VIRDO: Visio-tactile Implicit Representations of Deformable Objects

Youngsun Wi[1], Pete Florence[2], Andy Zeng[2] and Nima Fazeli[1]

***Abstract*— Deformable object manipulation requires computationally efficient representations that are compatible with robotic sensing modalities. In this paper, we present VIRDO: an implicit, multi-modal, and continuous representation for deformable-elastic objects. VIRDO operates directly on visual (point cloud) and tactile (reaction forces) modalities and learns rich latent embeddings of contact locations and forces to predict object deformations subject to external contacts. Here, we demonstrate VIRDOs ability to: i) produce high-fidelity cross-modal reconstructions with dense unsupervised correspondences, ii) generalize to unseen contact formations, and iii) state-estimation with partial visio-tactile feedback.**

I. Introduction

Dexterous manipulation of deformable objects is an important open problem in robotics [1], [2]. These objects are ubiquitous in our day-to-day lives and play a key role in many applications including cooking, manufacturing, in-home assistive care, and surgery. Despite their prevalence and importance, deformable objects have received less attention than their rigid counterparts owing to their inherent complexities in modeling, perception, and controls [2]–[5]. To illustrate, the states of rigid bodies with known geometries can be succinctly represented with 6D pose and velocity. However, deformable objects have an infinite continuum of states and their representation and perception remains an open problem [2].

In this paper, we present VIRDO – an implicit, dense, cross modal, and continuous architecture that addresses these fundamental representation and perception challenges for the class of elastically deformable objects. The central feature of our method is learning deformation fields informed by cross modal visual and tactile cues of external contacts. We further contribute a dataset of elastically deformable objects with boundary conditions used to evaluate. This paper focuses on dense geometric representations because they can facilitate downstream tasks such as state-estimation from partial views and estimating dense correspondences, as we demonstrate, as well as bootstrapping keypoint/affordance learning.

Problem Statement & Assumptions : Our goal is to derive a computationally efficient and generative model that: 1) predicts object deformations subject to external forces; and 2) is compatible with common robotic sensors. We assume the object geometry is described by its point cloud: an unordered set $\boldsymbol{P} := \{\boldsymbol{p} \in \mathbb{R}^3 : \text{SDF}(\boldsymbol{p}) = 0\}$ where SDF denotes the signed distance w.r.t. the surface of the object. Point clouds are obtained from commodity depth sensors or 3D scanners commonly found in industry. Contact locations are also given as a set of points $\boldsymbol{Q}$ which can be given by an upstream perception algorithm such as [6]–[8]. For the tactile input, net reaction force is given by $\boldsymbol{u} \in \mathbb{R}^3$ at the wrist of the robot which can be measured by common industrial F/T sensors or recovered from joint torques. In this paper, we derive a continuous and implicit representation of the deformed object geometry ($f(\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{u}) = s$) subject to external forces and their locations. Here the object geometry is given by the zero-level set of the implicit function; i.e. $s = 0$.

[1] Youngsun Wi and Nima Fazeli are with the Robotics Institute at the University of Michigan, MI, USA <yswi,nfz>@umich.edu

[2] Pete Florence and Andy Zeng with Robotics at Google, Mountain View, CA, USA <peteflorence,andyzeng>@google.com

II. Representing Deformable Objects using SDFs

At a high-level, VIRDO decomposes object representations into a nominal shape representation and a point-wise deformation field. Here, we choose signed distance fields as our underlying representation and discuss this choice in Sec. IV. The nominal shape representation decodes latent shape embeddings $\boldsymbol{\alpha} \in \mathbb{R}^l$ into continuous signed-distance fields, the zero-level set of which is the undeformed object geometry – similar to the architectures proposed in [9]–[11]. The point-wise deformation field is produced using a summary of all boundary conditions (contact locations, reaction force, and fixed constraints) leveraging a permutation invariant set operator. The structure is fully differentiable and can be learned end-to-end. In the following, we discuss each component in more detail.

A. Nominal Shape Representation

The nominal shape of an object is the geometry it takes in the absence of external contact forces. The nominal geometry is produced by the object module as $\boldsymbol{O}(\boldsymbol{x}|\boldsymbol{\Psi}_o(\boldsymbol{\alpha})) = s$ as a signed distance field, where $\boldsymbol{x} = (x, y, z)$ is a query point, and s is the signed-distance. The purpose of the object code (α) is to allow VIRDO to represent multiple objects. Here, we use a hyper-network $\boldsymbol{\Psi}_o(\boldsymbol{\alpha})$ to decode the object code into object module's weights and biases $\boldsymbol{\theta}_o$, similar to [12].

We pre-train VIRDO on nominal shapes before training on deformations. During this stage, we initialize the object codes as $\boldsymbol{\alpha} \sim N(0, 0.1^2)$ and update them with hyper-network $\boldsymbol{\Psi}_o$. The loss function for pretraining nominal shapes is:

$$L_{nominal} = L_{sdf} + \lambda_2 L_{latent} + \lambda_3 L_{hyper}, \tag{1}$$

where L_{sdf} is defined as:

$$L_{sdf} = \sum_{i=1}^{N} \Big(\sum_{\tilde{\boldsymbol{x}} \in \boldsymbol{\Omega}} |clamp(\boldsymbol{O}^i(\boldsymbol{x}), \delta) - clamp(s^*, \delta)| + \lambda \sum_{\tilde{\boldsymbol{x}} \in \boldsymbol{\Omega}_0} (1 - \langle \nabla \boldsymbol{O}^i(\boldsymbol{x}), \boldsymbol{n}^* \rangle) \Big).$$

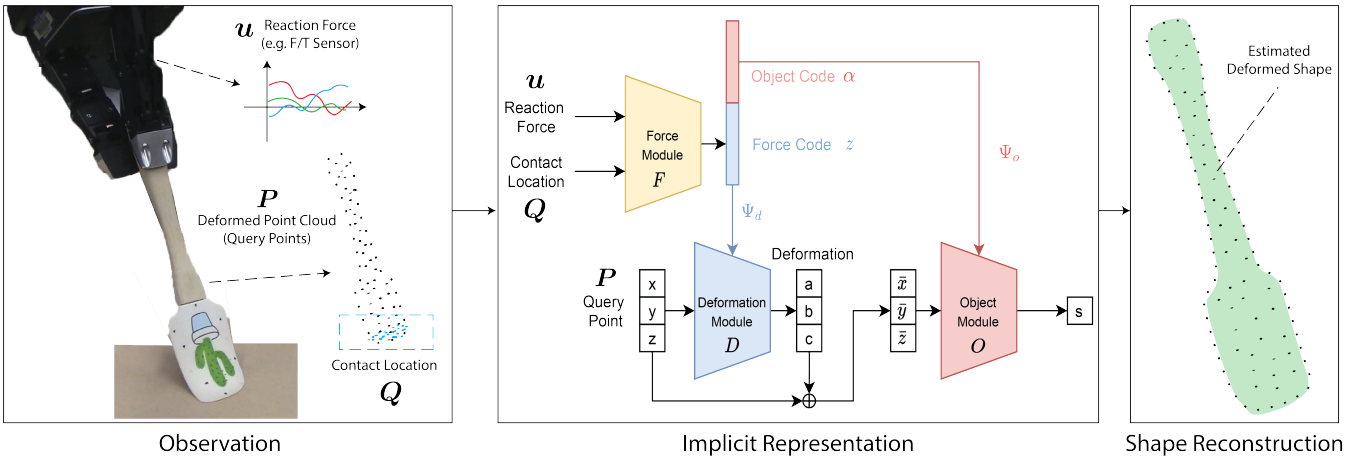

Fig. 1: **Representation Architecture:** The left panel depicts how visual data in the form of point clouds and tactile in the form of reaction forces may be collected in practical robotic settings. The middle panel depicts the network and how this information is processed to predict the implicit surface representation encoded as a signed-distance function. Finally, the right panel depicts the reconstruction of the estimated true surface given the external contacts and reaction force.

For convenience, we introduce a shorthand $\boldsymbol{O}^i(\boldsymbol{x}) = \boldsymbol{O}(\boldsymbol{x}|\Psi_o(\boldsymbol{\alpha_i}))$ to denote the signed-distance field corresponding to the i^{th} object code. Also, $\boldsymbol{\Omega}$ is the 3D querying space, $\boldsymbol{\Omega}_0$ is the zero-level surfaces in the querying space, $\boldsymbol{s}^*$ is the ground truth signed-distance, $\boldsymbol{n}^*$ is the ground truth normal, and ∇ denotes the spatial gradient. δ is for clamping off-surface signed-distances to concentrate network capacity on details near the surface, as demonstrated in [9]. For latent space regularization, we impose a Gaussian prior on the object code $L_{latent}(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \|\frac{1}{l}\boldsymbol{\alpha}^i\|_2$ and weights of the network $L_{hyper}(\boldsymbol{\theta}_o) = \sum_{i=1}^{N} \frac{1}{l_o}\|\boldsymbol{\theta}_o^i\|_2$ where l_o is the length of $\boldsymbol{\theta}_o$. After the pre-training, we fix the object codes as constant vectors, but continue updating the object module and the hyper-network with low learning rate($\sim$ 1e-8) for additional refinement.

B. Deformed Object Representation

The Force Module $\boldsymbol{F}$ is an encoder that summarizes the contact locations and reaction force $(\boldsymbol{Q}, \boldsymbol{u})$ into a force code $\boldsymbol{z} = \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{u})$. We assume that the contact location set $\boldsymbol{Q}$ is given as a subset of the nominal point cloud ($\boldsymbol{Q} \subset \boldsymbol{P}$) and the reaction force $\boldsymbol{u} \in \mathbb{R}^3$ is directly measured at the robot's wrist. Point clouds, including the contact location set $\boldsymbol{Q}$, are unordered and variable in length. Our contact location encoder utilizes the PointNet architecture [13].

We define the deformation field as a 3D vector field that pushes a deformed object back to its original (nominal) shape. VIRDO recovers the signed-distance field of the nominal shape by adding the deformation field to the SDF of the deformed shape. We highlight that VIRDO has learned to focus the deformation field around the boundary of the object with magnitude reflecting the amount of deformation.

VIRDO represents the deformation field as $\boldsymbol{D}\big(\boldsymbol{x}|\boldsymbol{\Psi}_d(\boldsymbol{z}, \boldsymbol{\alpha})\big)$, where the deformation module $\boldsymbol{D}$ shares the same structure as the object module $\boldsymbol{O}$ with parameters $\boldsymbol{\theta}_d$ predicted by the hyper-network $\boldsymbol{\Psi}_d$. We highlight that $\boldsymbol{\theta}_d$ is conditioned on the latent code pair $(\boldsymbol{z}, \boldsymbol{\alpha}) \in \mathbb{R}^{l+m}$ to capture the underlying object-specific deformation behavior. This results in $\boldsymbol{D}$ predicting different deformation fields for different objects despite similar contact locations and reaction force measurements. This is desirable because objects may be geometrically similar but deform differently due to varying material properties. We will demonstrate examples of this in Sec. III-B.

Using the shorthand $\boldsymbol{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{x}) = \boldsymbol{D}\big(\boldsymbol{x}|\boldsymbol{\Psi}_d(\boldsymbol{z}, \boldsymbol{\alpha})\big)$, we can relate the signed-distance fields of the deformed and nominal object via $s = \text{SDF}(\boldsymbol{x}) = \boldsymbol{O}_{\boldsymbol{\Phi}_o}\big(\boldsymbol{x} + \boldsymbol{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{x})\big)$. We note that the deformed point cloud $\boldsymbol{p} \in \boldsymbol{P}$ satisfies $\text{SDF}(\boldsymbol{p}) = \boldsymbol{O}_{\boldsymbol{\Phi}_o}\big(\boldsymbol{p} + \boldsymbol{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{p})\big) = 0$. To learn this mapping, we solve the optimization problem $\underset{\boldsymbol{\theta}_d}{\text{argmin}} f_c(\boldsymbol{P})$, where

$$f_c(\boldsymbol{P}) = \Big(\text{CD}(\boldsymbol{P} + \boldsymbol{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{P}), \bar{\boldsymbol{P}}^*) + \lambda_c \frac{1}{\bar{\boldsymbol{P}}} \sum_{\boldsymbol{p} \in \boldsymbol{P}} \|\boldsymbol{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{p})\|_2 \Big) \tag{2}$$

where $\bar{\boldsymbol{P}}^*$ is the true nominal point cloud of the same length as $\boldsymbol{P}$, λ_c is a weighting for the minimal correction prior similar to [14] and CD is the Chamfer Distance measure between two point clouds.

To learn the full model, we train the deformation, force, and pre-trained object modules end-to-end using the loss function:

$$L_{deformed} = f_c(\{\boldsymbol{x} | \boldsymbol{x} \in \boldsymbol{\Omega}_0\}) + \lambda_1 L_{sdf} + \lambda_3 L_{hyper} + \lambda_4 L_{latent}. \tag{3}$$

The first term solves the optimization in Eq. 2 with on-surface points. The second loss term L_{sdf} couples the deformation field to the signed-distance field of the object module:

$$L_{sdf} = \sum_{i=1}^{M} \Big(\sum_{\boldsymbol{x} \in \boldsymbol{\Omega}} |clamp(\boldsymbol{O}_{\boldsymbol{\Phi}_o}\big(\boldsymbol{x} + \boldsymbol{D}_{\boldsymbol{\Psi}_d}(\boldsymbol{x}), \delta\big) - clamp(s^*, \delta)| + \sum_{\boldsymbol{x} \in \boldsymbol{\Omega}_0} \lambda_n (1 - \langle \nabla \boldsymbol{O}^i(\boldsymbol{x}), \boldsymbol{n}^* \rangle) \Big) \tag{4}$$

where M is the total number of deformed objects. The $L_{hyper}(\boldsymbol{x})$ and $L_{latent}(\boldsymbol{x})$ are the Gaussian prior on the latent code and the network parameters. Since we are updating two

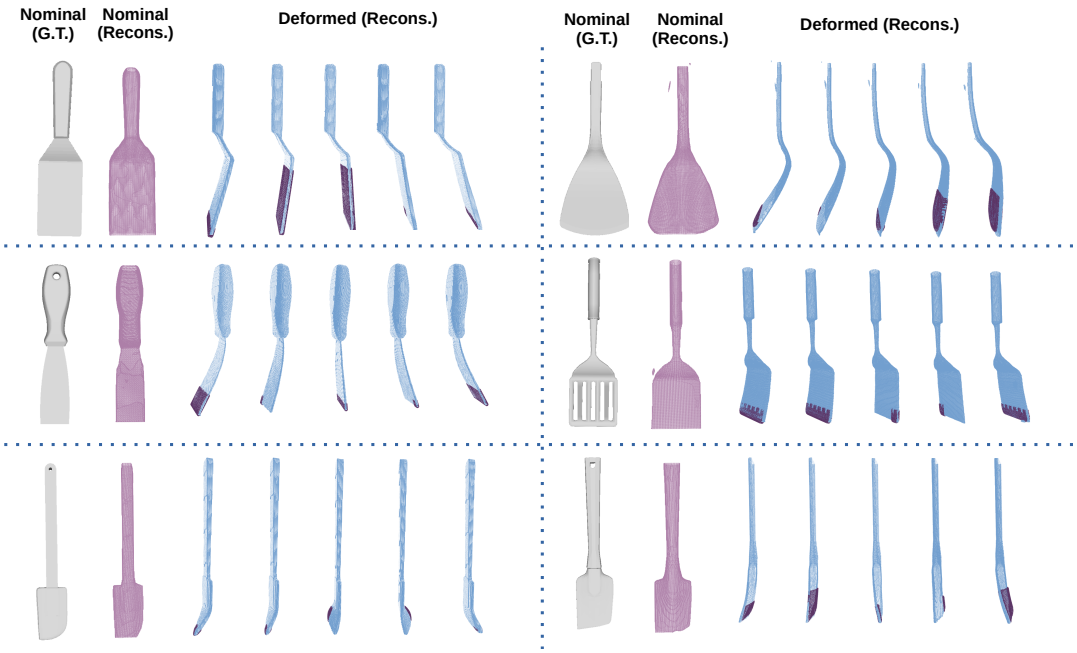

Fig. 2: **Reconstruction Results:** Reconstructions of multiple nominal shapes and their deformations, learned simultaneously by VIRDO. Marching Cube algorithm is used for the reconstruction, where we highlighted the contact location as purple region.

hyper-networks, L_{hyper} is the weighted sum of $\mathbf{\Psi}_o$ and $\mathbf{\Psi}_d$ and the L_{latent} is $\sum_{i=1}^{N} \|\frac{1}{l}\boldsymbol{z}\|_2$.

III. Experiments and Results

A. Data Preparation

In total, we generated 6 objects each with 24 unique boundary conditions using MATLAB PDE toolbox. The 3D meshes were collected from open-sourced 3D model repository. For the training, we normalized the point cloud with the geometric center at $[0,0,0]$.

B. Representing Known Deformable Shapes

The average reconstruction accuracy is $0.3474\text{x}10^3$ in Chamfer Distance(CD). CD is measured between reconstructions and query points unseen during training in average, where we utilized Marching Cube algorithm [15] for the reconstruction. We emphasize that only one neural network model was used for the entire data-set.

C. Deformation Field Inference

We test the model's ability to infer a deformation field given reaction force, partial pointcloud, and object code, seen from the training. Here, we infer the contact feature to estimate deformation. First, we randomly initialize the contact feature from $N(0, 0.01^2)$. Then, we update the feature with an L1 loss which only consumes a partial zero-level set: $L_{infer} = \sum_{\boldsymbol{x}\in\boldsymbol{\Omega_o}} |clamp(\boldsymbol{O}(\boldsymbol{x}), \delta)|$. The loss encourages VIRDO to update the contact feature by minimizing the mismatch between the initial guess and the partial observation. Fig. 3 is a partial pointcloud where the handle and the tip are occluded, rendered in simulation with a single pinhole camera. At epoch 0, the model already makes deformation field fairly close to the ground truth. This shows VIRDO's ability to perform state estimation when the vision is missing. As the gradient descent progresses on the contact feature, the estimated deformation converges towards the ground truth. We note that only the on-surface points are used for this experiment, since an RGBD camera would feasibly only give on-surface points in real-world experiments; however, it is also possible boost the inference performance by collecting off-surface samples along the camera ray and augment the partial observation.

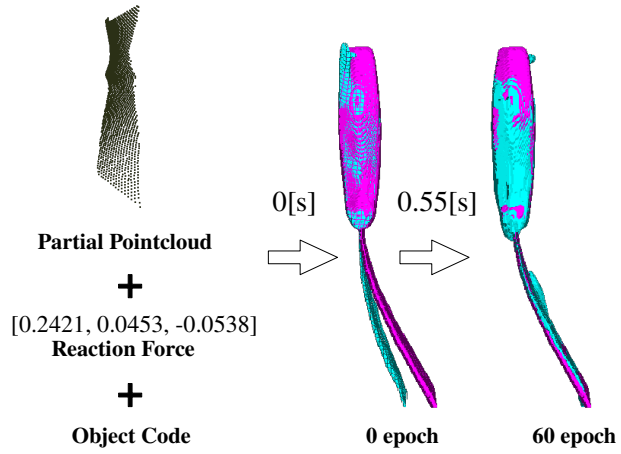

Fig. 3: **Inference:** Reconstructions with inferred deformation field (cyan), ground truth deformed object (magenta)

D. Generalization and Code Interpolation/Extrapolation

In this section, we evaluate VIRDO's ability to generalize to unseen contact formations. This functionality is important for robotic applications given the wide variety of contact interactions. To this end, we use the model from Sec. III-B (trained on 144 deformations) and evaluate the reconstruction accuracy of 6 unseen contact formations for the object

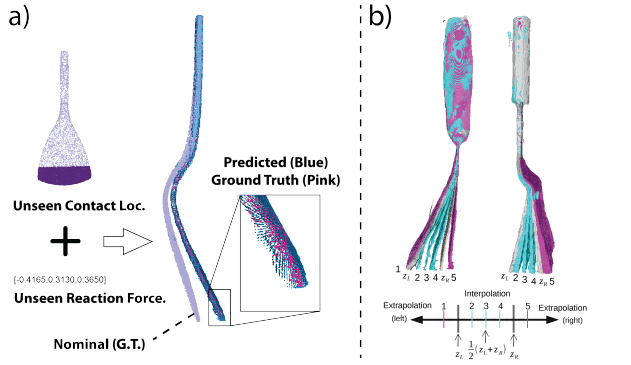

Fig. 4: **a) Generalization:** Example of shape estimation given unseen contact formation during training. **b) Latent Code Interpolation and Extrapolation:** Two trained force codes $\boldsymbol{z}_l$ and $\boldsymbol{z}_r$ are interpolated and extrapolated evenly as shown where 1-5 indicate tested force code and corresponding reconstruction result.

depicted in Fig.4a). To better understand the generalization properties of VIRDO, we interpolate and extrapolate in latent force code space. This task evaluates the continuity and semantic meaning of the latent space; e.g. whether the reconstructions are similar to direct inter/extrapolation of two deformations. Given an object, we pick $\boldsymbol{z}_l$ and $\boldsymbol{z}_r$ among the successfully trained force codes; $\boldsymbol{z}_l$ bears the maximum deflection to the left (or -x direction) and $\boldsymbol{z}_r$ generates moderate deflections in a random direction. We then linearly interpolate and extrapolate the 32 dimensional force code. Fig.4b) shows the resulting reconstructions. We note the interpolations and extrapolations are smooth, even, and continuous. This suggests a well-formed latent force space and explains the effectiveness of the model in generalizing to unseen contact formations.

We note that the reconstruction for extrapolation (1) is omitted from the object on the right in Fig. 4. This is because of a failure case resulting in a noisy and poor quality reconstruction. We found that the extrapolation results can be inaccurate if it exceeds the bound of maximum deflections seen in the training data.

IV. Discussion & Limitations

The fundamental principle driving VIRDO is the ability to learn deformation fields informed by visio-tactile sensing. VIRDO is the first learned implicit method to integrate tactile and visual feedback while modeling object deformations subject to external contacts. The representation has arbitrary resolution and is cheap to evaluate for point-wise sampling. Additionally, the latent code is well-behaved and can be used for inference.

V. Acknowledgement

This research is partly supported by Robotics at Google.

References

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019.

[2] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 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*, vol. 37, no. 7, pp. 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*, vol. 7, p. 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*, vol. 16, no. 3, p. 1 729 881 419 848 894, 2019.

[6] T. Hermans, F. Li, J. M. Rehg, and A. F. Bobick, "Learning contact locations for pushing and orienting unknown objects," in *2013 13th IEEE-RAS international conference on humanoid robots (humanoids)*, IEEE, 2013, pp. 435–442.

[7] M. Sharma and O. Kroemer, "Relational learning for skill preconditions," *arXiv preprint arXiv:2012.01693*, 2020.

[8] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 5062–5069.

[9] 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*, 2019, pp. 165–174.

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

[11] M. Tancik, P. P. Srinivasan, B. Mildenhall, *et al.*, "Fourier features let networks learn high frequency functions in low dimensional domains," *arXiv preprint arXiv:2006.10739*, 2020.

[12] Y. Deng, J. Yang, and X. Tong, "Deformed implicit field: Modeling 3d shapes with learned dense correspondence," *arXiv preprint arXiv:2011.13650*, 2020.

[13] 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*, 2017, pp. 652–660.

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

[15] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.