rigid objects, where the answer to these questions is: if chunks of matter always move together rigidly, then they form a single object, and not otherwise. It will typically be impossible to find this ground-truth segmentation from an image of a cluttered scene and, in general, it may not be necessary to find it so as to achieve a particular robotic manipulation goal.

We define our task as that of *uncertainty-aware object instance segmentation*. Given an image, a solution to the problem partitions a scene into disjoint regions and provides a single interpretation for each region of the scene

that has sufficiently low uncertainty and provides multiple interpretations for each region with high uncertainty. This is different from the classic instance segmentation task, where the objective is to deliver a single set of masks for the scene. By explicitly characterizing this region-factored uncertainty, we hope to enable improved performance on downstream tasks, for example, improving the choice of actions to gather additional information to disambiguate the segmentation.

A crucial question in this approach is how to characterize the uncertainty in a proposed segmentation. We develop an uncertainty estimation and hypotheses generation method based on multiple queries to large pre-trained, "promptable" models [4], [5]. Within a region of the image, we issue random point prompts and use consistency of the returned masks as an indication of uncertainty.

Having obtained object hypotheses, with multiple candidate segmentation of uncertain regions, we use the robot to do targeted exploration aimed at reducing the uncertainty. We use a maximal-uncertainty-reduction-driven action selection heuristic to lightly push a candidate object. We build a state estimator to track and update the object hypotheses. The most likely segmentation hypothesis can be computed from the resulting belief state and we show that the state estimation leads to better action choices which ultimately leads to better maximum-likelihood segmentation hypotheses.

The key contributions of this work are:

- UNCOS: An active prompting strategy for combining promptable top-down and bottom-up pre-trained object instance segmentation methods to obtain a distribution over image-segmentation hypotheses;
- EOS: A method for converting this segmentation distribution into a distribution over world models and using that for selecting robot perturbation actions to disambiguate the segmentation.

We demonstrate the effectiveness of the UNCOS image-segmentation strategy first as a stand-alone method, showing that taking its maximum-likelihood hypothesis performs better than state-of-the-art UOIS methods. Furthermore, we show that the hypothesis distribution produced by UNCOS can be used by EOS to generate targeted physical interactions with the scene that gather information much more efficiently than less-informed alternatives.

II. RELATED WORK

This paper is related to previous work on unseen object instance segmentation (UOIS), the use of large pre-trained models in image segmentation, estimating uncertainty in segmentation, and on embodied image segmentation.

Unseen object instance segmentation (UOIS) UOIS for robotics aims to find an instance segmentation of objects in the foreground, typically for a tabletop scene. Recent work leverages datasets generated in simulation using a large set of objects [1], [2], [3], [6]. A difference from common panoptic, semantic, and instance segmentation scenarios is that a depth image is assumed to be available. These methods make predictions based on both intensity cues and geometric cues. Although our goal is ultimately to obtain an object segmentation, the crucial difference is that our method estimates a factored distribution over segmentations, which is then improved by interacting with the scene, before committing to a particular segmentation hypothesis.

Segment Anything Model (SAM) Recent large vision models have shown impressive results for various tasks. SAM [4] is an image segmentation model that has been pre-trained on a large dataset of 11 million images. It can produce segmentation masks through point queries or box queries. Due to the flexible prompt interface and strong performance, it has been used to improve different tasks such as 3D scene segmentation [7], [8] and tracking [9], [10]. It has also been combined with other large pre-trained models such as GroundingDINO [11] to segment objects with text-prompts [12]. In our work we exploit the prompting interface for uncertainty estimation.

Uncertainty Estimation in Segmentation Many approaches to uncertainty estimation in segmentation have produced a heat map of uncertainty over pixels [3], [13], [14]. However, the uncertainty we care about is *object level* uncertainty, rather than pixel-wise uncertainty. Some previous approaches have produced probability distributions over relatively small image patches [15], [16]. The common failure modes in modern UOIS are over- and under-segmentation of objects, therefore representing uncertainty via distributions over grouping of individual segmentation masks is more appropriate for our setting.

Embodied Segmentation Using robot actions to complement and enhance visual perception has a long history in robotics and is variously known as active perception, interactive perception or embodied perception; the survey by Bohg *et al.* [17] reviews this body of work, which includes work on interactive/embodied segmentation.

A common strategy for interactive segmentation has been to take a bottom-up approach, starting from an over-segmentation of the scene, and identifying groupings by consistency in motion [15], [18], [19]. An action is chosen greedily to induce motion. There have been a number of strategies for choosing actions. In some cases, the explicit goal is to "singulate" (isolate) the objects [20]. Pajarinen *et al.* [16], on the other hand, formulate the action selection problem as a POMDP and try to pick actions that maximize long-term reward. Very recent work from Qian *et al.* [21] also seeks to improve segmentation based on a small number of robot interactions. The action is selected heuristically based on the pixel-wise uncertainty map from MSMFormer [3]. It differs from our focus on exploiting a representation of uncertainty obtained from prompting large pre-trained models. Another line of work aims to use robot interaction to gather data for self-supervised training of segmentation models [22], [23], [24]. This objective is in contrast with our objective of disambiguating only the current scene.

III. PROBLEM SETTING

Our ultimate objective is to obtain an accurate interpretation of potentially highly cluttered table-top scenes, in

the form of a set of partial point clouds corresponding to individual objects in the scene. We assume that all objects in the scene are rigid and do not address the problem of revealing completely occluded objects.

Scene segmentation is a fundamentally ambiguous problem: it may be both difficult and unnecessary to obtain a single, exactly correct interpretation. For these reasons, we focus on constructing a distribution over segmentation hypotheses, and updating that distribution over time given new observations in which some objects have moved.

The robot embodiment consists of a camera that can observe the entire scene and capture registered RGB and depth images, and a robot arm that can reach the observed objects and make small perturbations by "poking" the objects. Our goal is to produce good interpretations of the scene with a minimal amount of disturbance to the objects.

The robot is assumed to be able to make precise, local contact with objects in the scene. The pushing action is determined by selecting an initial end-effector position, orientation, and motion distance. After executing each motion, the robot retracts to a position that leaves the scene unoccluded.

Although our objective is to maintain a distributional estimate of the segmentation state, in order to compare most directly with existing segmentation methods, we will evaluate our segmentation results in terms of instance segmentation metrics on 2D image masks [6]. Given a hypothesized segmentation $\{s_1, \ldots, s_{N_s}\}$ where s_i is a set of pixels assigned to object i, and a ground-truth segmentation $\{g_1, \ldots, g_{N_g}\}$, we find an assignment ϕ mapping each hypothesized segment into a ground-truth segment (or none) that maximizes the sum of the individual F-scores, and report an overall object-size normalized (OSN) precision, recall, and F-score,

$$P_n = \frac{\sum_i P_i}{N_s}, \;\; R_n = \frac{\sum_i R_i}{N_g}, \;\; F_n = \frac{\sum_i F_i}{\max(N_s, N_g)}$$

where $F_i = 2P_iR_i/(P_i + R_i)$, $P_i = |s_i \cap \phi(s_i)|/|s_i|$, $R_i = |s_i \cap \phi(s_i)|/|\phi(s_i)|$. The object-size normalized metric differs from the standard P/R/F measures [1] in that they explicitly average the scores over *segments* rather than *pixels*. This ensures that simply getting a few large objects correct does not overwhelm the scores of badly segmented smaller objects, which is important for manipulation problems. We additionally wish to achieve a good segmentation result with as little disturbance to the scene as possible. We do not explicitly measure the amount of motion among the objects, but do measure the improvement in segmentation quality as a function of the number of actions performed.

IV. Embodied Uncertainty-Aware Segmentation

We propose *embodied uncertainty-aware object segmentation* (EOS), as illustrated in Fig. 1. EOS consists of three main components, including an *uncertainty-aware object segmentation model* (UncOS), a belief state estimator, and an action planner, operating in closed-loop interaction with the scene. The initial RGB-D image is processed using UncOS, which builds on a promptable image-segmentation model to construct a segmentation hypothesis set. This segmentation hypothesis set is used to initialize a *belief state*, representing a set of hypotheses about the structure of the 3D scene. Given a belief state, an action is selected and executed, and a new RGB-D observation is captured and used to update the belief. Finally, we generate a set of image masks corresponding to the most likely hypothesis.

A. Uncertainty-aware Object Segmentation Model

Our method, *uncertainty-aware object segmentation model* (UncOS), provides a novel strategy for combining multiple queries to pre-trained 2D RGB image-segmentation methods with some operations on the 3D point-cloud generated from a depth image, to produce a set of possible segmentation hypotheses, together with confidence estimates.

UncOS approaches solving the problem from two aspects:

- A "bottom-up" method, that when queried, can return masks that cover a region of interest in an image. This ensures that every region in the image can be accounted for. It is essential that this method have **high recall** so that multiple queries to this method is likely to return most of the correct instance masks. We refer to this method as BUHighRecSeg. We assume it can take densely issued query points, to form an initial set of high recall masks of the whole image. We refer to this as BUSeed.
- A "top-down" method that returns a set of image masks with **high precision**. These masks are very likely to correspond to correct segments, but they may not contain all the correct segments. We refer to this method as TDHighPrecSeg.

The general strategy could use any method meeting these requirements. In our implementation, we use the *Segment Anything Model* (SAM) [4]. Given an image, it can be queried either with a pixel location or a bounding box.

We use the *pixel-prompted* segmentation as our BUHighRecSeg module and its densely issued version (*automatic mask generation*) as our BUSeed module. Our experiments confirm that these two do indeed have very high recall.

We use GroundedSAM [11], [12], which utilizes *box-prompted* segmentation with a natural language prompt, as our TDHighPrecSeg module. GroundedSAM takes text as input, uses GroundingDINO [11] to generate detection bounding boxes for the text, and then prompts SAM to generate a binary mask for each detection box. We query GroundedSAM with a fixed prompt "`A rigid object .`". Our experiments confirm that this method does indeed have very high precision.

Algorithm 1 UncOS

Require: RGB image I, depth image D, camera params θ, text prompt $\triangleright$, overlap threshold γ

$P :=$ PointCloud(D, θ)
$C, U :=$ PartitionRegions(I, P)
$M :=$ TDHighPrecSeg$(I, \triangleright)$
for $u \in U$ **do**:
    $M_u = \{m \in M \mid m \cap u > \gamma\}$
    $H_u =$ GenerateRegionHypotheses(u, M_u)
return $C, \{H_u \mid u \in U\}$

The overall operation of UNCOS is described in Alg. 1. The driving insight is that segmentation uncertainty is strongly region-based. In some regions of the image, the interpretation is unambiguous and there will be a single reasonable hypothesis. However, for other regions, for example, one containing a stack of objects on the table, it is likely that the queried model will return a variety of under- and over-segmentations. However, such ambiguity is usually restricted to the local region and generally does not interact with the interpretation of a different pile of objects.

This insight leads us to *factor* the segmentation distribution by partitioning the image into regions and generating a distribution over hypotheses for each region. A distribution over segmentation hypotheses for the whole image can then be constructed by taking the Cartesian product of hypothesis distributions in each region (Fig. 1). If the scene is constructed in a way that no such locality can be leveraged, UNCOS will simply treat the whole scene as one region.

The algorithm begins by using bottom-up methods to partition the image into non-overlapping regions C, U. The elements of set C of regions are confidently considered to contain a single object. The elements of set U are regions in which the segmentation is deemed to be uncertain. Alg. 2 describes this process in detail. The initial call to BUSEED generates a large number of overlapping regions. We filter out table and background using depth information, by doing plane-estimation with RANSAC. We then construct a graph with the remaining regions as nodes, with an edge between any pair of regions with a substantial overlap. Regions with a single hypothesis are *verified* by calling BUHIGHRECSEG seeded at multiple randomly chosen points within the region: if this process generates substantially different segmentation results then the region will not be included in the *confident set* C. All remaining regions are returned in the *uncertain set* U.

After partitioning the image into disjoint confident and uncertain regions, we start to construct segmentation hypotheses for each uncertain region. To aid in interpreting the uncertain regions, we query TDHIGHPRECSEG to generate a set of candidate object masks for the whole image. We take those masks $v_1, \ldots, v_k$ that overlap with the uncertain region u, to be the seed masks for constructing candidate hypotheses. Our goal in this process is to generate a set of possible partitions H_u of the region u, seeded by these candidates. Alg. 3 illustrates this process: starting with each seed mask v_i (and then continuing beyond that number without seeding mask if we require more hypotheses), we subtract the seed mask out of the whole region u, and then randomly select a point in the remaining area to query BUHIGHRECSEG. If BUHIGHRECSEG returns a new mask that has a high intersection-over-union (IoU) with the unaccounted-for area, r, we accept it into hypothesis $h = \{v_i, ...\}$, remove its area from r, and continue until we have a set h of masks that nearly constitutes a partition of our target region u. Importantly, we also use the point cloud P to determine whether the 3D volume corresponding to a suggested mask is degenerate. Suggested masks that are flat (such as labels) will be rejected.

Once we have generated N_h complete segmentation hypotheses for this region, we check for near duplicates. Two hypotheses h_i and h_j are considered to be duplicates if 1) h_i and h_j have the same number of segments, and 2) the best matching between segments in h_i and those in h_j has an average IoU greater than a threshold. Using this test, we find equivalence classes of hypotheses, and return a single representative of each class; in addition, we compute and return a "bootstrap" confidence measure for each hypothesis class, equal to the number of elements in its class divided by the total number of samples. We use this score to determine the most likely image-segmentation hypothesis when no physical interaction evidence is available.

Finally, in Alg. 1, we return the confident regions C, and the factored hypothesis space, with a distribution over segmentation hypotheses for each uncertain region U.

Algorithm 2 PartitionRegions

Require: RGB image I, point cloud P, Intersection-over-min (IoM) threshold σ_m, IoU threshold σ_u, num verify tests n

$M :=$ BUSEED(I)
$M :=$ RemoveBackgroundRegions(M, P)
$E := \{(i,j) : |m_i \cap m_j| / \min(|m_i|, |m_j|) > \sigma_m\}$
$C :=$ DisconnectedNodes(M, E)
$U :=$ ConnectedComponents$(M - C, E)$
for $c \in C$ **do**
  for $i \in \{1, \ldots, n\}$ **do**
    $m_i =$ BUHighRecSeg$(I,$ RandomPoint$(c))$
    if $|m_i \cap c| / |m_i \cup c| < \sigma_u$ **then**
      $C := C - \{c\}$
      $U := U \cup \{\{c\}\}$
      break
return C, U

Algorithm 3 GenerateRegionHypotheses

Require: Uncertain region u, seed masks $v_1, \ldots, v_k$, point cloud P, number of hypotheses to produce N_h, thresholds α, β

$H := \{\ \}$
for $i \in \{1, \ldots, N_h\}$ **do**
  $r :=$ Copy(u)
  if $i \leq k$ **then**
    $h := \{v_i\}$; $r := r - v_i$
  else
    $h := \{\ \}$
  while $|r| > \alpha$ **do**
    $m :=$ BUHighRecSeg$(I,$ RandomPoint$(r))$
    if $|m \cap r| / |m \cup r| > \beta$ **and not** IsDegenerate(m, P) **then**
      $h := h \cup \{m\}$; $r := r - m$
  $H := H \cup \{h\}$
$EC :=$ EquivalenceClasses(H)
$H^* := \{(ec[0], |ec|/N_h) \mid ec \in EC\}$
return H^*

B. 3D Belief representation

Our embodied segmentation process starts with a belief state initialized with the results of UNCOS. This belief could be integrated into a general goal-directed manipulation planning process, which decides whether or not to invoke information-gathering actions, depending on its given task. The planner can select actions based on the residual uncertainty, picking actions that leads to plan success under any hypothesis (e.g., deciding to push something that might be a stack of objects from the bottom, rather than picking it up from the top, for the task of cleaning up the table).

For the purposes of testing the uncertain segmentation and belief-update process, we embed it in a loop in which the robot takes actions with the goal of reducing uncertainty in the segmentation. It selects an action based on the hypotheses in the initial belief, executes the action on the robot, and obtains a new RGB-D image of the scene after the interaction. We update the belief to both track the motion of the hypothesized objects and to get a new confidence score for each hypothesis. The process repeats for several steps. At any point in this process, we can retrieve the hypothesis with the highest confidence for evaluation against other strategies.

Our 3D belief representation $B = (C^+, U^+)$ retains the factored structure of the 2D segmentation output, but is lifted to 3D and aggregated over time. The set C^+ now consists of a set of 3D objects $c_1^+, \ldots, c_n^+$, represented as point clouds in a global frame. Each region $u_{(r)}^+ \in U^+$ consists of a set of region hypotheses: $u_{(r)}^+ = (h_{(r)1}^+, \ldots, h_{(r)n_r}^+)$, each of which is an interpretation of the region. For simplicity of notation, we will drop the (r) from now on. It should be ranging from one to $|U^+|$. Each region hypothesis h_j^+ consists of a set of 3D objects $(o_{j1}, \ldots, o_{jn_j})$. Each object o_{jk} consists of a point cloud η_{jk} and a confidence score s_{jk} indicating the likelihood that o_{jk} is either a single object or part of a large whole, that is not under-segmented.

As we get additional observations, we will adjust the confidence values s_{jk}. We define a score for each region hypothesis h_j^+ as

$$S(h_j^+) = \frac{1}{|h_j^+|} \sum_k s_{jk} - \lambda \left[|h_j^+| - \min_m |h_m^+| \right] \tag{1}$$

which combines the average "wholeness" confidence of the objects in the hypothesis with a penalty for having extra objects, thus preferring the simplest hypothesis that holds the rigidity assumption.

Since the structure of the 3D belief is the same as the 2D output of UNCOS, we construct the initial belief by simply using the 2D masks to extract segments from the original point cloud P. We initialize all s_{jk} to some fixed initial value p_0. Since the hypotheses in each region are independent of those in other ones, we take the most likely hypothesis for the whole scene to be the union of C^+ and the most likely hypothesis from each uncertain region.

C. Action selection

To demonstrate the utility of the belief representation, we use a robot to selectively poke objects in the scene using a simple greedy strategy that attempts to select a small perturbation that will maximize information gain. We take advantage of the factored uncertainty representation to select a region of the scene that has the highest degree of uncertainty and then select the action that, when applied to that region, induces an observation distribution that is maximally discriminating among its hypotheses.

We measure the uncertainty of a region in terms of the number of high-scoring hypotheses it has:

$$\kappa(u^+) = |\{h_j^+ \mid S(h_j^+ > \delta)\}| \tag{2}$$

After selecting the targeted region, we need to select an informative action. For example, if the two hypotheses for a region are about whether two horizontally aligned parts are rigidly attached, then pushing along the line connecting the part centers won't be as helpful as pushing perpendicular to that. We use the physical simulation result of motions with reconstructed world hypotheses as a heuristic for the potential information gain.

To evaluate the informativeness of an action, we construct simulated world models corresponding to all high-likelihood complete hypotheses. The worlds are constructed by taking the Cartesian product of the likely hypothesis sets for each region: $W = C^+ \times \bigotimes_{(r)} u_{(r)}^+$.

Each world $w \in W$ consists of a set of objects defined by partial point clouds. In order to carry out a simulation, we need to generate completions of these objects, represented as meshes. We follow the same object reconstruction pipeline as Curtis *et al.* [25]: we complete the partial point cloud using a shape completion network and vertical projection, filter out any inconsistency with the current depth image, and reconstruct a concave mesh.

Next, we sample k actions, a, as follows: within the selected target region, we randomly sample an object among all hypotheses for the target region. We then randomly sample a pushing direction across the centroid of that hypothesized object. Next, we simulate the effect of each action in each world, obtaining new depth images $D_{w,a}$. We select the action that induces most differences between the hypotheses:

$$a^* = \operatorname*{argmax}_a \frac{1}{|W|} \sum_w |D_{w,a} - \overline{D_{*,a}}| \tag{3}$$

where $\overline{D_{*,a}}$ is the averaged depth for all w under a. Given a^*, we do motion planning and execute in the real world.

D. Belief update

After executing an action, we update the belief based on the robot's observation. We cannot take advantage of dense observations during the action execution because the object is typically occluded by the robot arm. Instead, we capture a new RGB-D image after the execution has terminated.

To track each hypothesis mask, we use XMem [26] as a multi-object tracker for two neighboring frames. Specifically, at each time step t, for each hypothesized object o_{jk}, we initialize XMem with I^{t-1} and the 2D mask of o_{jk} at $t-1$. We query XMem with the new image I^t and get the updated mask. Compared to optical-flow-based methods

such as RAFT [27], XMem is more robust to occlusion and can handle larger movements.

With the tracked mask, we update the object point cloud and confidence based on our rigidity assumption. We register the point cloud η_{jk}^{t-1} to that of tracked masked area $\eta_{jk}^{t\downarrow}$ using RANSAC. This gives us a rigid transformation T_{jk}^{t}. We use the percentage of inlier points in registered point cloud as a measure of how well the point cloud motion follows the rigid assumption. This is our current time step score s_{jk}^{t}. We assume that the point clouds are sufficiently well registered so that we can just take their union as an update: $\eta_{jk}^{t} \leftarrow (T_{jk}^{t} \cdot \eta_{jk}^{t-1}) \cup \eta_{jk}^{t\downarrow}$. The final confidence score s_{jk} is the weighted average of $\{s_{jk}^{l}\}_{l=1,\dots,t}$ where the weights are determined by the displacement from T^{l} at each step.

V. Evaluation

We are interested in answering two main questions:

- Does performing uncertainty-aware object segmentation model on a single input RGB-D image and generating its most likely hypothesis as output result in image segmentation results that are comparable to other SOTA methods?
- Does the belief state initialized via uncertainty-aware object segmentation model and then updated via embodied uncertainty-aware object segmentation provide a good basis for selecting actions for interacting with the world?

We address these two questions in the following sections.

A. Segmentation from single images

We compare UncOS with several methods. The first two are state-of-the-art UOIS methods that predict a single set of object segmentation masks directly from an RGB-D image: (1) UOIS-Net-3D [1] (2) UCN [2]. The next group of methods use SAM in some way, but do not carry out the repeated queries as in UncOS.

3) SAM: returns output of the *automatic mask generation* query to SAM without further processing.
4) SAM-cluster: based on the observation that SAM tends to over-segment objects, we construct the connectivity graph as described in Alg. 2, and treat every connected cluster as a segmented object.
5) SAM-per-pixel-ML: assigns the highest SAM-*conf.* mask to the pixel if multiple masks contain it [7]. SAM-*conf.* refers to the predicted confidence from the scoring head of SAM that it outputs with every predicted mask.
6) GroundedSAM: GroundedSAM queried with a fixed prompt "`a rigid object`".

We consider our method, UncOS, and several ablations:

7) UncOS – BootstrapScore: returns the hypothesis from UncOS that has the highest average SAM-*conf.* value, instead of the bootstrap confidence score.
8) UncOS – TD: uses UncOS without the TDHighPrecSeg masks from GroundedSAM.
9) UncOS – TD – D: an ablation that further removes the depth filter for degenerate regions.
10) UncOS + UCN: add masks from UCN [2] as additional TDHighPrecSeg masks.

Method	Unc-Aware	P_n	R_n	$F_n \uparrow$
UOIS-Net-3D [1]	✗	86.3	89.1	83.6
UCN [2]	✗	86.7	90.3	84.1
SAM	✗	29.0	<u>91.2</u>	28.4
SAM-cluster	✗	86.3	82.1	78.7
SAM-per-pixel-ML	✗	80.3	86.4	76.1
GroundedSAM	✗	<u>92.7</u>	73.2	72.6
UncOS – BootstrapScore	✓	87.5	88.0	83.2
UncOS – TD	✓	88.5	88.3	84.4
UncOS – TD – D	✓	85.6	88.5	81.8
UncOS + UCN	✓	86.7	90.1	84.3
UncOS (Ours)	✓	89.2	88.9	**85.3**
Oracle UncOS – TD	✓	90.6	89.8	87.1
Oracle UncOS	✓	91.6	90.5	88.4

TABLE I: Comparison of all methods on object-size-normalized (OSN) precision, recall, and F-score. Unc-Aware indicates whether the method is uncertainty-aware. UncOS produces segmentations with highest F_n.

11) UncOS: our method as described in Alg. 1, returns the most likely hypothesis based on the bootstrap confidence score.

These last two methods are included to illustrate the quality of the oracle best hypothesis among all hypotheses generated by UncOS, rather than the one UncOS estimated to be best. It gives an indication of the potential performance improvements we can achieve through physical interaction.

12) Oracle UncOS – TD: The oracle best hypothesis from UncOS without TDHighPrecSeg masks.
13) Oracle UncOS: The oracle best hypothesis from UncOS.

Benchmark We compare the performance of these methods on the OCID dataset [28], which is a standard benchmark for unseen object instance segmentation. It consists of 2390 images of tabletop scenes. Each scene contains an average of 7.5, and up to 20 objects. We report object-size normalized scores $P_n/R_n/F_n$. We quote the results for UOIS-Net-3D [6] and run methods 3 to 13 on the whole dataset. We rerun UCN [2] using the released model from the author and compute the object-size-normalized scores.

Results The results are shown in Table I. Focusing on object-size-normalized F-score (F_n), we observe that UncOS has the highest performance of the non-oracle methods, outperforming (statistically significantly) the state-of-the-art UOIS-Net-3D and UCN methods. Methods based directly on SAM, without reprompting, generally perform worse. It confirms that our iterative uncertainty-aware query process helps to distill better segmentations, from the same underlying model. Comparison between UncOS and – BootstrapScore shows the advantage of using the bootstrap confidence measure in UncOS to select the best hypothesis. Removing the masks from GroundedSAM degrades the performance, which confirms the advantage of having both BUHighRecSeg and TDHighPrecSeg methods. Removing degenerate (flat) regions based on depth helps significantly, showing the advantage of leveraging point-cloud information in our robotics domain. Results from GroundedSAM have the highest precision among all methods, while those from SAM have the

	Method	0	1	2	3	ΔM	ΔSE
F	FINALFRAME	90.9	91.9	89.9	90.8	-0.1	0.8
	RANDOM	87.5	89.7	90.5	90.2	2.6	1.5
	EOS (Ours)	87.1	89.0	92.8	92.9	**5.7**	1.7
F_n	FINALFRAME	79.5	80.6	78.9	79.1	-0.4	1.7
	RANDOM	78.5	81.8	82.0	82.0	3.5	2.4
	EOS (Ours)	78.2	82.7	86.5	86.5	**8.3**	2.4

TABLE II: Real world segmentation results: segmentation quality initially and after each action step; final columns report the mean (M) and standard error (SE) of the changes (Δ) in segmentation quality from step 0 to 3.

highest recall (underlined). These results confirm their suitability for use as TDHIGHPRECSEG and BUHIGHRECSEG methods. Additionally, we find that adding masks from UCN to the hypothesis generation process reduces performance slightly, probably because masks from UCN have lower precision than those from GROUNDEDSAM.

There is a gap between the score of the actually best hypothesis and what UNCOS believes is the best. The gap between these values and those of UNCOS illustrates that there are, in at least some cases, good hypotheses that have not yet been recognized as correct, due to image ambiguity.

B. Improving segmentation through interaction

Once UNCOS has produced a distribution over possible segmentations, we use it to select physical interactions with the scene in order to reduce any remaining uncertainty. We evaluate our embodied uncertainty-aware object segmentation (EOS) system in the real world with a Franka Emika robot arm. To push the object precisely, the Franka grips a stick, as shown in Fig. 1. We use a bidirectional RRT for motion planning and check collisions between the arm and objects using the observed point cloud. The RGB and depth images are captured by a RealSense D435i camera mounted on the gripper. The two questions we want to answer through real-world experiments are: 1) Does UNCOS improve the efficiency of embodied segmentation; 2) Does building local memory and doing belief updates help with image segmentation.

Our primary method, EOS, uses the action-selection method from Sec. IV-C based on a belief initialized from the UNCOS results, and updates using the methods from Sec. IV-D. For evaluation, at each time step, we compare the highest scoring hypothesis from the 3D belief state to human-labeled ground-truth masks. We compare EOS with two ablations:

- RANDOM: We retain the belief state initialization and update methods from EOS, but instead of selecting actions to disambiguate the most uncertain region, we randomly select a hypothesized object to interact with and randomly select a pushing direction. Differences in performance between this method and EOS can be attributed to the use of the uncertainty in the belief representation to focus action selection.
- FINALFRAME: We use random actions, as above, but rather than maintaining a belief state and updating it after each action, we simply take the single image of the object configuration after per interaction step, apply UNCOS to it, and return the most likely hypothesis in UNCOS result. Differences in performance between this method and RANDOM can be attributed to the aggregation of observation information over time in the belief-update mechanism. If this method reveals improved segmentation quality from the first to last frames, it can be attributed to the random motions causing physical separation between the objects, thus makes the segmentation problem easier.

We set up 20 scenes with a collection of 74 diverse objects, shown in Fig. 2. We ran both the EOS and RANDOM methods on each scene (the FINALFRAME method uses the same images as RANDOM, but a different method for generating a predicted segmentation). Although the replication of the scenes for the two runs was not perfect, we set them up as similarly as possible (comparing initial images as we did so). The robot carries out 3 actions in each scene.

Fig. 2: Objects used for real-world evaluation.

Results The average pixel-wise F-score (F) and object-size-normalized F-score (F_n), after K steps of robot interaction are listed in Table II. Both our action selection strategy and the random strategy perform consistently better than the FINALFRAME baseline. With the number of interaction steps increasing, the methods with memory get an increasing segmentation quality, and are higher than that of FINALFRAME. It shows that the embodied segmentation procedure with belief update can help the robot to figure out the ambiguity in the scene and improve the segmentation quality. We include the qualitative results of EOS in Fig. 3.

Comparing our method to the random poking baseline, there is a larger increase in segmentation quality (for both metrics) with the same number of interaction steps. This shows the benefit of having UNCOS and belief update, which provide strong guidance for action selection in embodied segmentation. It is also interesting to note that the FINALFRAME method does not improve as a result of moving the objects, which means that the belief tracking is playing an important role in the performance of the overall system, and it is not just improving due to the physical singulation between objects. For more results, please visit https://sites.google.com/view/embodied-uncertain-seg.

VI. DISCUSSION

Limitations and Future Work. First, our method does not utilize multi-view images to reduce the uncertainty. We are looking to incorporate active perception strategies to reduce

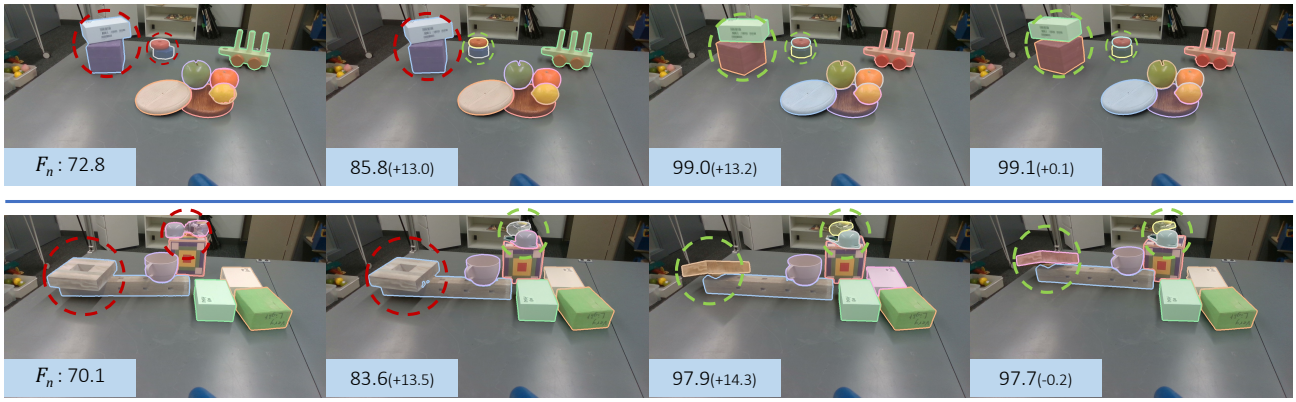

Fig. 3: Qualitative results from embodied segmentation. From left to right are the most likely segmentation results after 0 to 3 steps of interaction using EOS. Incorrect and corrected segmentations are highlighted using red and green dashed circles. F_n and change in F_n to the previous frame are shown in the corner.

the uncertainty caused by occlusion. Second, the current setup seeks to reduce ambiguity in the whole scene. We plan to explore a task-specific information-gathering strategy where only task-relevant regions are explored.

Conclusion. We formulate an uncertainty-aware object instance segmentation problem as the basis for embodied segmentation. Our method UNCOS produces a distribution over possible segmentation hypotheses. The most likely hypothesis from UNCOS has achieved state-of-the-art performance on the UOIS task. Through real-world experiments, we have demonstrated that UNCOS can guide the embodied interaction for efficient targeted disambiguation.

Acknowledgement We thank Emily Chen for helpful discussions on SAM. We gratefully acknowledge support from NSF grant 2214177; from AFOSR grant FA9550-22-1-0249; from ONR MURI grant N00014-22-1-2740; from ARO grant W911NF-23-1-0034; from the MIT Quest for Intelligence; and from the Boston Dynamics Artificial Intelligence Institute.

REFERENCES

[1] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "Unseen object instance segmentation for robotic environments," *IEEE T-RO*, 2021.
[2] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, "Learning rgb-d feature embeddings for unseen object instance segmentation," in *CoRL*, 2020.
[3] Y. Lu, Y. Chen, N. Ruozzi, and Y. Xiang, "Mean shift mask transformer for unseen object instance segmentation," *ICRA*, 2022.
[4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.
[5] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," *ICLR*, 2023.
[6] C. Xie, A. Mousavian, Y. Xiang, and D. Fox, "Rice: Refining instance masks in cluttered environments with graph neural networks," in *CoRL*, 2021.
[7] Y. Yang, X. Wu, T. He, H. Zhao, and X. Liu, "SAM3D: Segment anything in 3d scenes," *ICCV Workshop*, 2023.
[8] J. Cen, Y. Wu, K. Wang, X. Li, J. Yang, Y. Pei, L. Kong, Z. Liu, and Q. Chen, "SAD: Segment any RGBD," *arXiv preprint arXiv:2305.14207*, 2023.
[9] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang, "Segment and track anything," *arXiv preprint arXiv:2305.06558*, 2023.
[10] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng, "Track anything: Segment anything meets videos," in *arXiv preprint arXiv:2304.11968*, 2023.
[11] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.
[12] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded SAM: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.
[13] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *NeurIPS*, 2017.
[14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE T-PAMI*, 2017.
[15] H. V. Hoof, O. Kroemer, and J. Peters, "Probabilistic segmentation and targeted exploration of objects in cluttered environments," *IEEE T-RO*, 2014.
[16] J. Pajarinen, J. Lundell, and V. Kyrki, "POMDP manipulation planning under object composition uncertainty," *IEEE T-RO*, 2023.
[17] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE T-RO*, 2017.
[18] N. Bergström, C. H. Ek, M. Björkman, and D. Kragic, "Scene understanding through autonomous interactive perception," in *ICCV*, 2011.
[19] L. K. Le Goff, G. Mukhtar, P.-H. L. Fur, and S. Doncieux, "Segmenting objects through an autonomous agnostic exploration conducted by a robot," in *IEEE IRC*, 2017.
[20] L. Chang, J. R. Smith, and D. Fox, "Interactive singulation of objects from a pile," in *ICRA*, 2012.
[21] H. H. Qian, Y. Lu, K. Ren, G. Wang, N. Khargonkar, Y. Xiang, and K. Hang, "RISeg: Robot interactive object segmentation via body frame-invariant features," *ICRA*, 2024.
[22] Y. Lu, N. Khargonkar, Z. Xu, C. Averill, K. Palanisamy, K. Hang, Y. Guo, N. Ruozzi, and Y. Xiang, "Self-supervised unseen object instance segmentation via long-term robot interaction," *RSS*, 2023.
[23] H. Yu and C. Choi, "Self-supervised interactive object segmentation through a singulation-and-grasping approach," *ECCV*, 2022.
[24] D. Pathak, Y. Shentu, D. Chen, P. Agrawal, T. Darrell, S. Levine, and J. Malik, "Learning instance segmentation by interaction," in *CVPR Workshop*, 2018.
[25] A. Curtis, X. Fang, L. P. Kaelbling, T. Lozano-Pérez, and C. R. Garrett, "Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances," in *ICRA*, 2022.
[26] H. K. Cheng and A. G. Schwing, "XMem: Long-term video object segmentation with an Atkinson-Shiffrin memory model," *ECCV*, 2022.
[27] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," *ECCV*, 2020.
[28] M. Suchi, T. Patten, D. Fischinger, and M. Vincze, "Easylabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets," in *ICRA*, 2019.