DOI:
10.1109/TMECH.2022.3201057

Document Version
Peer reviewed version

Where Shall I Touch? Vision-Guided Tactile Poking for Transparent Object Grasping

Jiaqi Jiang[1,2], Guanqun Cao[1], Aaron Butterworth[1], Thanh-Toan Do[3] and Shan Luo[1,2]

***Abstract*—Picking up transparent objects is still a challenging task for robots. The visual properties of transparent objects such as reflection and refraction make the current grasping methods that rely on camera sensing fail to detect and localise them. However, humans can handle the transparent object well by first observing its coarse profile and then poking an area of interest to get a fine profile for grasping. Inspired by this, we propose a novel framework of vision-guided tactile poking for transparent objects grasping. In the proposed framework, a segmentation network is first used to predict the horizontal upper regions named as *poking regions*, where the robot can poke the object to obtain a good tactile reading while leading to minimal disturbance to the object's state. A poke is then performed with a high-resolution GelSight tactile sensor. Given the local profiles improved with the tactile reading, a heuristic grasp is planned for grasping the transparent object. To mitigate the limitations of real-world data collection and labelling for transparent objects, a large-scale realistic synthetic dataset was constructed. Extensive experiments demonstrate that our proposed segmentation network can predict the potential poking region with a high mean Average Precision (mAP) of 0.360, and the vision-guided tactile poking can enhance the grasping success rate significantly from 38.9% to 85.2%. Thanks to its simplicity, our proposed approach could also be adopted by other force or tactile sensors and could be used for grasping of other challenging objects. All the materials used in this paper are available at https://sites.google.com/view/tactilepoking.**

***Index Terms*—Transparent objects, tactile sensing, visual perception, multi-modal sensing, object segmentation, robot grasping and manipulation.**

I. INTRODUCTION

TRANSPARENT objects are widely used in our daily life, e.g., glass cups, plastic bottles and glass pan lids in a kitchen. They are also common in research laboratories [1], [2], e.g., vials, glass flasks and Petri dishes. Many of these objects are fragile and easy to break, therefore need to be handled with extra attention. To have robots work in such environments, it is essential for robots to have safe interaction with transparent objects. Without this capability, transparent objects may be broken or have their contents spilled. Broken glass or spilled liquid will pose hazards to the robot and people that share its space.

Fig. 1: An illustration of vision-guided tactile poking for transparent object grasping as we do in our daily lives. As shown in the figure on the left, a glass cup on the table is hard to detect due to its transparency, which brings difficulties to grasping the cup. Before we grasp the cup, we first have a glance at the cup and predict potential areas for contact. We then move our hands following the vision guidance to poke the cup, as shown on the left. The contact with the cup will give us an accurate localisation of the cup, which facilitates a stable grasp of the cup, as shown on the right.

However, it is still challenging for a robot to detect and grasp transparent objects [3], [4]. Most objects in previous object detection and grasping research have been opaque and the perception of transparent objects remains a challenging problem. Compared to opaque objects, transparent objects lack salient features in their surfaces such as colour and texture features. Moreover, their transparent materials violate the Lambertian assumption that optical 3D sensors (e.g., LiDAR and RGB-D cameras) are based on: the opaque objects reflect light evenly in all directions, resulting in a uniform surface brightness from all viewing angles, however, the surfaces of transparent objects both reflect and refract light. Hence, most of the depth data of transparent objects from depth sensors is invalid or contains unpredictable noise. Due to these challenges, most of the current grasping methods that rely on accurate depth information from cameras cannot be directly applied to the grasping of transparent objects.

Humans grasp objects with rich sensory information [5], [6], such as the visual information obtained from eyes and the tactile feeling via physical interaction. It is common that vision with a wide field of view is used first for fast localisation of objects, then touch providing accurate perception of compliance and contact force is used to align hand posture or grip strength to enable a stable grasp [7]. Research on the coordination of human visual and tactual input [8], [9] has shown that we use vision to anticipate an object's physical characteristics prior to contact, preparing the hand for grasping, whereas touch takes over control after the object is in the hand, in particular while interacting with transparent objects [10] as shown in Fig. 1.

Manuscript received: 29th December, 2021; revised: 9th May, 2022 and 12th July 2022; accepted 30th July, 2022.

This work was funded in part by the EPSRC ViTac project (EP/T033517/1), and in part by the University of Liverpool and China Scholarship Council Award.

[1]J. Jiang, G. Cao, A. Butterworth and S. Luo are with the smART-Lab, Department of Computer Science, University of Liverpool, Liverpool L69 3BX, United Kingdom. E-mails: {jiaqi.jiang, g.cao, a.butterworth, shan.luo}@liverpool.ac.uk.

[2]J. Jiang and S. Luo are also with Department of Engineering, King's College London, London WC2R 2LS, United Kingdom. E-mail: {jiaqi.1.jiang, shan.luo}@kcl.ac.uk.

[3]T.-T. Do is with Department of Data Science and AI, Monash University, Clayton, VIC 3800, Australia. E-mail: toan.do@monash.edu.

Inspired by those observations, we propose a novel vision-guided tactile poking approach for grasping transparent objects in this paper. Different from most prior studies [3], [4] using only RGB-D images to address these challenges of transparent objects, our method integrates visual and tactile sensing so that they aid each other and improve the grasping performance. We first train a deep neural network named *PokePreNet* with synthetic RGB images to predict the poking regions that are with similar surface normals to the table surface. The contacts with those areas contribute to good tactile readings while leading to minimal disturbance to the state of the object. A robotic arm equipped with a tactile sensor is then guided to contact those regions, so as to generate informative local profiles of the contacted transparent objects. Finally, using the improved profiles, a heuristic grasp proposal is generated for grasping the transparent object.

To evaluate the performance of our PokePreNet, we construct a high-quality synthetic dataset as well as a real-world test benchmark with over 9,000 RGB images and their corresponding ground truth annotations. To bridge the gap between the simulation and the real world, we randomise the simulator to expose the model to a wide range of environments while training. Our experiments demonstrate that our proposed method can learn vision-guided tactile poking regions, with a high mean Average Precision (mAP) of 0.360, and generalise to transparent objects in the real world. We also conduct real robot experiments and results show that our proposed method can enhance the success rate of transparent object grasping from 38.9% to 85.2%, compared to a vision based grasping. Thanks to its simplicity, the proposed method can be adapted to other settings that use other force or tactile sensors, and can also be used for grasping of other challenging objects.

Our contributions can be summarised as follows:

- We propose a vision-guided tactile poking approach for grasping transparent objects, which is the first of its kind;
- We introduce a novel poking region segmentation network trained with a pixel-level Positive-Negative-balanced loss, which boosts segmentation performance;
- We have collected a high-quality synthetic dataset for transparent object perception and grasping to bridge the reality gap, which is the largest of its kind.

The rest of this paper is structured as follows. Section II reviews the related works and Section III introduces the robot setup and the dataset; Section IV details our proposed vision-guided tactile poking method for transparent object grasping; Section V analyses the experimental results; Finally, Section VI summarises the paper and discusses the work.

II. Related Works

A. Transparent Object Grasping

There are two types of methods for robots to perform transparent object grasping in the literature. The first aims to reconstruct depth maps of transparent objects, so as to mitigate the sensor failures in depth images. In [11], an approach was proposed that matches pixels from Time-of-Flight images first and then reconstructs an approximated surface with triangulating methods. To enhance the matching speed and reduce the influence of noise, a method was proposed in [12] to match transparent edges instead of all pixels. However, those methods require multiple views of one object, which is not suitable for the case when the camera is fixed. To address this challenge, some other studies [3], [13], [14] focus on reconstructing the missing or noisy depth regions of transparent objects using a single RGB-D image. In [3], a global optimisation algorithm was adopted to reconstruct the depth values that are removed based on predicted object masks. In [13], a local implicit neural representation built on ray-voxel pairs was proposed to reconstruct depth information incorporated with an iterative self-correcting refinement model. In [14], an affordance-based depth reconstruction framework was proposed to facilitate the robotic manipulation of transparent objects.

Rather than reconstructing a depth map, [4], [15] generate the grasp proposal with only RGB images or noisy depth maps as input. In [4], transfer learning was used to transfer the grasping model trained on depth maps to transparent object grasping with RGB images. In [15], a two-stage approach was proposed to estimate 6-DoF pose of transparent objects from a single RGB-D image, which can be used to assist transparent object grasping. Nonetheless, there has been no work on transparent object grasping with both visual and tactile information. To our best knowledge, this is the first work to achieve the task.

B. Object Grasping using Vision and Touch

The coordination between vision and touch sensing plays an important role in robot perception and has been applied to a number of different tasks [16] such as object recognition [17], [18], shape exploration [19], and object grasping [20], [21]. The visual-tactile features can be fused via direct concatenation [20] or a Self-Attention mechanism [22]. Moreover, the coordination of vision and touch allows us to develop regrasping policies that will best grasp the object [20]. However, the above studies either assume that the object position is known or use the depth information from camera to localise the object. These assumptions are not suitable for detecting and grasping transparent objects that are placed at a random location on the table due to their noisy or missing depth maps. In contrast, in this work, we use visual feedback to provide geometric cues for guiding the tactile sensor to contact the transparent object, which facilitates its grasp.

C. Sim2Real Learning for Transparent Objects

Synthetic datasets have been used in a wide array of applications, such as object segmentation [23], human pose estimation [24], and tactile object classification [25]. However, there are only a few synthetic datasets for transparent objects. Most of those datasets [26], [13] were generated without considering the subtle effect of transparent objects, e.g., specular highlights and caustics. In contrast, our simulation method can not only generate realistic images of transparent objects with such effects considered, but also provide detailed annotations, using the LuxCoreRender [27] engine. Compared to the rendering method used in [3] that uses Cycles [28] engine, our method generates more natural synthetic images and builds the glass shader in an easier way.

Fig. 2: There are 9 objects in both synthetic (the first row) and real-world datasets (the second row), from left to right: large disposable cup, highball cup, rectangular cup, vial, jar, mug, small disposable cup, champagne cup and tumble cup.

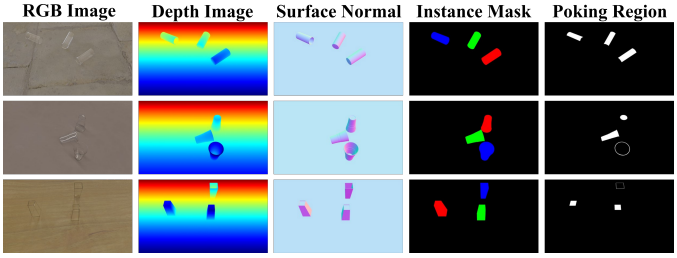

Fig. 3: Three visualisation examples of our synthetic dataset. The first four columns are RGB images, depth images, surface normals and instance masks of three scenes with a few transparent objects rendered in Blender, respectively, and the poking region masks in the last column are generated from them.

III. System Setup

In this section, we first introduce a high-quality synthetic dataset generation framework. It can auto-generate the poking areas that are hard to be annotated by humans, e.g., the side surface of a cylindrical cup. Then we introduce the robot setup used for the real world data collection and experiments. We introduce the system setup before the methodology as we believe the synthetic data generation will also be a contribution of this paper, and can help the reader better understand the methods to be introduced later.

A. Synthetic Data Generation

We first generate the synthetic data of transparent objects, i.e., RGB images, depth images, surface normals and instance masks, before we conduct real world experiments, due to two reasons. First, key cues that determine the poking regions, i.e., surface normals, cannot be obtained in real world experiments. Second, labels like instance masks of transparent objects can be generated automatically in synthetic data, whereas it is challenging and time consuming for human annotators to annotate instance masks in real data of transparent objects.

We use Blender's physics engine [29] and LuxCoreRender rendering engine [27] to generate our synthetic dataset. Through simulating the flow of light, LuxCoreRender can not only produce photo-realistic images, but also simulate important effects caused by the presence of transparent objects such as reflections and caustics. The dataset consists of 9 objects modelled after real-world transparent glass objects, as shown in Fig. 2. To enrich the variety of the synthetic data, we employed 33 HDRI lighting environments and 20 textures for the ground plane underneath the transparent objects.

To bridge the gap between simulation and real environments, we first set the camera intrinsics based on the parameters of the Intel RealSense D415 camera that we use in the real experiments. Then we randomly select one HDRI lighting environment and one ground plane surface texture applied with a random rotation angle for each scene. Finally, several CAD model objects were created above the plane surface to increase the learning efficiency. For each scene, the ground truth data from Blender includes: (1) rendered monocular RGB image, (2) aligned depth in meters, (3) instance masks of all transparent objects, (4) the camera pose, and (5) surface normals of the scene.

Using the rendered data, the ground truth of poking regions can be generated as follows. First, we get the dot product map via calculating the dot product of each pixel and the table surface normal. Then, we apply a pre-defined threshold to the dot product map to get initial poking regions. However, not all the initial poking regions are suitable for tactile poking, for example, the inner surface of a cup. To remove those areas, we calculate the height map relative to the ground plane using the depth image and the camera pose. If the height of one pixel is lower than a predefined threshold, the pixel will not be set as part of the poking region. Figure 3 shows some examples of rendered images and their corresponding ground truth of poking regions for transparent objects. In total, there are over 9,000 views of 9 objects generated in the sythetic dataset.

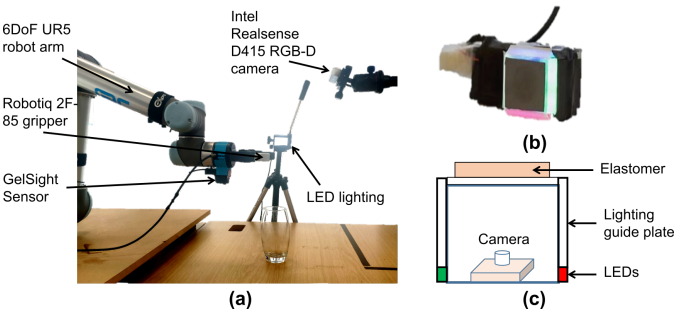

Fig. 4: **Robot setup. (a)** An overview of the experimental setup that consists of a UR5 robotic arm, Robotiq 2F-85 gripper, a GelSight sensor and an Intel RealSense D415; **(b)** The GelSight Sensor; **(c)** A sketch diagram of the GelSight sensor.

B. Robot Setup for Transparent Object Grasping

As shown in Fig. 4, our robot setup consists of a 6-DOF UR5 robot equipped with a Robotiq 2F-85 adaptive gripper and a GelSight sensor, as well as an Intel RealSense D415 RGB-D camera mounted on the tripod for overseeing the environment. The GelSight is a camera-based optical tactile sensor that can detect contact and capture fine details of the object surface. In a GelSight sensor, a webcam is placed under an elastomer that captures the deformations of the elastomer on the top when contacted. The sensor has a flat sensing area of $14mm \times 10.5mm$ and can capture tactile images with a resolution of 1280×720 at a frequency of 30 Hz [30]. In previous works [20], the gripper's fingers were replaced with GelSight sensors to generate tactile images for grasped objects. However, this replacement limits the sensing area to the inner side of the finger. To address this limitation, as shown in Fig. 4(a), the GelSight sensor is attached to the end-effector's side with a 90° angle to achieve a tactile poking action, i.e., the GelSight's y-axis and z-axis coincide with the end-effector's y-axis and x-axis, respectively. The rotation between the end-effector and the GelSight sensor is then set as $(0, \frac{\pi}{2}, 0)$, and the translation is obtained with an Opti-Track

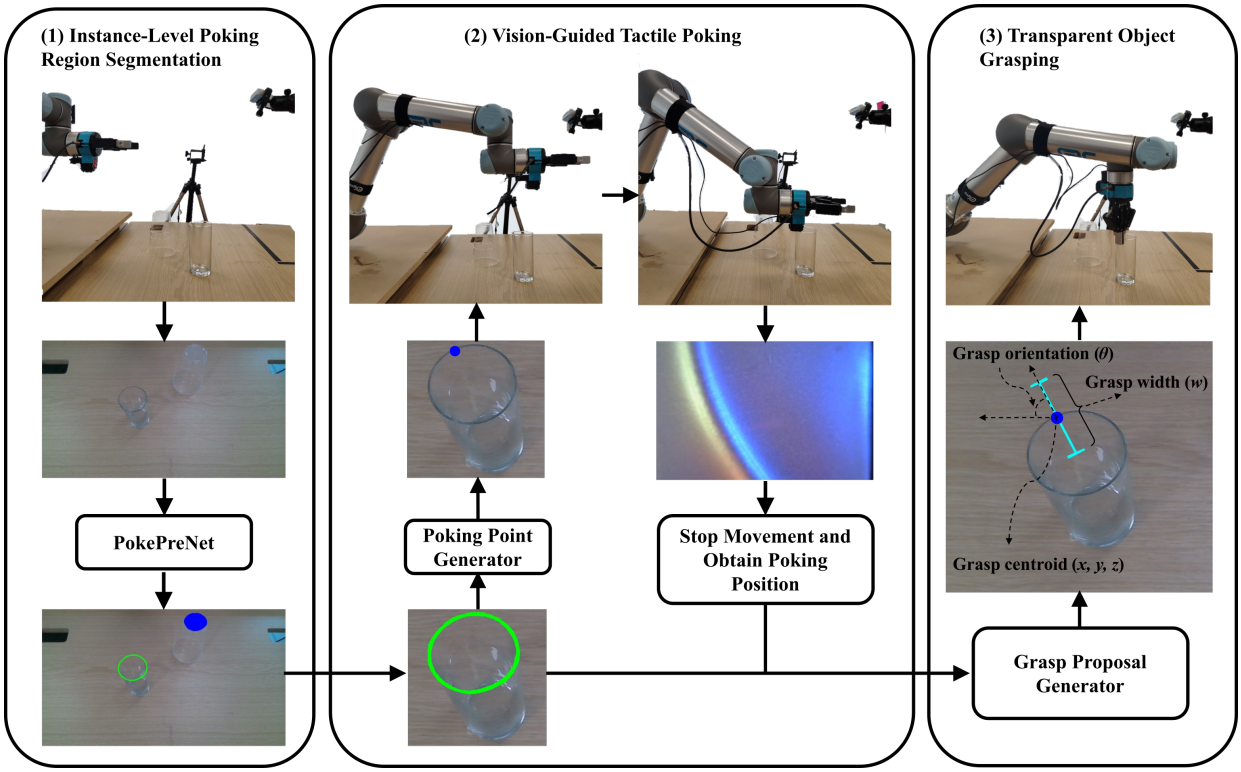

Fig. 5: An overview of our vision-guided tactile poking approach for transparent object grasping. **From left to right:** First, the PokePreNet takes the RGB image, and outputs the segmented poking regions where different colours represent different instances. Then based on the detected poking regions, the poking point generator is used to generate the potential poking point that guides the robotic arm to move towards the transparent object until the equipped GelSight sensor contacts the object. Lastly, with predicted poking region and the obtained local profiles from tactile poking, a heuristic grasp proposal is generated for grasping the transparent object.

Motion Capture system. Moreover, the classic Tsai hand-eye calibration method [31] is applied to the calibration between the RealSense camera and the UR5 robot.

C. Real-World Dataset Collection

To test the generalisation of the proposed PokePreNet, we also create a dataset of real-world transparent objects in the laboratory space. As illustrated in Fig. 2, there are 4 transparent plastic and 5 glass objects in both our synthetic and real-world datasets. The real-world dataset consists of 180 images of those nine known objects used in synthetic training data. Each image contains one object randomly placed on the table. Due to the difficulty of annotating the side surface of cylindrical objects, only the rectangular cup and the jar, i.e., the third column and fifth column in Fig. 2, have the cases where they stand on their sides.

IV. Methodology

In this work, we propose a vision-guided tactile poking approach for transparent object grasping, with an overview of the framework illustrated in Fig. 5. First, the poking region prediction network (*PokePreNet*) takes a single RGB image and outputs the poking region segmentation in the instance level. Based on the detected poking region, a poking point is then generated to guide the robotic arm to move towards the transparent object. The robotic arm will stop once a contact between the equipped GelSight sensor and the object is detected. Finally, with the predicted poking region and obtained local profiles from tactile poking, a heuristic grasp proposal is generated for grasping the transparent object.

A. Poking Region Segmentation

The poking region segmentation is treated as an instance segmentation problem. In the instance segmentation, every pixel will be simultaneously classified whether it belongs to the poking region and which instance it is part of. One of the most popular instance segmentation techniques is Mask R-CNN [32]. However, the poking region only occupies a small part of the bounding box, which causes a bad precision of Mask R-CNN. To solve this issue, our PokePreNet introduces two novel improvements to the original Mask R-CNN for segmenting the poking regions: (1) a larger output feature map via adding more deconvolutional layers; (2) a new pixel-level Positive-Negative-balanced loss.

Larger output feature map. We add two more deconvolutional layers to increase the size of poking region masks from 28×28 to 112×112. The filters in all the deconvolutional layers have a size S_f of 2×2, with zero padding $d = 0$ and stride $s = 2$, which can double the size of the feature map:

$$S_o = s * (S_i - 1) + S_f - 2 * d \tag{1}$$

where S_i and S_o are the sizes of the input feature map and the output feature map, respectively.

Pixel-level Positive-Negative-balanced loss. We use a multi-task loss L to jointly train the instance segmentation network

to predict the object class, bounding box position, and poking region mask on each Region of Interest (RoI) as follows:

$$L = L_{cls} + L_{loc} + L_{mask} \tag{2}$$

where L_{cls} is the multinomial cross entropy loss; L_{loc} is the *Smooth L1* loss [33] between the regressed box offsets $t = \{t_x, t_y, t_w, t_h\}$ and the ground-truth box offsets $v = \{v_x, v_y, v_w, v_h\}$:

$$L_{loc}(t, v) = \sum_{k \in \{x,y,w,h\}} Smooth_{L1}(t_k - v_k) \tag{3}$$

where

$$Smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases} \tag{4}$$

L_{mask} is the poking region mask loss. Following [32], [33], the weighting factors for each sub-loss are set to 1.

In the vanilla Mask R-CNN, the average binary cross-entropy loss is used for training instance masks. However, the distribution of positive/negative pixels (positive pixels are the pixels that are part of the poking regions, and negative pixels are ones that are not) in the objects such as the cup in Fig. 6 is heavily biased: only 5% of the bounding box area is part of the poking region. To this end, the cross-entropy loss from the poking region only contributes to a small part of the total loss, and leads to a bad precision of the poking region.

To address this issue, we define the following pixel-level Positive-Negative-balanced (PN) loss for the poking region mask L_{mask} in Eq. 2:

$$\begin{aligned} L_{mask}(X_i) = -\beta_i \sum_{j \in Y_i^+} \log \Pr\left(y_j = 1 | \, X_i\right) \\ - \sum_{j \in Y_i^-} \log \Pr\left(y_j = 0\right) | X_i) \end{aligned} \tag{5}$$

where Y_i^+ and Y_i^- denote the positive and negative ground truth label sets for the i^{th} RoI X_i, respectively; β_i is the weight on an instance basis to balance the loss between positive and negative pixels, as illustrated in Fig. 6.

Specifically, β_i is set to $|Y_i^-|/|Y_i^+|$ and 1 when $|Y_i^+|$ is larger than 0 and equal to 0, respectively. $|.|$ function is used for calculating the set size, and j represents the pixel index. $\Pr\left(y_j = 1 | \, X_i\right) = \sigma\left(a_j\right) \in [0, 1]$ is computed using sigmoid function $\sigma\left(.\right)$ on the activation value a_j at pixel j.

In our initial experiments, we find that the PN loss boosts the poking region segmentation performance in the experiments in the real environment. However, in our synthetic dataset results, there are some extremely small poking regions and as a result β_i is very large in such cases, which results into a large number of false positives and lowers the performance. Hence, to enhance the performance in the simulation, we use a log function to restrict large values and use a Log-Positive-Negative-balanced (LPN) loss for L_{mask} instead with β_i:

$$\beta_i = \begin{cases} \ln(\frac{|Y_i^-|}{|Y_i^+|}) & \text{if} \quad |Y_i^+| > 0 \\ 1 & \text{if} \quad |Y_i^+| = 0 \end{cases} \tag{6}$$

The proposed method can be recognised as a kind of Hard Example Mining method [34], i.e., mining of examples that are hard to be classified or detected. The hard examples in this work are the pixels from the instance with a small poking region where a biased pixel distribution exists. Hence, we use this heuristic to accelerate the mining of hard examples without the need to classify each pixel in each RoI individually, which makes our method more efficient.

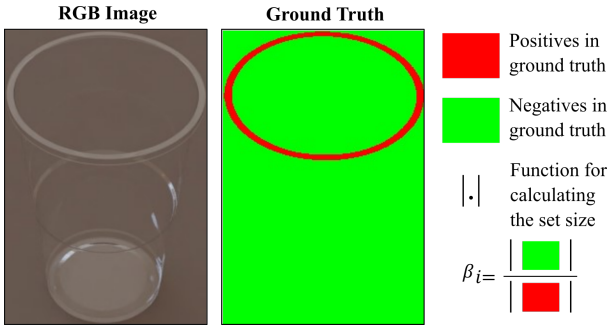

Fig. 6: An illustration of how the Positive-Negative-balanced weight β_i is computed. Red pixels in the ground truth are the positive pixels that are part of the poking region, whereas green pixels are the negative pixels that are not part of the poking region.

B. Vision-guided Tactile Poking

Given the detected poking region from Sec. IV-A, we generate a poking point $\boldsymbol{P_t} = [x_t, y_t]$ in the image frame for every transparent object. To generate the poking point, we first find the external contour of the poking region mask using OpenCV function *findContours*. Then we use OpenCV function *fitEllipse* to fit the contour and get the centroid $\boldsymbol{P_c}$. Similar to [35], the poking points are generated based on the primitive shapes. As shown in the output of our PokePreNet in Fig. 5, the 2-D poking regions are simplified into two types of primitive shapes: a simply connected mask (the blue mask) if $\boldsymbol{P_c}$ is part of the poking region, and a ring shape connected mask (the green mask) if $\boldsymbol{P_c}$ is out of the poking region.

If the poking region is a simply connected mask, the poking point will be set to the ellipse centroid $\boldsymbol{P_c}$, as centroids are widely used for grasping the objects with simple rectangular or cylindrical shapes [36]. On the other hand, if the poking region is a ring shape mask, $\boldsymbol{P_c}$'s nearest positive pixel will be set as the poking point to avoid getting the GelSight sensor into the object. The algorithm used to find the poking point in the image frame is summarised in Alg. 1.

Algorithm 1 Poking point generation

Input: M_{poking}: a poking region mask.
Output: $\boldsymbol{P_t} = [x_t, y_t]$: a poking point in the image frame

```
external_contour ← findContours(M_poking)
ellipse ← fitEllipse(external_contour)
if ellipse.centroid in M_poking then
    [x_t, y_t] ← ellipse.centroid
else
    [x_t, y_t] ← findNearestPositive(ellipse.centroid)
end if
```

Guided by the poking point, the robotic arm is moved towards the transparent object until the equipped GelSight sensor contacts the object. The GelSight sensor is set parallel

to the table so as to minimise the horizontal force and avoid the change of the object's state. The tactile contact is detected with a simple image subtraction-based algorithm [37]. First, a tactile image is captured as the reference. Then, the element-wise absolute difference between the reference and the current frame is computed and applied with a binary thresholding in every channel. Finally, the contact will be recognised in the current frame, if the number of positive pixels in the difference frame is larger than a predefined threshold. We also considered detecting the contact by thresholding the Structural Similarity Index Measure (SSIM) of the reference and contact images. However, compared to the image subtraction-based algorithm, the computational cost of SSIM is much higher (0.2s vs. 0.02s for processing each tactile image). To stop the robotic arm as soon as a contact is detected and avoid destroying fragile transparent objects, the image-subtraction method is used.

C. Heuristic Transparent Object Grasping

Based on the predicted poking region and the object's local profiles (i.e., contact position) from the tactile poking, a heuristic grasp representation in the world frame is generated for the top-down parallel grasping. The grasp representation is defined as a 5-dimensional vector $\boldsymbol{G_{hrst}} = [x, y, z, w, \theta]$ as shown in Fig. 5, where $[x, y, z]$ represents the grasp centroid in the world frame. w and θ represent the width and the orientation of the heuristic grasp, respectively. Note that θ is the one-dimensional angle around the vertical axis of gravity direction to facilitate the top-down parallel grasping.

If $\boldsymbol{P_c}$ belongs to the poking region, the poking position $\boldsymbol{P_t^W}$ in the world frame will be equal to the position of centre $\boldsymbol{P_c^W}$. Hence, a centroid-based grasp [36] is used for grasping the transparent object. In detail, $[x, y, z]$ of $\boldsymbol{G_{hrst}}$ will be set to $\boldsymbol{P_t^W}$. The grasp width w and the orientation θ are set to the maximum value of gripper width, and the fitted ellipse rotation angle for grasping along the short axis of the ellipse, respectively. If $\boldsymbol{P_c}$ is not part of the poking region, the grasp centroid will be set according to the distance $D(\boldsymbol{P_c^W}, \boldsymbol{P_t^W})$ between $\boldsymbol{P_c}$ and $\boldsymbol{P_t}$ in the world frame. Under the assumption that $\boldsymbol{P_c^W}$ and $\boldsymbol{P_t^W}$ are at the same height, the centroid of the fitted ellipse in the world frame $\boldsymbol{P_c^W}$ can be calculated with a pin-hole camera model.

If D is larger than the half of the finger width, the gripper finger could be inserted into the transparent object. Hence, an edge grasp is used for grasping the transparent object as the grasp proposal shown in Fig. 5. $[x, y, z]$ will be the poking position $\boldsymbol{P_t^W}$. The grasp width w and the orientation θ are set to the twice of D and parallel to the vector $< \boldsymbol{P_c^W}, \boldsymbol{P_t^W} >$, respectively. Otherwise, a centroid-based grasp is used and the grasp position will be set to $\boldsymbol{P_c^W}$. The grasp width w and orientation θ are set to the maximum value and the fitted ellipse rotation angle. The algorithm used to generate the heuristic grasp is summarised in Alg. 2.

Algorithm 2 Heuristic grasp generation

Input: $\boldsymbol{P_t^W}$: poking position in the world frame; M_{poking}: a poking region mask; ellipse: fitted ellipse from Alg. 1.
Output: $\boldsymbol{G_{hrst}} = [x, y, z, w, \theta]$: a heuristic grasp proposal.

```
if ellipse.centroid in M_poking then                    ▷ centroid grasp
    [x, y, z] ← P_c^W ← P_t^W
    w ← maximum_gripper_width
    θ ← ellipse.rotation_angle
else
    P_c^W ← calculateWorldPosition(ellipse.centroid)
    D ← calculateDistance(P_c^W, P_t^W)
    Angle ← calculateAngle(P_c^W, P_t^W)
    if D > 0.5×finger_width then                        ▷ edge grasp
        [x, y, z] ← P_t^W
        w ← 2×D
        θ ← Angle
    else                                                ▷ centroid grasp
        [x, y, z] ← P_c^W
        w ← maximum_gripper_width
        θ ← ellipse.rotation_angle
    end if
end if
```

V. Experiments

In this section, we conduct a series of experiments to evaluate our vision-guided tactile poking for transparent objects grasping. The goal of the experiments are three-fold: (1) To evaluate the poking region segmentation accuracy of our PokePreNet in both synthetic and real-world datasets; (2) To investigate how poking regions improve the success rate of tactile poking against bounding boxes and instance masks; (3) To investigate how the feedback from tactile poking can improve the success rate of transparent objects grasping.

A. Poking Region Segmentation Experiments

To evaluate the poking region segmentation accuracy, the standard Average Precision (AP) metric is used. To be more specific, we used mean AP (mAP), AP_{50} and AP_{75}, i.e., AP at different Intersection over Union (IoU) thresholds 50% and 75%, and AP_S, AP_M and AP_L (AP at different scales, i.e., small, medium and large). It should be noted that the object scale is determined by the poking region size instead of the bounding box size. We only evaluate the poking region segmentation results in this work, as the bounding box detection is not related to our tactile poking approach. Similar to the previous studies [32], our PokePreNet uses a Region Proposal Network to extract 1000 proposals for each image. Our poking region segmentation experiments are organised as follows. Firstly, we compare our Positive-Negative-balanced loss (PN) and Log-Positive-Negative-balanced loss (LPN) against vanilla cross-entropy loss and weighted cross-entropy loss in both the synthetic and real-world datasets. Secondly, we analyse the effect of the output size of the poking region map. Thirdly, we examine the domain randomisation's effect on generalisation.

Evaluation of different loss functions. We evaluate PokePreNet on both the synthetic and real-world benchmarks. Table I compares the performance of using different types of losses for poking region segmentation. The vanilla loss represents the average binary cross-entropy loss used in the vanilla Mask R-CNN. Weighted loss adds a fixed large weight

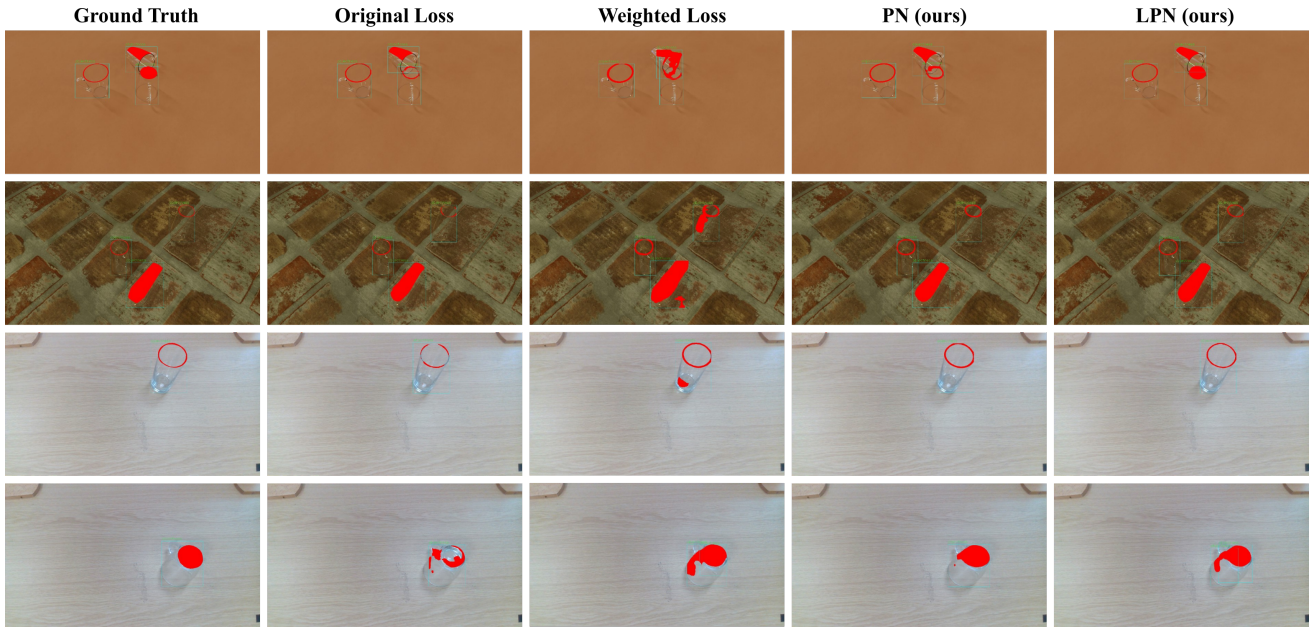

Fig. 7: Visual comparison of poking region segmentation results using different loss functions. The top two rows and the bottom two rows compare results on the synthetic dataset and the real-world dataset, respectively. As shown above, our PN and LPN based methods generate much better poking regions, compared to the vanilla loss and weighted loss.

to the cross-entropy loss of positive pixels.

In the synthetic dataset, the weighted loss and the PN loss although bring 7.9% and 3.5% gains in terms of AP_S respectively, nevertheless lead to a significant drop of the overall performance. This is because when the poking region areas are extremely small in the synthetic dataset, the balanced weight on positive pixels will result in more false positives and lower the performance. After using the log function to compress the value range of the balanced weight in PN loss, the LPN loss achieves an improvement of 8.7% and 3.2% on AP_S and mAP, respectively.

In the real-world test benchmark, both our PN loss and LPN loss outperform the original loss and the weighted loss: PN loss leads to the best overall performance and LPN loss results in largest improvement on AP_M. These different trends might be caused by two main reasons: (1) Manual annotations of the real-world benchmark is not as accurate as synthetic annotations; (2) Domain randomisation is not sufficient to bridge the domain gap between the simulation and the real world. To address this problem, domain adaptation will be considered in the future work. We also show the qualitative results of poking region segmentation in Fig. 7. As illustrated, the original loss and the weighted loss result in a lot of false negatives and false positives. Our PN loss and LPN loss yield highest quality poking region segmentation results on the real-world images and the synthetic images.

TABLE I: BASELINE COMPARISONS ON SYNTHETIC AND REAL BENCHMARK.

Test data	Loss type	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Synthetic	vanilla	0.530	0.843	0.600	0.020	**0.509**	0.744
Synthetic	weighted	0.468	0.865	0.467	0.099	0.399	0.733
Synthetic	PN [ours]	0.472	0.749	0.500	0.053	0.388	0.752
Synthetic	LPN [ours]	**0.562**	**0.916**	**0.601**	**0.107**	0.507	**0.775**
Real	vanilla	0.319	0.672	0.248	N/A	0.149	0.540
Real	weighted	0.330	0.669	0.292	N/A	0.155	0.542
Real	PN [ours]	**0.360**	**0.778**	**0.304**	N/A	0.181	**0.576**
Real	LPN [ours]	0.356	0.757	0.221	N/A	**0.234**	0.536

Evaluation of the poking region output feature map size. We also analyse the effect of the poking region output feature map size. Mask R-CNN uses only one deconvolutional layer to create the 28×28 mask map from the 14×14 feature map. Following the setup of Mask R-CNN, we add one, two, and three more deconvolutional layers to create the 56×56, 112×112, and 224×224 poking region mask map, respectively.

Table II summarises the segmentation accuracy of the mentioned networks on both the synthetic dataset and real-world dataset. The results show that the segmentation accuracy is gradually improved when the output size is increased from 28×28 to 112×112, however, it is not further improved when adding one more deconvolutional layer to make the output size to 224×224, on both synthetic and real datasets. It should be noted that similar results have also been reported in [38]. One possible reason is that 112×112 is large enough to show the details of the object and, compared to 224×224, is closer to the real feature map size of transparent objects. Moreover, the best models tested in the synthetic dataset and the real dataset are trained with our LPN loss and PN loss, respectively.

TABLE II: EFFECT OF OUTPUT MASK SIZE.

Test data	Output size	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Synthetic	28×28	0.397	0.654	0.425	0.018	0.299	0.660
Synthetic	56×56	0.442	0.708	0.477	0.051	0.354	0.752
Synthetic	112×112	**0.562**	**0.916**	**0.601**	**0.107**	**0.507**	**0.775**
Synthetic	224×224	0.501	0.868	0.501	0.064	0.398	0.749
Real	28×28	0.198	0.425	0.183	N/A	0.081	0.337
Real	56×56	0.271	0.555	0.224	N/A	0.117	0.454
Real	112×112	**0.360**	**0.778**	**0.304**	N/A	**0.181**	**0.576**
Real	224×224	0.337	0.751	0.281	N/A	0.170	0.554

Evaluation of domain randomisation. Despite not being trained on real transparent objects for poking region segmentation, our models can be adapted well to the real-world domain. To evaluate the importance of our data generation methodology, we assessed the model's sensitivity to the number of

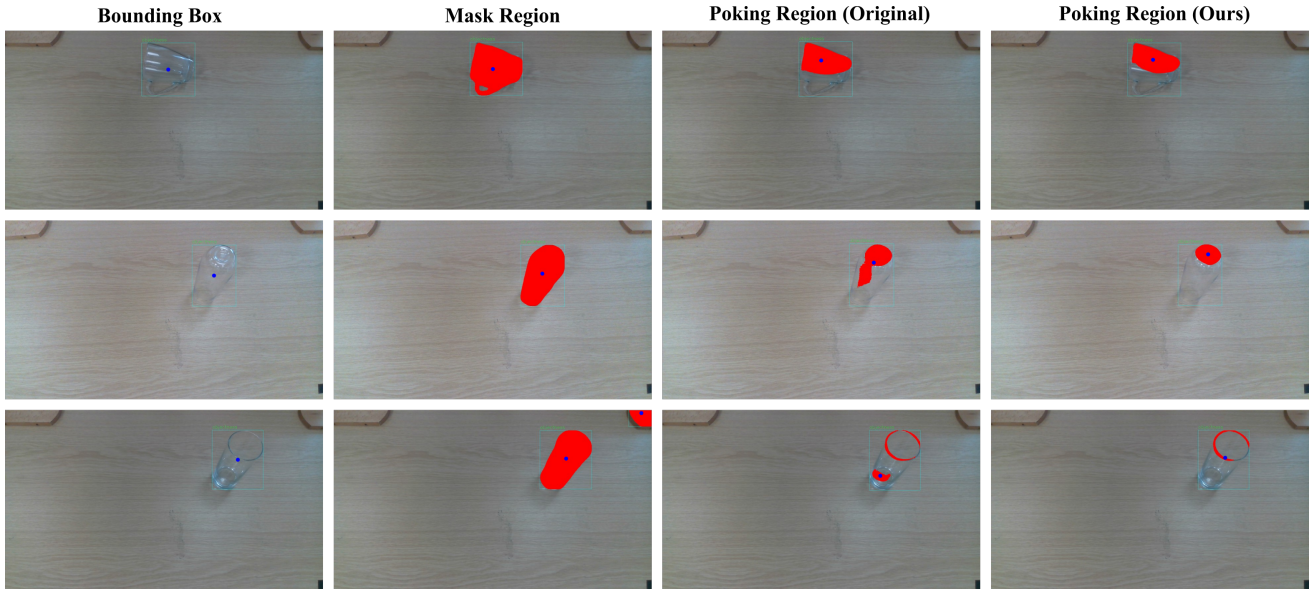

Fig. 8: Examples of the poking points generated with the bounding box, the instance mask, the poking region with vanilla Mask R-CNN loss (Original) and the poking region with our PN loss (Ours). The red colour and blue dot represent the segmentation results and generated poking points, respectively.

unique textures seen in the training. Table III shows that the domain randomisation method via applying different textures significantly improves the mAP of poking region segmentation accuracy in real-world dataset from 31.3% to 36.0%.

TABLE III: EFFECT OF DOMAIN RANDOMISATION (DR).

Test data	DR	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Synthetic	×	0.472	0.803	0.515	0.073	0.420	0.672
Synthetic	√	**0.562**	**0.916**	**0.601**	**0.107**	**0.507**	**0.775**
Real	×	0.313	0.697	0.249	N/A	0.160	0.506
Real	√	**0.360**	**0.778**	**0.304**	N/A	**0.181**	**0.576**

B. Vision-Guided Tactile Poking Experiments

In this subsection, we conduct real-world experiments to evaluate the performance of our vision-guided tactile poking method. We define that the poking action will be recognised as successful, if no protective stop happens to the robotic arm, and a contact happens between the transparent object and the GelSight sensor. During the poking motion, the robotic arm will stop if the height of the end-effector is lower than a predefined threshold, which means the poke misses the object entirely and will be taken as a failure.

Table IV compares the success rates of vision-guided tactile poking methods using four different input sources for poking point generation. The "bounding box" and "mask region" approaches guide the tactile poking with the centroid position of predicted bounding boxes and predicted instance masks, respectively. The "poking region" represents those methods that use the poking region segmentation results as the input of poking point generator. [Vanilla] and [Ours] respectively represent the PokePreNet trained with the vanilla binary cross-entropy loss and our PN loss. Similar to [3], we had 12 attempts in grasping of each object, i.e., a total of 108 attempts for testing the above approaches. The results show that the poking region is a better cue for guiding the tactile poking compared to bounding box and instance mask, and the better poking region segmentation contributed by our PN loss can further improve the poking success rate from 84.3% to 89.8%.

The poking points generated with different methods have been visualised in Fig. 8. It is noticed that the generated poking points based on bounding box and instance mask sometimes have different surface normals against the table surface e.g., the first and second columns), which will lead to failed poking motions. We can also observe that the bad poking region segmentation caused by the vanilla cross-entropy loss can also result in a failed tactile poking as shown in the third column.

TABLE IV: COMPARISON ON VISION-GUIDED TACTILE POKING.

Object Category	BBox	Mask	PR (Vanilla)	PR (Ours)
Big disposable cup	4/12	4/12	8/12	8/12
Highball cup	5/12	5/12	9/12	11/12
Rectangular cup	7/12	6/12	10/12	10/12
Vial	8/12	8/12	12/12	12/12
Jar	12/12	12/12	12/12	12/12
Mug	6/12	8/12	10/12	12/12
Small disposable cup	5/12	6/12	8/12	8/12
Champagne cup	5/12	6/12	11/12	12/12
Tumble cup	6/12	7/12	11/12	12/12
Average success rate	53.7%	57.4%	84.3%	89.8%

C. Transparent Object Grasping Experiments

To demonstrate the advantage of vision-guided tactile poking, we compare four different objects grasping approaches, as shown in Table V. "Baseline1" and "Baseline2" generate grasp proposals from object instance masks and poking regions (PR) based on the depth obtained from an RGB-D camera, respectively. In contrast, our methods use the contact position from tactile poking to generate the heuristic grasp proposal. Same as the tactile poking experiment, every grasping approach was tested with 12 attempts on each object including 4 attempts with the object upright, 4 attempts with the object upside down, and 4 attempts with the object standing on its sides.

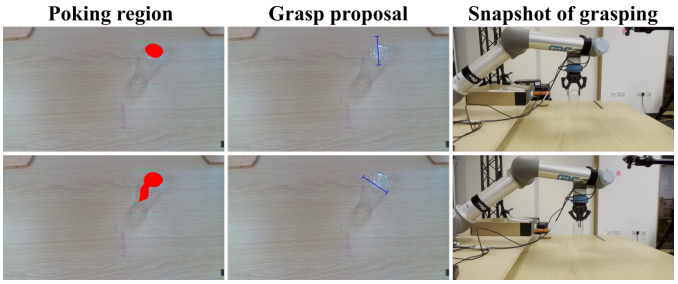

Fig. 9: Examples of successful and failed grasps. **Top**: A successful grasp contributed by the poking region predicted with our PokePreNet. **Bottom**: A failure grasp caused by bad poking region segmentation when using the vanilla cross-entropy loss. The poking region, grasp proposal and snapshot of grasping are shown for each case.

TABLE V: COMPARISON ON TRANSPARENT OBJECT GRASPING.

	Baseline1	Baseline2	Ours1	Ours2
Region Type	Masks	PR	PR	PR
Loss	vanilla	PN	vanilla	PN
Localisation source	camera	camera	poking	poking
Big disposable cup	4/12	4/12	8/12	8/12
Highball cup	2/12	3/12	8/12	10/12
Rectangular cup	4/12	5/12	9/12	10/12
Vial	4/12	4/12	11/12	11/12
Jar	8/12	8/12	12/12	12/12
Mug	2/12	4/12	8/12	10/12
Small disposable cup	4/12	5/12	8/12	8/12
Champagne cup	4/12	4/12	10/12	11/12
Tumble cup	3/12	5/12	10/12	12/12
Average success rate	32.4%	38.9%	77.8%	85.2%

As reported in Table V, the transparent objects are hard to grasp with "Baseline1" or "Baseline2" due to their noisy and missing depth information. Moreover, our methods significantly improve the grasping success rate from 38.9% to 77.8% and 85.2% via using the accurate local profile (i.e., contact position) from vision-guided tactile poking. Similar to the results in tactile poking experiments, the bad poking region segmentation results caused by the vanilla cross-entropy loss can result in a failed grasp, i.e., the second column in Fig. 9.

D. Tactile Alignment for Grasping Small Objects

Apart from providing the contact position for grasping, the tactile sensor can also sense the local shape of contact regions in the poking. Here, "*contact regions*" are the regions validated by the tactile sensor, whereas the above "*poking regions*" are from visual appearances and indicate the functional interactions of the object parts with humans or robots from the affordance perspective [39]. Due to hand-eye and sensor-end-effector calibration errors mentioned in Section III-B, there would be an offset between the expected poking point predicted from vision and the centre of the contact region. The offset will result in an error in estimating the centroid of the fitted ellipse detailed in Section IV-B and therefore deteriorate the performance of our centroid-based grasp. To address the offset, a tactile alignment method is used to rectify the estimated centroid using the local shape obtained from the tactile image. Due to the limited perceptive field of Gelsight sensor, we only test the tactile alignment method with the small vial (the 4th object in Fig. 2). It should be noted that the tactile alignment experiment is not the main focus of this paper, but to demonstrate the potential of the current work.

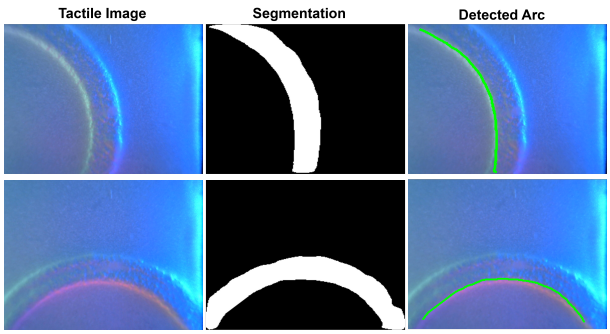

Fig. 10: Examples of the position alignment predictions using tactile readings.

The contact region is first obtained with a convolutional segmengtation neural network [40] as shown in Fig. 10. Similar to the centroid prediction in the visual images detailed in Section IV-B, the OpenCV function *findContours* is applied to the contact region to extract an arc of the inner ring of the vial's upper surface, and the OpenCV function *fitEllipse* is used to estimate its centroid position in the tactile image frame. The pin-hole camera model [40] is then applied to obtain the rectified centroid position of the vial's upper surface.

To validate that the tactile alignment can enhance the robustness of grasping, we introduce a random translation error ranging from -12 $\sim$ 12 mm in x-axis of the world frame to the hand-eye transformation. We test with 20 attempts of grasping the vial and observe that the tactile alignment can improve the grasping success rate of the vial from 80% to 100%.

E. Failures Analysis

Failures in Poking Region Segmentation. In the real world experiments, it has been noticed that thin poking regions of the rectangular cup (the 3rd object in Fig. 2) were hard to be detected when the object is placed upright. This was due to that the camera introduces noise to the captured images and as a result the poking regions are blurred. It could be improved by fine-tuning the PokePreNet with real dataset or incorporating it with other pixel-wise semantic segmentation methods.

Failures in Tactile Poking. One failure mode for tactile poking is the fall of transparent objects that are placed upright after being poked by the GelSight sensor. In this paper, we assume that contacting poking regions will generate reliable tactile readings, while causing minimum disturbance to the object state. However, for light objects such as disposable cups (the 1st and 8th objects in Fig. 2), the extremely small gravity cannot prevent the object from turning. As shown in Fig. 11(a), when the cup is under static equilibrium, the torques of the gravity G and the normal force F are equal, i.e., $F * d_2 = G * d_1$. As a result, the maximum force applied to the disposable cup by the GelSight sensor is around $0.1N$. The robotic arm cannot react in time to such a small force due to network latency and image processing, which will lead to a failed tactile poking with the falling of the disposable cup. The excessive torque could be avoided by evenly contacting the whole poking region at the same time using a larger tactile sensor.

Failures in Grasping. As shown in Tables IV and V, our approach can poke and grasp cylindrical objects without making

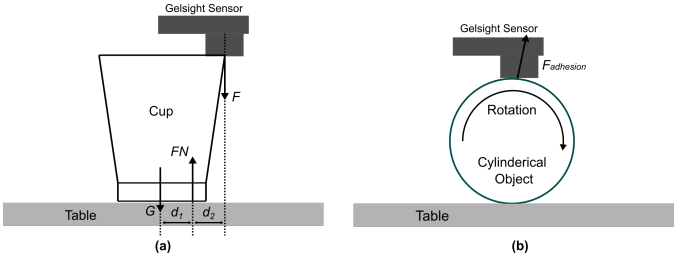

Fig. 11: Side views of the tactile poking, where the GelSight sensors are in contact with (a) a cup placed upright and (b) a cylindrical object placed on its side. (a): The force analysis of the cup under static equilibrium, where G, FN and F represent the gravity of the cup, the normal force from the table and the normal force from GelSight sensor, respectively. d_1 and d_2 represent the arms of G and F, respectively; (b): The force analysis of the cylindrical cup when an adhesion exists.

them roll on the table for most of the attempts. However, when the GelSight sensor moves away from the contacted object after poking, the adhesion between the GelSight's elastomer and the object might cause disturbance to the object's state as shown in Fig. 11(b), which will lead to a failed grasp. We could solve this problem with a dual-arm manipulation, i.e., one arm is used to poke and fix the object on the table, and the other arm is used to explore and grasp the object.

VI. CONCLUSIONS AND DISCUSSIONS

In this paper, we introduce a novel vision-guided tactile poking method for grasping transparent objects. Compared to previous methods, the proposed framework is the first that coordinates vision and tactile sensing to address the challenges of grasping transparent objects. The extensive experiments show that our proposed method can learn vision-guided tactile poking using only synthetic data for training and can generalise to the real world settings. The robot grasping experiments demonstrate that the informative local profile (i.e., position and local shape of contact regions) from tactile poking can enhance the performance of transparent objects grasping.

We have the robot poke the object for once to detect the contact and update the local profiles of transparent objects, which is different from the previous tactile exploration works that contacts the object multiple times [41], [42]. Tactile exploration can be used to estimate the object shape so as to facilitate grasping. However, in those works strong assumptions were made: either the object is 2D and the pushing process is quasi-static [43], [44] or the object is fixed on the table [42]. These assumptions are not suitable for our cases as 3D and movable objects are used in our investigated scenarios. For example, a cylindrical cup can roll a long way on the table with a small horizontal force. In this case, the assumptions in the tactile exploration will not stand any more as the cup is highly movable and is not quasi-static. In contrast, our vision-guided tactile poking method would output a good tactile reading while maintaining minimal disturbance to the object's state, so that the modelling of object dynamics is not needed.

The GelSight tactile sensor in this work plays two different roles. First, the GelSight sensor is used as a contact detector to validate the poking regions predicted by our PokePreNet, so as to replace the noisy depth from vision and facilitate the grasp. Second, as the GelSight sensor can extract the local shapes of small transparent objects, it is used to align grasp proposals to mitigate bias in the calibration error. When a large calibration error exists, there will be a large offset between the actual contact position and the expected position, which may lead to a failed grasping, and the geometric information of the object obtained from the tactile images can remedy grasping.

It is worth noting that the first role can also be fulfilled by other tactile sensors such as the tactile finger [45] and the GelTip sensor [46], or force sensors like Nano 17 force/torque sensors. It means that our proposed method is general and can be easily transferred to other settings. Force sensors with sensitive force estimation could achieve better control of the poking motion compared to the GelSight sensor. However, force sensors cannot replace tactile sensors for the second role without matrix-based force readings or high-resolution tactile images. Given that, we use the tactile alignment experiment to demonstrate the advantage of using a high-resolution GelSight sensor in vision-guided tactile poking. In the future work, we will investigate the tactile alignment for grasping transparent objects further without prior knowledge of the object shape.

REFERENCES

[1] S. Eppel, H. Xu, M. Bismuth, and A. Aspuru-Guzik, "Computer vision for recognition of materials and vessels in chemistry lab settings and the vector-labpics data set," *ACS Cent. Sci.*, vol. 6, no. 10, 2020.
[2] B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes, *et al.*, "A mobile robotic chemist," *Nature*, vol. 583, no. 7815, pp. 237–241, 2020.
[3] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 3634–3642.
[4] T. Weng, A. Pallankize, Y. Tang, O. Kroemer, and D. Held, "Multi-modal transfer learning for grasping transparent and specular objects," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 3791–3798, 2020.
[5] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Rev. Neurosci.*, vol. 10, no. 5, pp. 345–359, 2009.
[6] S. J. Lederman and R. L. Klatzky, "Haptic perception: A tutorial," *Atten. Percept. Psychophys.*, vol. 71, no. 7, pp. 1439–1459, 2009.
[7] P. Jenmalm, S. Dahlstedt, and R. S. Johansson, "Visual and tactile information about object-curvature control fingertip forces and grasp kinematics in human dexterous manipulation," *J. Neurophysiol.*, vol. 84, no. 6, pp. 2984–2997, 2000.
[8] T. G. Bower, J. M. Broughton, and M. Moore, "The coordination of visual and tactual input in infants," *Percept. Psychophys.*, 1970.
[9] T. M. Barrett, E. Traupman, and A. Needham, "Infants' visual anticipation of object structure in grasp planning," *Infant Behav. Develop.*, vol. 31, no. 1, pp. 1–9, 2008.
[10] A. Sheya and L. B. Smith, "Development through sensorimotor coordination," *Enaction: Toward a new paradigm for Cogn. sci.*, 2010.
[11] U. Klank, D. Carton, and M. Beetz, "Transparent object detection and reconstruction on a mobile platform," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 5971–5978.
[12] C. J. Phillips, M. Lecce, and K. Daniilidis, "Seeing glassware: from edge detection to pose estimation and shape recovery." in *Proc. Robot. Sci. Syst.*, vol. 3, 2016.
[13] L. Zhu, A. Mousavian, Y. Xiang, H. Mazhar, J. van Eenbergen, S. Debnath, and D. Fox, "Rgb-d local implicit function for depth completion of transparent objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4649–4658.
[14] J. Jiang, G. Cao, T.-T. Do, and S. Luo, "A4t: Hierarchical affordance detection for transparent objects depth reconstruction and manipulation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9826–9833, 2022.
[15] C. Xu, J. Chen, M. Yao, J. Zhou, L. Zhang, and Y. Liu, "6dof pose estimation of transparent object from a single rgb-d image," *Sensors*, vol. 20, no. 23, p. 6790, 2020.
[16] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, pp. 54–67, 2017.

[17] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2722–2727.
[18] J.-T. Lee, D. Bollegala, and S. Luo, ""Touching to See" and "Seeing to Feel": Robotic Cross-modal Sensory Data Generation for Visual-tactile Perception," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2019, pp. 4276–4282.
[19] S. Luo, W. Mou, K. Althoefer, and H. Liu, "Localizing the object contact through matching tactile features with visual map," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 3903–3908.
[20] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3300–3307, 2018.
[21] S. Cui, R. Wang, J. Wei, F. Li, and S. Wang, "Grasp state assessment of deformable objects using visual-tactile fusion perception," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 538–544.
[22] S. Cui, R. Wang, J. Wei, J. Hu, and S. Wang, "Self-attention based visual-tactile fusion learning for predicting grasp outcomes," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5827–5834, 2020.
[23] N. Li, C. Eastwood, and R. Fisher, "Learning object-centric representations of multi-object scenes from multiple views," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 5656–5666, 2020.
[24] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen, "Synthesizing training images for boosting human 3d pose estimation," in *Proc. IEEE Int. Conf. 3D Vis.*, 2016, pp. 479–488.
[25] D. F. Gomes, P. Paoletti, and S. Luo, "Generation of gelsight tactile images for sim2real learning," *IEEE Robot. Autom. Lett.*, pp. 4177–4184, 2021.
[26] G. Chen, K. Han, and K.-Y. K. Wong, "Tom-net: Learning transparent object matting from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9233–9241.
[27] LuxCoreRender - Open Source Physically Based Renderer, LuxCoreRender project, 2020. [Online]. Available: https://luxcorerender.org
[28] Cycles, Blender Cycles, 2019. [Online]. Available: https://docs.blender.org/manual/en/2.91/render/cycles/introduction.html
[29] Blender - a 3D modelling and rendering package, Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org
[30] G. Cao, Y. Zhou, D. Bollegala, and S. Luo, "Spatio-temporal attention model for tactile texture recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9896–9902.
[31] R. Y. Tsai, R. K. Lenz, *et al.*, "A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration," *IEEE Trans. robot. and autom.*, vol. 5, no. 3, pp. 345–358, 1989.
[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 91–99, 2015.
[34] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 761–769.
[35] Y. Lin, C. Tang, F.-J. Chu, and P. A. Vela, "Using synthetic data and deep networks to recognize primitive shapes for object grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 10 494–10 501.
[36] U. Asif, J. Tang, and S. Harrer, "Ensemblenet: Improving grasp detection using an ensemble of convolutional neural networks." in *BMVC*, 2018.
[37] D. F. Gomes, Z. Lin, and S. Luo, "Geltip: A finger-shaped optical tactile sensor for robotic manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9903–9909.
[38] G. Zhang, X. Lu, J. Tan, J. Li, Z. Zhang, Q. Li, and X. Hu, "Refinemask: Towards high-quality instance segmentation with fine-grained features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6861–6869.
[39] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 5908–5915.
[40] J. Jiang, G. Cao, D. F. Gomes, and S. Luo, "Vision-guided active tactile perception for crack detection and reconstruction," in *Proc. 29th Mediterranean Conf. Control Autom.*, 2021, pp. 930–936.
[41] C. Yang and N. F. Lepora, "Object exploration using vision and active touch," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6363–6370.
[42] D. Watkins-Valls, J. Varley, and P. Allen, "Multi-modal geometric learning for grasping and manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 7339–7345.
[43] K.-T. Yu, J. Leonard, and A. Rodriguez, "Shape and pose recovery from planar pushing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 1208–1215.
[44] S. Suresh, M. Bauza, K.-T. Yu, J. G. Mangelson, A. Rodriguez, and M. Kaess, "Tactile slam: Real-time inference of shape and pose from planar pushing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 11 322–11 328.
[45] P. Piacenza, K. Behrman, B. Schifferer, I. Kymissis, and M. Ciocarlie, "A sensorized multicurved robot finger with data-driven touch sensing via overlapping light signals," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 5, pp. 2416–2427, 2020.
[46] D. F. Gomes, Z. Lin, and S. Luo, "Blocks world of touch: Exploiting the advantages of all-around finger sensing in robot grasping," *Front. Robot. AI*, vol. 7, 2020.

Jiaqi Jiang received the B.S. and M.S. degrees from Beijing Institute of Technology, in 2016 and in 2019, respectively. He is currently a Ph.D. candidate in the Department of Engineering, King's College London. He was a Ph.D. candidate in the Department of Computer Science, the University of Liverpool. His research interests include robot grasping and sensory synergy of vision and touch.

Guanqun Cao received the B.S. degree from Nanjing Audit University, and the M.Sc. degree from the University of Liverpool. He is currently a Ph.D. candidate in the Department of Computer Science, the University of Liverpool. His research interests include tactile perception and multimodal perception.

Aaron Butterworth received an M.Eng. degree from the University of Liverpool in 2020. He is currently a Ph.D. candidate in the Department of Computer Science and Leverhulme Research Centre for Functional Materials Design. His research interests include robot grasping and synergy of visual and tactile perception.

Thanh-Toan Do is a Senior Lecturer at the Faculty of Information Technology, Monash University. He obtained his Ph.D. in Computer Science at the French National Institute for Research in Computer Science and Control (INRIA) in 2012. From 2013 to 2016, he was a Research Fellow at the Singapore University of Technology and Design. From 2016 to 2018, he was a Research Fellow at the Australian Centre for Robotic Vision and the University of Adelaide. From 2018 to 2020, he was a Lecturer at the University of Liverpool. His research interests include computer vision and machine learning.

Shan Luo is a Senior Lecturer (Associate Professor) at the Department of Engineering, King's College London. Previously, he was a Lecturer at the University of Liverpool, and Research Fellow at Harvard University and University of Leeds. He was also a Visiting Scientist at the Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT. He received the B.Eng. degree in Automatic Control from China University of Petroleum, Qingdao, China, in 2012. He was awarded the Ph.D. degree in Robotics from King's College London, UK, in 2016. His research interests include tactile sensing, robot learning and robot visual-tactile perception.