VIRDO++: Real-World, Visuo-tactile Dynamics and Perception of Deformable Objects

Anonymous Author(s)
Affiliation
Address
email

Abstract: Deformable objects manipulation can benefit from representations that seamlessly integrate vision and touch while handling occlusions. In this work, we present a novel approach for, and real-world demonstration of, multimodal visuo-tactile state-estimation and dynamics prediction for deformable objects. Our approach, VIRDO++, builds on recent progress in multimodal neural implicit representations for deformable object state-estimation [1] via a new formulation for deformation dynamics and a complementary state-estimation algorithm that (i) maintains a belief over deformations, and (ii) enables practical real-world application by removing the need for privileged contact information. In the context of two real-world robotic tasks, we show: (i) high-fidelity cross-modal state-estimation and prediction of deformable objects from partial visuo-tactile feedback, and (ii) generalization to unseen objects and contact formations.

Keywords: Deformable Object Manipulation, Multimodal Representation Learning

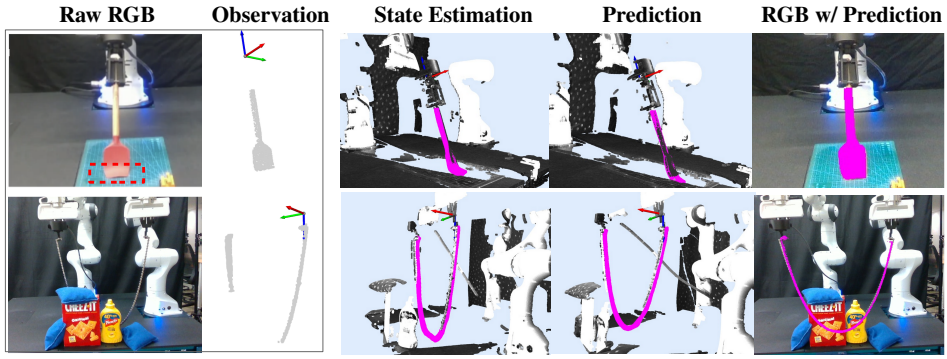

Fig. 1: Given partial point cloud observations (left), and multimodal sensor measurements (end effector wrist reaction wrench and pose), VIRDO++ can accurately predict the 3D geometry (magenta) and reaction forces of deformable objects such as spatulas (top) and bike chains (bottom) conditioned on the next robot action under severe occlusions – which can be artificial (red dotted box) or natural due to obstacles (e.g., from other objects).

1 Introduction

Deformable objects are ubiquitous in our everyday lives – the clothes we wear, the food we eat, and many of the tools we use are just a few examples. As such, helpful robots of the future may benefit from the ability to master dexterous manipulation of deformable objects. At the heart of this mastery is the interplay between object geometry and force transmission, perceived by vision and touch. Deformable object manipulation is difficult due to the complexity of this interplay (e.g., infinite dimensional state-spaces and nonlinearity) and the ensuing challenges in perception (e.g., partial observability and occlusion) and controls [2, 3, 4, 5]. Multimodal visuo-tactile representations can help address many of these challenges by exploiting mutual information and complementary cues.

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

Existing deformable object manipulation approaches typically use one modality (mostly vision) and rely on finite element/particle-based techniques [6, 7, 8, 9, 10, 11, 12, 13] or leverage deep learning for visual affordance/latent dynamics learning [14, 15, 16, 17, 18, 19]. The former methods typically rely on privileged knowledge (e.g., occluded or unknown boundary conditions) and stop at system identification, limiting their ability to refine the underlying physics model by learning from data. Latter methods lack i) tactile feedback which is far more informative of contact than vision, and ii) structured representations which can limit generalization and increase sample-complexity.

Here, we build on recent work [1] that provides a framework for visuo-tactile state estimation using multimodal implicit neural representations. This approach offers a number of advantages including multimodal sensory fusion, direct integration with raw sensory feedback, and computational speed. However, [1] does not address 1) deformation predictions (dynamics) and 2) is not directly applicable to real-world robotics settings. The first limitation is due to the lack of a dynamics module and an emphasis on static scenes. The second limitation is due to the need for the contact patch which are typically impractical to measure in real-world robotic settings.

The primary contribution of this work is (i) a novel approach to, and (ii) real-world demonstration of, performing multi-modal visuo-tactile state estimation and dynamics prediction for deformable objects. Fig. 1 shows how our proposed approach uses partial views and tactile feedback for deformation prediction and state-estimation in real-world applications. Our contributions are:

1. Representation of deformable objects *dynamics* conditioned on actions: Multimodal neural implicit representation with action-conditioned dynamics to predict future deformations. To demonstrate the utility of the dynamics model, we introduce a state estimation algorithm based on particle filtering to maintain a belief distribution over object deformations.

2. Real-world demonstrations on two challenging tasks: Scraping with a spatula and chain manipulation. We demonstrate both 3D geometry estimation in heavy occlusion and action-conditioned deformation/reaction wrench prediction. This is enabled by introducing a novel contact embedding that eliminates the need for explicit contact patch information used in [1]. We use these tasks to demonstrate **generalization to unseen objects and novel environments**.

2 Methodology

Our approach, shown schematically in Fig. 2, is composed of a latent deformation dynamics model and an implicit dense geometric representation. Using Hidden Markov Models as an analogy, the former component plays the role of the hidden state transition, and the latter is the observation model. We expand on the latent deformation dynamics in Sec. 2.1 and discuss the implicit geometric representation in Sec. 2.2.

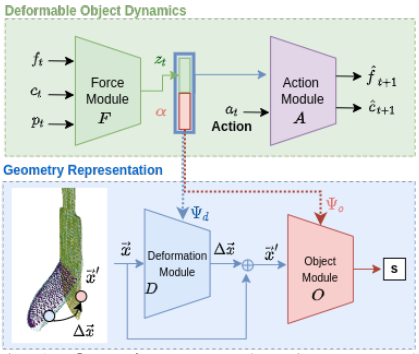

Fig. 2: **Overview.** VIRDO++ is composed of an implicit signed-distance field representation of geometry (bottom), informed by a deformation dynamics model (top), where dotted lines indicate hyper networks that decode embeddings into network weights.

2.1 Deformable Object Dynamics

VIRDO++ represents deformable object states using latent object ($\boldsymbol{\alpha}$) and force ($\boldsymbol{z_t}$) codes. The object code condenses the undeformed shape information into a feature representation that can extrapolate to new unseen object variants. The force code is generated by the Force Module (F) which encodes boundary conditions as $\boldsymbol{z}_t = \mathbf{F}(\boldsymbol{f}_t, \boldsymbol{c}_t, \boldsymbol{p}_t)$ where $\boldsymbol{f}_t \in \mathbb{R}^6$ is the reaction wrench at the wrist, $\boldsymbol{p}_t \in \mathbb{R}^6$ is robot end-effector pose, and $\boldsymbol{c}_t \in \mathbb{R}^{l_c}$ is the contact embedding.

The contact embedding $\boldsymbol{c}_t$ is a learnable latent vector that can change over time, whose primary role is to store information about contact. However, it can also implicitly integrate information about the object's physical properties (e.g., stiffness) and historical information about the object's states and deformations. This embedding generalizes the original VIRDO [1] in terms of both (i) no longer needing known contact patch locations, unlocking the capability to run VIRDO in the real-world, and (ii) representing additional useful information beyond just the contact location as was the case

in VIRDO. In more detail, VIRDO [1] used ground truth contact patches encoded by a PointNet encoder [20] directly which required access to this privileged information. Here, the contact patch and encoder are replaced by the learnable contact embedding.

The Action Module ($\mathbf{A}$) predicts the next step's boundary conditions given the current (latent) state and robot action: $\mathbf{A}(\boldsymbol{\alpha}, \boldsymbol{z_t}, \boldsymbol{a_t}) = \hat{\boldsymbol{f}}_{t+1}, \hat{\boldsymbol{c}}_{t+1}$ where $\hat{\boldsymbol{f}}_{t+1} \in \mathbb{R}^6$ is the predicted reaction wrench at the wrist, $\hat{\boldsymbol{c}}_{t+1}$ is the predicted contact embedding, and $\boldsymbol{a}_t \in \mathbb{R}^6$ is the robot action (Cartesian displacement). This predictive capability enables state-estimation and planning for deformable object manipulation. VIRDO lacked this functionality and could only infer the object geometry statically.

2.2 Geometry Representation

VIRDO++ decouples the 3D geometric representation of deformable objects into an undeformed (nominal) shape signed-distance field (SDF) and a set of deformation fields, similar to [1, 21]. The deformation field is a 3D vector field that, when summed with the deformed object SDF, results in the nominal shape SDF. More precisely, any query point $\boldsymbol{x} \in \mathbb{R}^3$ in the robot wrist frame belonging to the deformed shape SDF can be mapped back to the corresponding point $\boldsymbol{x}'$ of the nominal shape SDF by applying the point-wise deformation field $\Delta\boldsymbol{x}$ as $\boldsymbol{x} + \Delta\boldsymbol{x} = \boldsymbol{x}'$. In more detail:

Nominal Shape Representation: The Object Module $\mathbf{O}(\boldsymbol{x}')$ is the parametric representation of the nominal shape SDF – the object shape with no external contacts corresponding to $\Delta\boldsymbol{x} = 0$. Here, we use a feedforward neural network to parameterize the SDF [22] and write $\mathbf{O}(\boldsymbol{x}'|\boldsymbol{\Psi}_o(\boldsymbol{\alpha})) = s$ where s is the signed-distance at query point $\boldsymbol{x}'$ and $\boldsymbol{\Psi}_o(\boldsymbol{\alpha})$ is a hyper-network that predicts the weights of $\mathbf{O}$ conditioned on the object code $\boldsymbol{\alpha}$. The object code and hyper-network weights are learned end-to-end in an auto-decoder approach [22]. We simplify our notation to $\mathbf{O}(\boldsymbol{x}')$ for the remainder.

Deformation Field Representation: The generalized geometric representation of a deformable object is given by $\text{SDF}(\boldsymbol{x}) = \mathbf{O}_{\boldsymbol{\Phi}_o}\big(\boldsymbol{x} + \mathbf{D}(\boldsymbol{x}|\Psi_d(\boldsymbol{z}_t, \boldsymbol{\alpha}_t))\big) = s$. Here, the Deformation Module $\mathbf{D}(\boldsymbol{x}|\Psi_d(\boldsymbol{z}_t, \boldsymbol{\alpha}_t)) = \Delta\boldsymbol{x}$ produces an object's deformation field given force and object code pairs. Similar to the Object Module, Deformation Module's weights are given by the hyper-network Ψ_d.

2.3 Training and Loss Formulation

The training dataset for VIRDO++ is a set of trajectories $\mathcal{S} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_N\}$, where each trajectory $\mathcal{T}_j = \{(\boldsymbol{p}_0, \boldsymbol{f}_0, \boldsymbol{P}_0, \boldsymbol{a}_0), (\boldsymbol{p}_1, \boldsymbol{f}_1, \boldsymbol{P}_1, \boldsymbol{a}_1), ..., (\boldsymbol{p}_k, \boldsymbol{f}_k, \boldsymbol{P}_k, \boldsymbol{a}_k)\}$ has k sequential observation and action tuples. Here $\boldsymbol{P}_j$ is a partial point cloud. Our first step is to pretrain a nominal shape representation (i.e., hyper-network $\boldsymbol{\Psi_o}$ and object code $\boldsymbol{\alpha}$). The loss for training m nominal shapes is $\mathcal{L}_{nominal} = \mathcal{L}_{sdf} + \lambda_1\mathcal{L}_{latent} + \lambda_2\mathcal{L}_{hyper}$ as in [1], where $\mathcal{L}_{latent} = \sum_{i=1}^{m} \|\boldsymbol{\alpha}^i\|_2$, $\mathcal{L}_{hyper} = \sum_{i=1}^{m} \|\boldsymbol{\Psi}_o(\boldsymbol{\alpha}^i)\|_2$, and:

$$\mathcal{L}_{sdf} = \sum_{i=1}^{m} \Big(\sum_{\bar{\boldsymbol{x}} \in \boldsymbol{\Omega}} |clamp(\mathbf{O}^i(\boldsymbol{x}), \delta) - clamp(s^*, \delta)| + \lambda \sum_{\bar{\boldsymbol{x}} \in \boldsymbol{\Omega}_0} (1 - \langle \nabla \mathbf{O}^i(\boldsymbol{x}), \boldsymbol{n}^* \rangle) \Big).$$

$\boldsymbol{n}^*$ is the ground truth normal vector, δ is a parameter that clips SDF predictions, and s^* is the ground truth signed distance. After the pretraining, we train the rest of the modules using the three losses $\mathcal{L}^{geo}$, $\mathcal{L}^{pred}$ and $\mathcal{L}^{reg}$ summed over a fixed time horizon (w) while freezing Ψ_o and the object code α:

$$\mathcal{L}_{tot} = \sum_{t=t_0}^{t_0+w} (\mathcal{L}_t^{geo} + \lambda_3\mathcal{L}_t^{pred} + \lambda_4\mathcal{L}_t^{reg}) \tag{1}$$

The geometry representation loss is defined as:

$$\begin{aligned}\mathcal{L}^{geo} = \lambda_5 \underbrace{\sum_{\boldsymbol{x} \in \boldsymbol{\Omega}} \|\mathbf{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{x})\|_2}_{\text{minimum correction}} + \lambda_6 \underbrace{\text{CD}(\boldsymbol{P} + \mathbf{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{P}), \bar{\boldsymbol{P}}^*)}_{\text{correspondence}} + \lambda_7 \underbrace{\sum_{\boldsymbol{x} \in \boldsymbol{\Omega}_0} (1 - \langle \nabla_{\boldsymbol{x}} \mathbf{O}_{\Psi_o}(\boldsymbol{x}'), \boldsymbol{n}^* \rangle}_{\text{normal aligning}} \\ + \lambda_8 \underbrace{\sum_{\boldsymbol{x} \in \boldsymbol{\Omega}} |clip(\mathbf{O}_{\boldsymbol{\Psi}_o}\big(\boldsymbol{x} + \mathbf{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{x}), \delta\big) - clip(s^*, \delta)|}_{\text{signed distance regression}}.\end{aligned} \tag{2}$$

where $\boldsymbol{x}' = \boldsymbol{x} + \mathbf{D}_{\Psi_d}(\boldsymbol{x})$, Ω is the 3D querying space, $\Omega_0 \subset \Omega$ is the on-surface region, $\mathbf{P} := \{\boldsymbol{p} | \boldsymbol{p} \in \Omega_0\}$ is an unordered set of the on-surface points, and $\bar{\boldsymbol{P}}^*$ is the ground truth nominal shape point cloud. Next, we add a reaction wrench prediction loss as $\mathcal{L}_{pred} = \|\boldsymbol{f}_{t+1} - \hat{\boldsymbol{f}}_{t+1}\|$. The prediction loss for $\hat{\boldsymbol{c}}_{t+1}$ is implicitly handled by Eq. 2 because the contact embedding prediction is recursively used as next step's input during training. In contrast, we do replace $\hat{\boldsymbol{f}}_{t+1}$ with real measurement in Eq. 2. Finally, regularization losses $\mathcal{L}_t^{reg} = \lambda_9 \|\boldsymbol{z}_t\|_2 + \lambda_{10} \|\boldsymbol{c}_t\|_2 + \lambda_{11} \|\boldsymbol{\Psi}_d(\boldsymbol{\alpha}, \boldsymbol{z}_t)\|_2$, which regularize the force codes, contact embedding, and the weights of $\mathbf{D}$ respectively.

2.4 Inference and State-estimation Algorithm

In order to maintain a belief distribution over the object deformation, we develop a particle filter-based inference algorithm, Alg. 1. Each particle represents a contact embedding which can be used to reconstruct the full 3D object geometry. The `Refine` step updates the particle set by performing gradient descent on $\min_{\hat{\mathbf{C}}_t} \mathcal{L}^{geo}(\boldsymbol{P}_t, \hat{\mathbf{C}}_t, \boldsymbol{f}_t, \boldsymbol{p}_t)$ with the visible object point cloud. The particles are then weighted according to their wrench error prediction with `Weight-Function` defined as $\boldsymbol{w}_t^i = exp(-\gamma(\hat{\boldsymbol{f}}_t^i - \boldsymbol{f}_t))$ and resampled with probability proportional to this weight vector. Here, $\hat{\boldsymbol{f}}_t^i$ is the reaction wrench prediction of the ith particle and $\boldsymbol{f}_t$ is a ground truth measurement (see **Appendix**). Finally, the particles are propagated through the Force and Action Modules, a small amount of uncertainties is applied $\sim \mathcal{N}(0, 0.01)$ to the particles and the algorithm is repeated.

Algorithm 1 Particle Filter based Inference Algorithm

procedure
  $\hat{\mathbf{C}}_0 \leftarrow \{\hat{\boldsymbol{c}}_0^0, \hat{\boldsymbol{c}}_0^1, ..., \hat{\boldsymbol{c}}_0^{n-1}\} \sim \mathcal{N}(0, 0.01)$ ▷ Init Particles
  for each $scan$ in $Trajectory\ Length$ **do**
    $\mathbf{C}_t \leftarrow$ `Refine`$(\mathbf{P}_t, \hat{\mathbf{C}}_t, \boldsymbol{f}_t, \boldsymbol{p}_t)$
    if $scan$ is 0 **then**
      $\bar{\mathbf{C}}_\mathbf{t} \leftarrow \mathbf{C}_t$
    else
      $\boldsymbol{w}_t \leftarrow$ `Weight-Function`$(\boldsymbol{f}_t, \hat{\boldsymbol{f}}_t^0, \hat{\boldsymbol{f}}_t^1, ..., \hat{\boldsymbol{f}}_t^{n-1})$
      $\bar{\mathbf{C}}_\mathbf{t} \leftarrow$ `Re-sampling`$(\boldsymbol{w}_t, \mathbf{C}_t)$
    end if
    $\mathbf{Z}_\mathbf{t} \leftarrow$ `Force-Module`$(\bar{\mathbf{C}}_t, \mathbf{f}_t, \mathbf{p}_t, \alpha)$
    $\{(\hat{\boldsymbol{f}}_{t+1}^0, \hat{\boldsymbol{c}}_{t+1}^0), (\hat{\boldsymbol{f}}_{t+1}^1, \hat{\boldsymbol{c}}_{t+1}^1), ..., (\hat{\mathbf{f}}_{t+1}^{n-1}, \hat{\boldsymbol{c}}_{t+1}^{n-1})\} \leftarrow$ Action-Module $(\mathbf{Z}_\mathbf{t}, \boldsymbol{\alpha}, \boldsymbol{a}_t)$
    $t \leftarrow t + 1$
  end for
end procedure

3 Experiments

For real robot experiments, we investigate two classes of objects: **spatulas** and **bike chains**. These objects exhibit different deformation behaviors: (i) spatulas are representative of bendable elastic objects that retain their shape when external forces are removed, whereas (ii) bike chains (with the 2 end points controlled by a bi-manual manipulator) can undergo significantly larger deformations (e.g., from gravity) – requiring algorithms that can reason over free-space while estimating in-contact shapes and forces. Spatula deformations are generally local and experience significant reaction wrenches, while chain deformations are more global but with much smaller reaction wrenches.

3.1 Spatula Manipulation

Dataset. Train and test objects are shown in Fig. 3. The training set consists of 4 spatulas, each with 672 samples ($N = 28$, $k = 24$), using 9:1 train/test split (25 train and 3 test trajectories) for representation learning. The test dataset consists of 2 held-out spatulas, each with 72 samples ($N = 3$, $k = 24$), and is used to evaluate zero-shot transfer and not used for representation learning. At the beginning of each trajectory ($k = 0$), the end-effector moves to a random world Cartesian position in a bounding box of size $(0.1 \times 0.24 \times 0.06)$ meters and the tool is brought into contact with

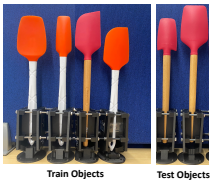

Fig. 3: Train and test objects differ in terms of (i) geometry, and (ii) how they react under wrench.

the table. Within a trajectory, the tool maintains contact while sampling a random action from $(x, y, z, r, p, y) = \pm(0.02, 0.02, 0.01, 0.05, 0.04)$ where translations are in meters and rotations in radians. For the hardware, we use Franka Emika's Panda robot, ATI Gamma F/T sensor mounted on the robot's wrist and the Photoneo PhoXi 3D Scanner L. Wrench measurements are averaged over 0.05 [s] and the point cloud is registered to the wrist frame. $\boldsymbol{P}$ is then the result of segmenting the object pointcloud and normalizing the scale into a bounding box of size $2 \times 2 \times 2[m]$ centered around $(0, 0, 0)$. The test dataset point clouds $\boldsymbol{P}$ contain occlusions at the bottom $0.15 \times$ `spatula-height` as shown in the top observation panel in Fig. 1.

3.1.1 State Estimation and Predictions

Here, we evaluate the deformation prediction (Sec. 2.1) and state-estimation (Alg. 1) accuracy of VIRDO++ on training and test trajectories. Fig. 4 shows an example of state-estimation for a test trajectory where the ground truth (black) and reconstruction (magenta) are overlaid. We note the high accuracy of state-estimation indicated by the overlay agreement, where the average distance between the measured points and reconstructions is within $0.03[m]$ of normalized scale, just 1% of the object length. The quantitative results, reported in Tab. 1 and Fig. 5, show that VIRDO++ is able to faithfully predict and estimate the object deformation and reaction force despite significant occlusions. Force and torque norm errors are less than 0.567 [N], 0.106 [Nm] (train) and 1.130 [N], 0.135 [Nm] (test) respectively (see more details in **Appendix**).

CD (x1e3)	Spatula w/ table	Bike Chain
Train Est.	0.777 (0.215)	0.708 (0.331)
Train Pred.	0.830 (0.330)	0.762 (0.224)
Test Est.	0.934 (0.210)	0.760 (0.234)
Test Pred.	1.041 (0.348)	1.247 (0.391)

Table. 1: VIRDO++ can accurately estimate object geometry deformations under significant occlusions, measured in terms of mean Chamfer Distances (CD) (std.) $[m^2]$ across 5600 on-surface points between predicted and ground truth reconstructions.

Our ablation study, Tab. 2(b), helps further anchor the scale of these state-estimation and dynamics prediction errors. On state estimation, for example, our full VIRDO++ method achieves a Chamfer distance ($\times 1e3$) of normalized geometry is 0.93 $[m^2]$. The baseline model without a novel contact embedding input has a 12% increased Chamfer distance as 1.06 $[m^2]$. Further if we assume the object is rigid and reduce our state estimation to a "rigid pose estimation" pipeline, the Chamfer distance ($\times 1e3$) increases considerably to 16.19 $[m^2]$. These ablations directly measure the quantitative value of VIRDO++'s contact embedding and deformation estimation and prediction.

We note that the geometry predictions are slightly less accurate than state-estimation. This is because the geometry prediction are computed by the learned dynamics using previous observation and action, while state-estimation uses the current measurement to update the prediction. Despite this, the prediction Chamfer distances are within 14% of the state estimation results. Moreover, the train and test trajectories results show 25% difference in Chamfer distance. This is interesting because the test trajectories are not only from unseen tool configurations but also with occlusions (see **Dataset**). This suggests that learned dynamics of VIRDO++ generalize effectively to unseen contact formations with occlusions.

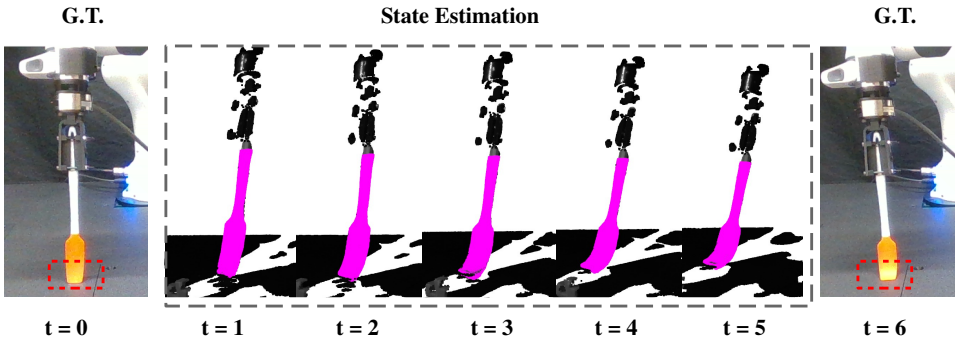

Fig. 4: Even under occlusions (generated by cropping out data from the within the red dotted box), our approach can accurately infer the how the geometry of a test spatula deforms over time (t = 0 to 6), indicated by high overlap agreement between the prediction (magenta) overlaid on the ground truth (black).

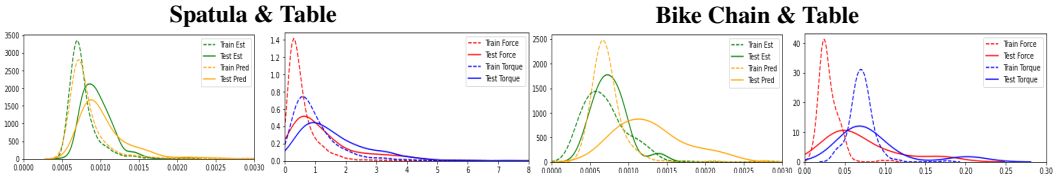

Fig. 5: **KDE plots** of train and test results for spatula and bike chain applications show that VIRDO++ is within small margins of error as measured by (a) Chamfer distance (x1e3) from geometry predictions and state-estimation, and (b) l_2 norm of force prediction error calculated as $\|\boldsymbol{f}_t[:3]\|$ [N], and torque prediction error (x10) calculated as $10 * \|\boldsymbol{f}_t[3:]\|$ [Nm].

3.2 Generalizations To Unseen Objects

This section demonstrates VIRDO++ generalization ability to unseen objects and their dynamics. For each unseen object, the generalization dataset contains 1 nominal point cloud and 3 trajectories, each with 24 transitions with occlusion. Here, we directly apply the same pre-trained model from Sec. 3.1 to the generalization test dataset.

3.2.1 Object code inference

VIRDO++ is able to represent nominal geometries unseen during training within distribution. Fig. 6 shows reconstructions of both seen and unseen object in Fig. 3 using a single pretrained Object Module. For the unseen objects, we start from a randomly initialized object code $\alpha \sim \mathcal{N}(0, 0.01)$ and update the code by gradient descent with Adam. We use the same loss function from Eq. 1 with pre-trained VIRDO++ while freezing the weights of the network. The first row of Tab. 2a shows the Chamfer distance error of the unseen objects alongside training objects, indicating the relatively small drop in performance. When comparing Fig. 6 and the ground truth in Fig. 3, we note that the test object reconstructions are close to the ground truth aside from some fine details at the edges.

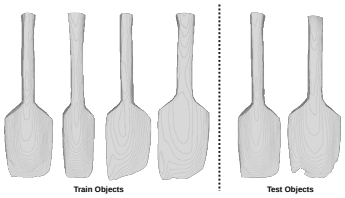

Fig. 6: VIRDO++ can reconstruct train and test objects with high fidelity (same order as Fig. 3).

3.2.2 Zero Shot Dynamics Prediction and State Estimation

With the object codes obtained in Sec. 3.2.1, VIRDO++ is able to predict the unseen object's dynamics based on their nominal shape. Here, we evaluate both prediction and state-estimation performance as in Sec. 3.1.1. We note that the observations of the unseen objects include occlusions as discussed in **Dataset**. We present the results of this evaluation in Tab. 2a and use Fig. 7 to provide intuition for Chamfer distance values. The results show comparable performance between training and test objects with a small drop in accuracy, highlighting our model's generalization capability.

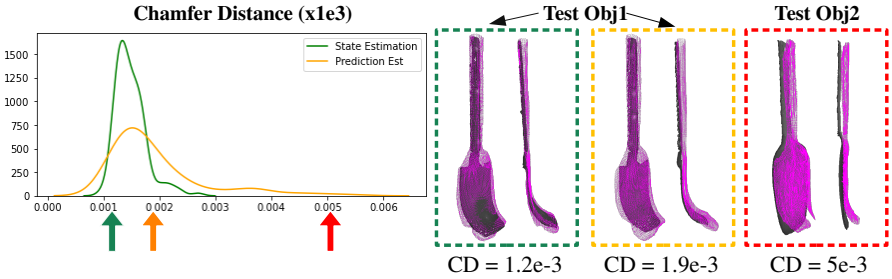

Fig. 7: **Generalizing to unseen object dynamics**. Left panel: KDE plot of Chamfer distances (x1e3) $[m^2]$ using normalized geometries. We show both test objects state estimation using partial point clouds and geometry predictions. Similar to training objects, state estimation has smaller mean and standard deviation of the error. Right panel: We visualized three representative generalization results and corresponding Chamfer distances.

3.3 Wok Scraping Application

In this section, we show VIRDO++ generalization to novel environments – a wok with high curvatures producing greater variances in contact formations. Here, we add a training dataset with 2 scraping trajectories on the wok and perform state estimation and predictions given a test trajectory with 13 transitions. As in Fig. 8, VIRDO++ provides high-fidelity dense geometry of deformable

CD (x1e3)	Train Obj.	Test Obj. 1	Test Obj. 2
Geo. – Nom.	0.65 (0.08)	1.08	1.29
Geo. – S.E.	0.93 (0.21)	1.39 (0.20)	1.56 (0.29)
Geo. – Pred.	1.04 (0.35)	1.54 (0.45)	2.40 (1.08)
Wr. – Pred.	1.14 (1.05)	1.50 (1.00)	2.14 (1.75)

(a) **Train and Test Objects**:

CD (x1e3)	w/o $\boldsymbol{c}_t$	Rigid Asm.
Geo. – S.E.	1.06 (0.25)	16.19 (11.13)
Geo. – Pred.	1.15 (0.36)	16.49 (11.18)

(b) **Ablation:** comparisons to baselines "w/o $\boldsymbol{c}_t$"= without contact embedding as force module input. Rigid asm. = rigid model assumption.

Table. 2: **State Estimation and Dynamics Prediction using *Test* Dataset:** First column of (a) is a result of unseen trajectories for training object. Next two are generalization results to unseen objects. Here, a single pretrained VIRDO++ for all tasks in (a). (b) Ablation study using the same dataset as "(a) Train Obj". Numbers indicate mean(std.) of each error distribution. Chamfer distance(x1e3) uses 5600 points. Geo. = Geometry, Nom. = Nominal, S.E. = State Estimation, Pred. = Prediction, Wr = Wrench calculated as $\|\hat{\boldsymbol{f}}_t - \boldsymbol{f}_t\|$.

objects in high occlusion cases without a 3D model or configuration of the environment; however, due to the complex environment, it occasionally produces undesired artifacts (2 out of 13 trans.).

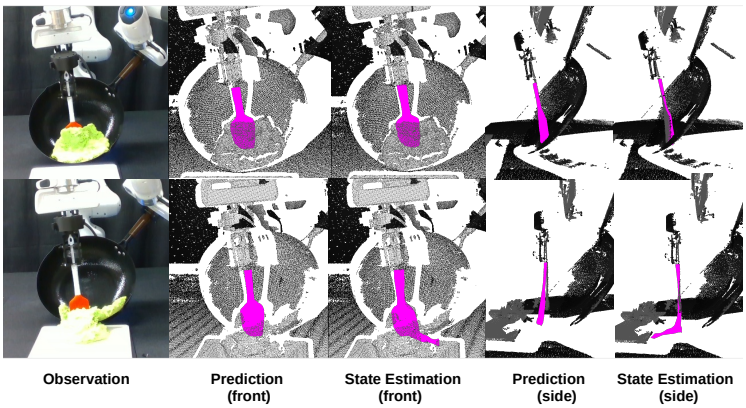

Fig. 8: VIRDO++ can reconstruct high-fidelty dense geometry of deformable objects under occlusions in new environments, such as scraping objects off a wok (successes, top row), but can also at times produce undesirable artifacts (failure modes shown in bottom row).

3.4 Bike Chain Application

In this experiment, the chain ends are grasped by two arms. One end of the chain is moved randomly through the vertices of an equally-spaced $6 \times 6 \times 6$ grid fitting inside of a $0.1 \times 0.18 \times 0.2$ meter bounding box. The visited vertices of this grid make up the training dataset. The test dataset is generated by sampling 20 random positions uniformly from the bounding box. Unlike spatula manipulation, the end-effector orientations were fixed during bike chain manipulation because geometry changes are dominantly caused by translation rather than rotation; therefore, $\boldsymbol{p}_t, \boldsymbol{a}_t \in \mathbb{R}^3$.

As shown in Fig. 9, VIRDO++ is able to predict/estimate both free-space and in-contact chain geometry. The in-contact phase occurs when the arm moves sufficiently far down such that the chain lays on the table. We report the quantitative results of geometry estimation and dynamics prediction in Tab. 1 and Fig. 5. The qualitative results are shown in Fig. 9. The empirical results are comparable

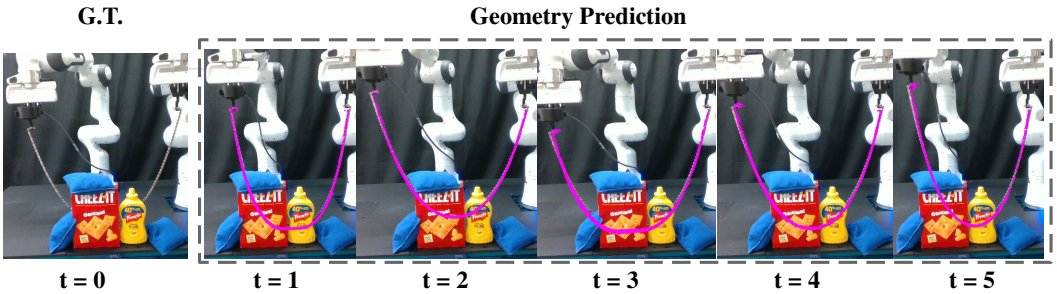

Fig. 9: VIRDO++ can accurately estimate the bike chain geometry (predictions overlaid in magenta) under occlusions due to objects at the front. Note that the chain intermittently makes contact and rests on the table, which is correctly predicted by VIRDO++ despite significant occlusion of where the chain makes contact.

to the spatula experiments despite the significantly larger shape variations. Details on wrench prediction errors are in the **Appendix**, where the standard deviation in error is approximately an order of magnitude smaller than the spatula. We hypothesize that this because i) the ground truth wrench scale is an order of magnitude smaller than that of the spatula and ii) the action space dimension is half the one of the spatula resulting in a simpler model to learn.

4 Related Work

Deformable object modeling. Recent studies have tried to address challenges in system identification and high computation costs in conventional continuum mechanics [6, 7, 8, 9, 10, 11, 12, 13]. To ease the burden of system identification, prior studies infer the physics models' parameters based on high-fidelity physics engines [23, 24, 25] and simple force-deformation relationship (e.g. Hooke's law) [26, 27]. The inherent limitations of these approaches are that i) performance is confined to the underlying physics model, often with strong assumptions on the objects (uniform density/elasticity) and the force-deformation relationship (linearity), and ii) access to privileged information such as occluded or unknown boundary conditions. More recent approaches propose computationally efficient elastic object modelings with minimum system identification using, for example, a potential energy propagation [28] and geometric motion estimates [29]. However, these methods suffer from balancing between model approximation and computation cost as well as maintaining seamless integration with robotic sensing modalities (RGB-D cameras and F/T sensors). In this study, we adopt a data-driven approach using neural networks [1, 30, 31]. Learning directly from observation, our approach does not require object parameters (Young's modulus, Poisson's ratio) or a strong assumption of object composition. Even so, it can capture complex non-linear elastic behaviors and exhibit fast computation times due to the architecture of the neural networks.

3D geometric representation of deformable objects. Most recent studies have adopted discrete geometry representation for deformable objects with finite resolution such as meshes [7, 10, 32] and key points [15, 24], sometimes accompanied with graph neural networks [33, 31]. These approaches have been used largely for 1D and 2D objects (rope, cloth, top view of cylindrical objects) with minimum occlusion. On the other hand, we address general 3D geometry representations, not confined to specific types of objects, which can work with large occlusions (e.g., wok-scraping). Our work builds on dense implicit geometric representation which include signed distances fields [1, 30, 34], occupancy [35], volume density [36], and dense descriptors [37]. These approaches have several advantages over their discrete counterparts in dealing with large occlusion, high resolution surface representation [38], parameterization of 3D geometries [22, 21], and are useful for downstream tasks (e.g., contact location detection similar to Sec. 3.4). We extend these representations to real-world multimodal deformable object dynamics and perception by integrating touch and robot actions.

5 Discussion and Limitations

VIRDO++ provides a novel approach to multimodal deformation prediction using deformation fields conditioned on action, contact, and tactile/force feedback. Our proposed approach to combining an explicit multi-modal deformation field with an implicit geometric representation can have broader application to other implicit approaches (e.g. Neural Radiance Fields and Dense Descriptors). While our results with VIRDO++ are promising, its limitations give rise to future work:

1) Non-rigid grasps: VIRDO++ assumes that the gripper is rigidly attached to a fixed set of points on the object. When it comes to non-rigid grasping, we need to estimate relative motion of the objects w.r.t the gripper and the force transmission is now a function of this relative motion. One potential solution is to use collocated tactile sensor feedback [39, 40, 41] instead of simple F/T sensing, which enables reasoning over reaction forces conditioned on the grasp. **2) Inertial effects:** our approach assumes quasi-static manipulation (negligible accelerations). In terms of fast motion, the tools' inertial effects may also impact the wrench measurements. In such cases, we suggest supplementing VIRDO++ 's input with additional term such as end-effector accelerations. **3) Effective generalization across different environments:** Although we provide an example of environment generalization (wok experiments shown in Fig. 8), the performance is not as stable as our other demonstrated applications. VIRDO++ may benefit from a complete or partial 3D model of the environment to integrate inter-penetration losses [27] or collision checkers [42].

References

[1] Y. Wi, P. Florence, A. Zeng, and N. Fazeli. VIRDO: visio-tactile implicit representations of deformable objects. *CoRR*, abs/2202.00868, 2022. URL https://arxiv.org/abs/2202.00868.

[2] H. Yin, A. Varava, and D. Kragic. Modeling, learning, perception, and control methods for deformable object manipulation. *Science Robotics*, 6(54), 2021.

[3] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *The International Journal of Robotics Research*, 37(7):688–716, 2018.

[4] V. E. Arriola-Rios, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano, and J. L. Wyatt. Modeling of deformable objects for robotic manipulation: A tutorial and review. *Frontiers in Robotics and AI*, 7:82, 2020.

[5] Y. C. Hou, K. S. M. Sahari, and D. N. T. How. A review on modeling of flexible deformable object for dexterous robotic manipulation. *International Journal of Advanced Robotic Systems*, 16(3): 1729881419848894, 2019.

[6] I. A. Essa, S. Sclaroff, and A. Pentland. A unified approach for physical and geometric modeling for graphics and animation. In *Computer Graphics Forum*, volume 11, pages 129–138. Wiley Online Library, 1992.

[7] F. Ficuciello, A. Migliozzi, E. Coevoet, A. Petit, and C. Duriez. Fem-based deformation control for dexterous manipulation of 3d soft objects. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4007–4013. IEEE, 2018.

[8] B. Frank, C. Stachniss, R. Schmedding, M. Teschner, and W. Burgard. Learning object deformation models for robot motion planning. *Robotics and Autonomous Systems*, 62(8):1153–1174, 2014.

[9] M. Müller and M. H. Gross. Interactive virtual materials. In *Graphics interface*, volume 2004, pages 239–246, 2004.

[10] A. Sengupta, A. Krupa, and E. Marchand. Tracking of non-rigid objects using rgb-d camera. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 3310–3317. IEEE, 2019.

[11] A. M. Howard and G. A. Bekey. Intelligent learning for deformable object manipulation. *Autonomous Robots*, 9(1):51–58, 2000.

[12] A. Nurnberger, A. Radetzky, and R. Kruse. A problem specific recurrent neural network for the description and simulation of dynamic spring models. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 1, pages 468–473. IEEE, 1998.

[13] D. L. Tonnesen. *Dynamically coupled particle systems for geometric modeling, reconstruction, and animation.* Citeseer, 2000.

[14] B. Thach, B. Y. Cho, A. Kuntz, and T. Hermans. Learning visual shape control of novel 3d deformable objects from partial-view point clouds. *arXiv preprint arXiv:2110.04685*, 2021.

[15] B. Thach, A. Kuntz, and T. Hermans. Deformernet: A deep learning approach to 3d deformable object manipulation. *arXiv preprint arXiv:2107.08067*, 2021.

[16] T. Weng, S. M. Bajracharya, Y. Wang, K. Agrawal, and D. Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *Conference on Robot Learning*, pages 192–202. PMLR, 2022.

[17] X. Lin, Z. Huang, Y. Li, J. B. Tenenbaum, D. Held, and C. Gan. Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools. *arXiv preprint arXiv:2203.17275*, 2022.

[18] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg. Deep transfer learning of pick points on fabric for robot bed-making. *arXiv preprint arXiv:1809.09810*, 2018.

[19] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg. Visuospatial foresight for physical sequential fabric manipulation. *Autonomous Robots*, pages 1–25, 2021.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[21] Y. Deng, J. Yang, and X. Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. *CoRR*, abs/2011.13650, 2020. URL https://arxiv.org/abs/2011.13650.

[22] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[23] P. Sundaresan, R. Antonova, and J. Bohg. Diffcloud: Real-to-sim from point clouds with differentiable simulation and rendering of deformable objects. *arXiv preprint arXiv:2204.03139*, 2022.

[24] R. Antonova, J. Yang, P. Sundaresan, D. Fox, F. Ramos, and J. Bohg. A bayesian treatment of real-to-sim for deformable object manipulation. *IEEE Robotics and Automation Letters*, 7(3):5819–5826, 2022. doi:10.1109/LRA.2022.3157377.

[25] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravanszky, G. State, M. Lu, et al. Factory: Fast contact for robotic assembly. *arXiv preprint arXiv:2205.03532*, 2022.

[26] S. Yao and K. Hauser. Online estimation of point-based volumetric stiffness model using joint torque sensors.

[27] A. Petit, S. Cotin, V. Lippiello, and B. Siciliano. Capturing deformations of interacting non-rigid objects using rgb-d data. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 491–497. IEEE, 2018.

[28] J. Zhang, Y. Zhong, J. Smith, and C. Gu. Neural dynamics-based poisson propagation for deformable modelling. *Neural Computing and Applications*, 31(2):1091–1101, 2019.

[29] Y. Wang, D. McConachie, and D. Berenson. Tracking partially-occluded deformable objects while enforcing geometric constraints. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14199–14205. IEEE, 2021.

[30] P. Palafox, A. Božič, J. Thies, M. Nießner, and A. Dai. Npms: Neural parametric models for 3d deformable shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12695–12705, October 2021.

[31] X. Ma, D. Hsu, and W. S. Lee. Learning latent graph dynamics for deformable object manipulation. *arXiv preprint arXiv:2104.12149*, 2021.

[32] I. Santesteban, N. Thuerey, M. A. Otaduy, and D. Casas. Self-supervised collision handling via generative 3d garment models for virtual try-on. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11763–11773, June 2021.

[33] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. *arXiv preprint arXiv:2205.02909*, 2022.

[34] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam. isdf: Real-time neural signed distance fields for robot perception. *arXiv preprint arXiv:2204.02296*, 2022.

[35] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *International Conference on Computer Vision*, Oct. 2019.

[36] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021.

[37] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg. Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9411–9418. IEEE, 2020.

[38] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

[39] I. Taylor, S. Dong, and A. Rodriguez. Gelslim3. 0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. *arXiv preprint arXiv:2103.12269*, 2021.

[40] J. Yin, G. M. Campbell, J. Pikul, and M. Yim. Multimodal proximity and visuotactile sensing with a selectively transmissive soft membrane. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pages 802–808. IEEE, 2022.

[41] N. Kuppuswamy, A. Alspach, A. Uttamchandani, S. Creasey, T. Ikeda, and R. Tedrake. Soft-bubble grippers for robust and perceptive manipulation. *arXiv preprint arXiv:2004.03691*, 2020.

[42] P. Mitrano, D. McConachie, and D. Berenson. Learning where to trust unreliable models in an unstructured world for deformable object manipulation. *Science Robotics*, 6(54):eabd8170, 2021.