DefGoalNet: Contextual Goal Learning from Demonstrations for Deformable Object Manipulation

Bao Thach\textsuperscript{1,§}, Tanner Watts\textsuperscript{1,§}, Shing-Hei Ho\textsuperscript{1}, Tucker Hermans\textsuperscript{1,2}, Alan Kuntz\textsuperscript{1}

Abstract—Shape servoing, a robotic task dedicated to controlling objects to desired goal shapes, is a promising approach to deformable object manipulation. An issue arises, however, with the reliance on the specification of a goal shape. This goal has been obtained either by a laborious domain knowledge engineering process or by manually manipulating the object into the desired shape and capturing the goal shape at that specific moment, both of which are impractical in various robotic applications. In this paper, we solve this problem by developing a novel neural network DefGoalNet, which learns deformable object goal shapes directly from a small number of human demonstrations. We demonstrate our method’s effectiveness on various robotic tasks, both in simulation and on a physical robot. Notably, in the surgical retraction task, even when trained with as few as 10 demonstrations, our method achieves a median success percentage of nearly 90%. These results mark a substantial advancement in enabling shape servoing methods to bring deformable object manipulation closer to practical real-world applications.

I. INTRODUCTION

Deformable object manipulation defines a fundamental challenge in robotic manipulation due to its wide-ranging applications [1, 2]. Unlike rigid objects, deformable materials such as clothing, fabrics, soft tissues, or food items have intricate dynamics and infinite degrees of freedom. This complexity necessitates innovative techniques to enable robots to manipulate deformable objects effectively. Whether in healthcare (surgical robots for delicate tissue manipulation), manufacturing (handling soft materials like textiles), or homes (folding laundry), enabling effective deformable object manipulation empowers robots to perform a broader range of tasks, increasing their utility.

Shape servoing, a robotic task dedicated to controlling objects to desired goal shapes, has recently garnered significant attention from the deformable object manipulation community [3–15]. Our prior work in this area, DeformerNet [3], leverages point clouds as the state representation for deformable objects. DeformerNet defines a neural network that takes the object’s current and goal point clouds as inputs and computes the desired robot action to drive the object toward the target shape. However, a significant weakness of DeformerNet and other shape servoing methods is the requirement to explicitly define goal shapes, e.g., as a point cloud. Goals have previously been obtained either by laborious domain knowledge engineering or by manual manipulation of an object into the desired shape and capturing the shape at that specific moment—approaches that often prove impractical in various robotic applications.

In this work, we address this critical challenge by introducing DefGoalNet, a novel neural network that autonomously learns deformable object goal shapes from human task demonstrations. Our method fills the vital gap in DeformerNet by supplying it with a goal point cloud predicted from additional sensory information encoding the task context. We denote this as the contextual point cloud. The contextual point cloud provides crucial features for defining the success of a specific task. The design choice for the contextual point cloud varies by task but will generally include other task-relevant objects or environment components. When integrating DefGoalNet and DeformerNet into a unified pipeline, we transform the robot policy formulation, making it reliant solely on practically accessible parameters: the current point cloud of the deformable object and an additional point cloud encoding contextual information specific to the current task.

In our new learning-from-demonstration shape servoing pipeline, we first train DefGoalNet on a set of demonstration trajectories. This enables it to predict goal shapes that vary as task context changes. At runtime, our neural network reasons over the current deformable object point cloud and the contextual point cloud, generating a goal point cloud corresponding to a successful task outcome under that context. This output from DefGoalNet then becomes the input to DeformerNet, which computes the desired robot end-effector action to drive the deformable object toward the goal shape.
accomplishing the task in a closed-loop manner (see Fig. 1). Our pipeline decouples goal generation from the control policy. This in turn enables learning a robust control policy across a large set of related data (e.g., from simulation) while learning goals from potentially few samples of the target task.

We evaluate our method with experiments on surgery-inspired robotic tasks. We first conduct experiments in simulation, with simulated demonstrations, on two common surgical sub-tasks: retraction and tissue wrapping. We then perform a zero-shot sim-to-real transfer experiment with a physical robot, using the model entirely trained on simulated data. Finally, we train DefGoalNet with a small set of real human demonstrations and evaluate our method on a physical mock surgical retraction task. In all experiments, we find DefGoalNet achieves high performance. Notably, on a surgical retraction task, our method achieves high performance when trained on as few as 10 demonstrations.

While a small demonstration dataset can still lead to the successful completion of the task, our experiments show that increasing the size of the demonstration dataset results in more realistic and easily interpretable goal shape prediction. Visualizing these high-quality goal shapes as an intermediate step before executing the robot policy enhances safety and transparency in robot learning from demonstration. This is a key feature that distinguishes DefGoalNet from imitation learning methods. Additionally, DefGoalNet offers the benefit of not requiring a human to explicitly specify task success metrics, which can be challenging to encode and require significant engineering effort for each task. Instead, DefGoalNet enables learning task-specific goals directly from demonstrations.

Our paper is the first effort to tackle the challenge of learning goal shape specification in shape servoing tasks from demonstration. The experimental outcomes we have achieved represent a significant stride toward making shape servoing more applicable to real-world robotic applications.

We publish all code and data at: https://sites.google.com/view/defgoalnet/home.

II. RELATED WORK

Machine learning has enabled robots to manipulate rigid objects by harnessing complex, high-dimensional sensor data, such as point clouds [16–21]. Neural networks have revolutionized approaches to challenging robotic tasks, such as shape completion [21], pose estimation [20], and grasping [16, 17, 20, 21]. Even tasks demanding long-horizon planning and a range of skills, such as the comprehensive removal of all food items from a table [22], have found solutions by deploying cutting-edge learning tools. Inspired by these advancements, we examine a learning-based method to address 3D deformable object shape control.

Shape servoing historically has been tackled by mostly learning-free approaches [9–11, 23–25]. These methods often represent the object as a set of hand-picked feature points, thus struggling to generalize to unseen objects and being vulnerable to sensor noise. Shetab-Bushehri et al. [25] leverage a 3D lattice to describe deformable objects enabling accurate 3D shape control. Nevertheless, an assumption of feature correspondence limits its application.

Hu et al. [13] use the fast point feature histogram (FPFH) [26] of deformable objects as the state representation for shape control learning. However, Thach et al. [4] showed that the FPFH fails to capture the complex dynamics of 3D deformable objects. DeformerNet [3, 4] is the current state-of-the-art in 3D shape servoing, operating effectively in both simulation and physical-robot experiments. It offers a neural network that takes as input the object’s current and goal point clouds and computes the desired robot action to drive the object toward the target shape. However, a significant weakness of DeformerNet and other shape servoing methods is the requirement to define goal shapes explicitly.

While point cloud generative networks show strong performance [27–30], their application in robotics is somewhat limited. In terms of goal generation for robotics, Waveren et al. [31] introduce a generative model that can render a goal image for a rearrangement task, given natural language instructions. However, this method does not reason about changing deformable object geometries.

Various learning-based methods have been examined for robotic automation of surgical procedures, such as cutting [32, 33], suturing [34, 35], retracting tissues [36, 37], navigating surgical tools [38], and tissue tracking [39, 40].

III. PROBLEM FORMULATION

We address the problem of orchestrating robotic manipulation of a 3D deformable object to achieve a specific task $T$. The term 3D indicates that no single dimension in the object significantly surpasses the other two in scale [1].

We define the 3D object to be manipulated as $O \subset \mathbb{R}^3$. $O$ undergoes dynamic changes during robotic manipulation. Due to the inherent limitations in directly sensing the entirety of $O$, we instead work with a partial-view point cloud $\mathcal{P} \subset O$, encompassing a subset of points on the object’s surface. We represent the current point cloud of the object as $\mathcal{P}_c$.

The robot can reshape the object by grasping it at a pair of manipulation points \{$p_{m_i}$\}_{i=1,2} and subsequently moving its end-effectors. We define robot actions $A$ as a pair of homogeneous transformation matrices: $\mathcal{A} \in SE(3) \times SE(3)$, representing the desired change of end-effector poses. Note that for tasks that do not require bimanual manipulation, we only need a single manipulation point and a single transformation matrix for the action.

We frame our problem as a contextual learning problem wherein there is observable context present at the start of the task execution, which, when combined with the observable state of the deformable object, dictates how the task should be performed. We encode the task-specific context as a contextual point cloud $\mathcal{P}_T$. This point cloud does not belong to the object volume but provides crucial features for the task’s success. It may include task-relevant objects or features in the surrounding environment. The design choice for the contextual point cloud varies from task to task and will be elaborated on in Sec. V. For instance, for a surgical robot tasked with lifting a deformable tissue layer off a
kidney, $\mathcal{P}_T$ could include the partial view of the kidney observable at the start of the task execution.

For a given task $T$, we assume an associated dataset, $\mathcal{D}$, which consists of a set of demonstration trajectories that accomplish the task under various contexts. We define each trajectory $\tau$ as an ordered sequence of $M$ waypoints ($P_1, P_2, \ldots, P_M$), where each $P_i$ is a point cloud observation of the deformable object at the associated time. The problem then becomes to learn a policy to autonomously perform $T$ in new, unseen contexts. To learn this, the robot has the available training data of associated contexts, initial object states, and demonstration trajectories.

IV. Method

Our formulation assumes that a goal shape $\mathcal{O}_g$ exists, whereby if the object reaches this particular configuration, the task succeeds. Leveraging this intuition, we decompose the initial complex problem into two distinct sub-problems.

The first examines contextual goal learning. Here, we aim to predict a goal point cloud $\mathcal{P}_g$ corresponding to a successful task outcome, based on the current state and the context: $\mathcal{P}_g = \Phi(\mathcal{P}_c, \mathcal{P}_T)$. We can train a task-specific goal generation model using the demonstration dataset $\mathcal{D}$ associated with task $T$. At runtime, DefGoalNet reasons over the current point cloud observations and generates a goal point cloud that corresponds to a successful task outcome under that context.

Given this point cloud, we can examine the second question of goal-conditioned shape control. We address this by learning a policy that maps the current point cloud, goal point cloud, and manipulation point to a robot action: $\pi(\mathcal{P}_c, \mathcal{P}_g, \mathcal{P}_m) = A$. By applying this policy repeatedly, the robot gradually transforms the object towards its goal shape, ultimately accomplishing the task. The robot selects manipulation points using the dense predictor network from [3].

Note that DeformerNet [3] presents an effective solution to the second sub-problem. Therefore, the remainder of this section primarily focuses on the contextual goal learning problem. We first provide details of the proposed architecture for DefGoalNet. We then explain how we train the model. We conclude the section by describing how we integrate the goal generation network with DeformerNet.

A. DefGoalNet Architecture Details

We adopt an encoder-decoder architecture for DefGoalNet. Our neural network feeds the inputs $\mathcal{P}_c$ and $\mathcal{P}_T$ into two identical PointConv [41] encoder channels $g$, generating two feature vectors $\psi_c = g(\mathcal{P}_c)$ and $\psi_T = g(\mathcal{P}_T)$. By concatenating them together, we obtain the final feature vector $\psi_1 = \psi_c \bigoplus \psi_T$. This feature vector is fed into a decoder $d$, which consists of a series of fully connected layers and eventually outputs a 1D vector with $3 \times N$ elements, where $N$ is the desired number of points in the goal point cloud. The 1D vector is then reshaped to construct a point cloud of shape $3 \times N$. The composite goal generator thus takes the form: $\mathcal{P}_g = \Phi(\mathcal{P}_c, \mathcal{P}_T) = d(g(\mathcal{P}_c) \bigcup g(\mathcal{P}_T))$.

Prior to training, we employ farthest point sampling [42] to downsample both $\mathcal{P}_c$ and $\mathcal{P}_T$ to $N$ points. In all our experiments, we choose $N$ to be 512.
To effectively capture the complex goal point clouds of deformable objects, we adopt a loss function that combines Chamfer distance and earth mover’s distance, both widely recognized point cloud distance metrics [43, 44]. Chamfer distance measures the average distance of each point in one set to the nearest point in the other set:

\[ C(p_a, p_b) = \sum_{x \in p_a} \min_{y \in p_b} ||x - y||^2 + \sum_{y \in p_b} \min_{x \in p_a} ||x - y||^2. \]

In contrast, earth mover’s distance quantifies the dissimilarity between two point distributions:

\[ E(p_a, p_b) = \min_{\xi: p_a \rightarrow p_b} \sum_{x \in p_a} ||x - \xi(x)||_2, \]

The training loss, a linear combination of these two distances, measures the disparity between predicted and ground-truth point clouds. We train DefGoalNet end-to-end using the Adam solver [45] on a single RTX 3090 GPU.

C. Integration with DeformerNet

DeformerNet provides an effective policy for deformable object manipulation: \( \pi_s(p_c, p_g, p_m) = A \). The shape servoing policy takes as inputs a manipulation point along with the current and goal object point clouds. An issue arises, however, with the reliance on the specification of a goal shape, which is impossible to obtain in some applications.

Herein lies the significance of DefGoalNet, which generates a goal point cloud \( p_g = \Phi(p_c, P_T) \). This predicted goal point cloud becomes the input for DeformerNet. As a result, the robot policy within DeformerNet no longer hinges on the potentially difficult-to-generate \( P_g \), but is instead formulated as a function of practically obtainable parameters, namely \( P_c \) and \( P_T \):

\[ \pi_s(p_c, p_g, p_m) = \pi_s(p_c, \Phi(p_c, P_T), p_m) = A. \]

Running this policy in a closed-loop fashion enables the robot to accomplish the task. Prior to manipulation, given the learned \( P_g \), the dense predictor network of DeformerNet can select the manipulation points on the object that the robot should grasp. Note that we leverage the DeformerNet and dense predictor models trained in the original paper [3] without any fine-tuning. This demonstrates an advantage of our approach, enabling independent training of goal prediction and deformable shape control.

V. EXPERIMENTS AND RESULTS

We assess the performance of our method in simulation and on a real-world robotic setup. In simulation, we employ a model of the patient-side manipulator from the da Vinci research kit surgical robot [46] using the Isaac Gym platform [47]. We test on a Baxter robot equipped with a laparoscopic tool and an Azure Kinect camera for capturing point clouds for our real-world experiments.

We evaluate our method on two simulation-based robotic tasks: surgical retraction and tissue wrapping. We then conduct a zero-shot sim-to-real transfer experiment with a physical robot, using the model trained entirely on simulated data. Finally, we experiment with a real human demonstration dataset on a physical mock surgical retraction task.

A. Demonstration data collection

Here we describe the procedure for collecting training data for DefGoalNet for our experiments. First, we collect \( M \) demonstration trajectories that accomplish the task. Note that in some experiments we utilize scripted demonstrations only to facilitate evaluation of our method in simulation. In practice, however, the demonstrations will be provided by humans (as in our physical robot experiments). We record each trajectory’s initial and terminal object point clouds and save them as current and goal point clouds (\( P_c \) and \( P_g \), respectively) for training. We also record the contextual point cloud \( P_T \). We detail the design choice for \( P_T \) in the following sections after formally introducing each task. Finally, we construct input-output pairs for DefGoalNet training. The input is \((P_c, P_T)\), and the output is \( P_g \).

B. Simulation Experiments

1) Surgical Retraction: Surgical retraction plays a pivotal role in nephrectomies, a procedure during which an adipose tissue layer needs to be retracted off of a kidney [48]. Here, we designed a simulated robotic task to emulate this surgical procedure, as depicted in Figure 3. The robot is assigned to lift the tissue to reveal the kidney beneath it.

To train DefGoalNet on this task, we create a dataset of different kidneys with sizes sampled from the distribution of typical, adult human kidneys [49], along with different box-like tissue layers with geometries sampled from a uniform distribution. We also randomize the kidney and tissue poses. The contextual point cloud \( P_T \) for this task is the partial-view point cloud of the part of the kidney unoccluded by the adipose layer. The demonstration trajectories are generated using scripted robot actions. We set a target plane that bisects...
Fig. 5: Simulated retraction results - Chamfer distance between predicted and ground truth goal point clouds on the test set across multiple training dataset sizes. Example predicted goals are visualized on top of each violin plot.

The kidney into two halves along its longitudinal dimension. The robot demonstrator grasps the tissue and retracts it until the entire tissue layer passes beyond the target plane.

We examine the influence of dataset size on the performance of DefGoalNet by training separate models on 10, 100, 1000, and 10000 demonstrations. The 10000 demonstration set is obtained by running the robot demonstrator on 100 different sampled kidneys, each with 100 different sampled tissue layers, all with randomly sampled poses. The smaller sets are selected uniformly at random from the largest set. We train the 10 demonstration model 5 times with 5 different sets of demonstrations to examine performance variance at small data scales. We evaluate our method on a test set comprising 100 unseen configurations (kidney size and pose, and tissue size and pose). A representative manipulation sequence is visualized in Fig. 3. Two evaluation metrics quantify the performance of our method.

The first and most important metric directly measures the success rate. On each test configuration, we run the DeformerNet policy on the predicted goal generated by DefGoalNet and record the final object point cloud. We leverage the robot demonstrator’s target plane to assess how good each retraction trajectory is. We count the percentage of points in the final point cloud that successfully pass through the target plane, which we call success percentage. As visualized in Fig. 4, even with as few as 10 demonstrations, our method can still achieve a median success percentage of nearly 90%.

The second metric is the Chamfer distance between the predicted and ground truth goal point clouds. The Chamfer results and example predicted goals are visualized in Fig. 5. Unsurprisingly, as we increase the demonstration dataset size, the generated goals become more realistic and easily interpretable; however, the structure of the goal is visible even from 10 demonstrations.

2) Tissue Wrapping: To showcase the breadth of deformable manipulation tasks DefGoalNet can handle, we conduct further experiments on a tissue wrapping task inspired by surgical procedures such as aortic stent placement. This entails the cooperation of two robotic arms to encase a thin layer of tissue around a cylindrical tube, with the goal of maximizing tissue coverage on the tube’s surface. The contextual point cloud \( P_T \) for this task is the partial-view point cloud of the cylindrical tube.

We train DefGoalNet on a varying number of demonstrations: 10, 100, and 1000. For the case of 10 demonstrations, we also train 5 times with different random seeds to validate the robustness of the approach. A representative manipulation sequence is visualized in Fig. 6.

We compute the tissue coverage percentage to quantify task success, developed in [3]. This measures the percentage of tube surface area being wrapped by the tissue. We run DeformerNet on the test set with goals generated by DefGoalNet and visualize the results in Fig. 7. At a dataset size of 100, our method starts achieving competitive performance with a median coverage percentage of almost 90%.

Fig. 8 visualizes the Chamfer distance between predicted and ground truth goal point clouds on a test set of 100 unseen demonstrations. As with retraction, there is a clear trend that more data makes the predicted goals more similar to the ground truth shapes.

Fig. 6: Sample manipulation sequence on the tissue wrapping task, in simulation. The red point cloud visualizes the goal shape generated by DefGoalNet.

Fig. 7: Simulated tissue wrapping - Tissue coverage percentage.

C. Physical Robot Experiments

1) Zero-Shot Sim-to-Real Transfer: We first evaluate whether the learned goal point clouds trained in simulation can be directly utilized to perform retraction tasks with a physical robot. We employ the model trained with 100 demonstrations in Sec. V-B.1 for this experiment. We use a
We train the model with 15 demonstrations, holding 5 demonstrations for testing purposes. The Chamfer distances between the predicted and ground truth goal point clouds on this test set are 1.72, 0.97, 1.64, 0.45, and 0.64 millimeters. Fig. 11 visualizes the two worst examples of the predicted goal point clouds on the test set. The goals look similar to the ground truth, showcasing DefGoalNet’s ability to generalize even when trained on a relatively small dataset.

We evaluate the robot using three distinct goals, each predicted from distinct contextual point clouds of the human’s hand pose. We execute 5 trials for each scenario to ensure robustness for a total of 15 runs. Fig. 1 visualizes a representative retraction result. Qualitatively, our method predicts goal shapes that align well with the human demonstrator’s objectives and computes the necessary robot actions to drive the object to these goals.

VI. CONCLUSIONS

We have presented a novel method for learning deformable object manipulation from demonstrations. At the heart of this pipeline is DefGoalNet, a neural network that predicts the desired object shape to successfully execute a given task. We demonstrate DefGoalNet’s effectiveness on a diverse set of tasks, achieving a 100% success rate on a zero-shot sim-to-real task. Crucially, we show how contextual goal generation can be learned from relatively few demonstrations while still leveraging a control policy learned on a large and diverse dataset independent of the specific downstream task.

DefGoalNet currently assumes that the final state of the object solely defines success. In many robotic applications, the overall trajectory is a critical consideration of task success. To address this limitation, future research could explore methods for predicting a sequence of goal shapes instead of just a single goal instance. In addition, DefGoalNet is currently deterministic, where only one goal is output to achieve the task given a specific current shape and context. We will investigate a version of DefGoalNet that generates a distribution over goal shapes. Furthermore, while DefGoalNet is inherently a task-specific model, it holds the potential to learn across multiple tasks. Lastly, we will aim to employ the method in other domains such as in home and warehouse environments.
REFERENCES


