

NISP: Neural Inverse Source Problem

Youngsun Wi, Jayjun Lee, Miquel Oller and Nima Fazeli

Abstract—Reconstructing unknown external source functions is an important perception capability for a large range of robotics domains including manipulation, aerial, and underwater robotics. In this work, we propose a Physics-Informed Neural Network (PINN [1]) based approach for solving the inverse source problems in robotics, jointly identifying unknown source functions and the complete state of a system given partial and noisy observations. Our approach demonstrates several advantages over prior works (Finite Element Methods (FEM) and data-driven approaches): it offers flexibility in integrating diverse constraints and boundary conditions; eliminates the need for complex discretizations (e.g., meshing); easily accommodates gradients from real measurements; and does not limit performance based on the diversity and quality of training data. We validate our method across three simulation and real-world scenarios involving up to 4th order partial differential equations (PDEs), constraints such as Signorini and Dirichlet, and various regression losses including Chamfer distance and L2 norm. <https://www.mmintlabor.com/nisp>

I. INTRODUCTION

We are interested in differential equations g with an unknown external source f :

$$g(\mathbf{x}, \Phi, \nabla_{\mathbf{x}}\Phi, \nabla_{\mathbf{x}}^2\Phi, \dots) = f(\mathbf{x}), \quad \Phi : \mathbf{x} \mapsto \Phi(\mathbf{x}). \quad (1)$$

Our objective is to determine the external source function $f(\mathbf{x}) \in \mathbb{R}^q$ and the full mapping from spatial/temporal coordinates $\mathbf{x} \in \mathbb{R}^s$ to the quantity of interest $\Phi(\mathbf{x}) \in \mathbb{R}^r$ from known governing equations g and partial/noisy observations. This problem, known as an inverse source problem [2]–[4], is of significant importance in various scientific and engineering domains such as signal processing [5], fluid dynamics [6], [7], optics [8], and more.

To address these inverse source problems in robotics, a common strategy is to use finite element analysis integrated with optimization techniques [9], [10] or physics engine [11], [12]. While these approaches focus on computation efficiencies, they suffer two notable limitations. First, they rely on complex discretization and/or meshing, which affects precision and introduces significant complexity. Second, these approaches require unique treatment for different equations and constraints, which significantly limits their generality. Recent progress in representation learning addresses these issues via learning priors over the simulated source functions, where the data implicitly contains all the constraints and boundary conditions. For example, prior works in manipulation simulate extrinsic contact, a source function exerting forces on deformable [13]–[16] or rigid objects [17], [18]. Another example involves simulating sensor deformations as

a source function transducing electric signals [19]. However, the performance of these approaches relies heavily on the diversity and quality of training data and does not guarantee adherence to the physics, especially with out-of-distribution inputs. Latent force models [20], [21] present an alternative approach using Gaussian processes to identify unknowns in differential equations; however, they are restricted to linear models, and it remains unclear how complex constraints could be effectively integrated.

In this paper, we propose a method for solving the inverse source problem based on Physics-Informed Neural Networks [1] that enables the simultaneous inference of both the source function and the full state given partial and noisy observations. Our method offers several advantages over prior work in robotics: compatibility with various n th-order differential equations (including ordinary and partial differential equations), flexibility in integrating various constraints, initial values, and boundary conditions, elimination of the need for non-trivial discretizations like meshing, ease of accommodating gradients from real measurements, and does not limit performance based on the diversity training data.

II. PROBLEM FORMULATION

Our goal is to learn a mapping $\Psi : \mathbf{x} \mapsto (\Phi(\mathbf{x}), f(\mathbf{x}))$ that satisfies

$$g(\mathbf{x}, \Phi, \nabla_{\mathbf{x}}\Phi, \nabla_{\mathbf{x}}^2\Phi, \dots) = f(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_n \quad (2)$$

subject to

$$C_m(\mathbf{x}, \Phi_\theta, \nabla_{\mathbf{x}}\Phi, \nabla_{\mathbf{x}}^2\Phi, \dots, f(\mathbf{x})) = 0, \quad \forall \mathbf{x} \in \Omega_m. \quad (3)$$

where \mathbf{x} is a spatial/temporal coordinates we have partial/noisy access to, $\Phi(\mathbf{x})$ is the fully/partially observable quantity of interest (e.g., deformation field, electric field, etc.), $f(\mathbf{x})$ is a Lipschitz continuous source function, and Ω_n, Ω_m are subsets of a bounded domain Ω . When we approximate the mapping with a neural network as $\Psi_\theta(\mathbf{x})$ and $f_\theta(\mathbf{x})$, this problem becomes solving for the network parameters θ , satisfying the differential equations $g(\mathbf{x}, \Phi, \nabla_{\mathbf{x}}\Phi_\theta, \nabla_{\mathbf{x}}^2\Phi_\theta, \dots) - f_\theta(\mathbf{x}) = 0$ and M constraints $\{C_m(\mathbf{x}, \Phi_\theta, \nabla_{\mathbf{x}}\Phi_\theta, \nabla_{\mathbf{x}}^2\Phi_\theta, \dots, f_\theta(\mathbf{x}))\}_{m=1}^M$. We cast this problem into a loss:

$$\frac{1}{|\Omega_n|} \int_{\Omega_n} (g_n(\mathbf{x}, \Phi_\theta, \nabla_{\mathbf{x}}\Phi_\theta, \dots) - f_\theta(\mathbf{x}))^2 d\mathbf{x} \quad (4)$$

$$+ \sum_{m=1}^M \frac{1}{|\Omega_m|} \int_{\Omega_m} C_m(\mathbf{x}, \Phi_\theta, \nabla_{\mathbf{x}}\Phi_\theta, \dots, f_\theta(\mathbf{x}))^2 d\mathbf{x}. \quad (5)$$

¹ Youngsun Wi, Jayjun Lee, Miquel Oller, and Nima Fazeli are with the Robotics Department at the University of Michigan, MI, USA <yswi, jayjun, oller, nfz>@umich.edu

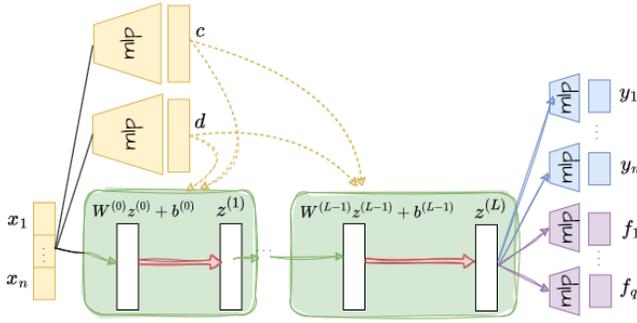


Fig. 1: Modified MLP with L layers. Green arrows indicate fully connected layers, and red arrows indicate Eq. 6 operations. L is the number of the green box module from the input x to the output $\Psi_\theta(\mathbf{x}) = (\mathbf{y}(\mathbf{x}), \mathbf{f}(\mathbf{x}))$.

where the first term represents residual losses (\mathcal{L}_r) of the differential equation (DE), and the second term represents the constraint loss. The constraint loss includes various forms of regression losses (\mathcal{L}_{reg}) enforcing desired output at specific coordinates (e.g., Chamfer Distance, Mean Square Error), boundary conditions (e.g., Dirichlet, Neumann, Signorini), and equality and inequality constraints.

III. METHODOLOGY

A. Modified Multi-layer Perceptron

We use a modified multi-layer perceptron (MLP), which has been shown to excel at learning PINNs [22], [23]. The modified MLP's l -th layer has 2 steps:

$$z'^{(l)} = \tanh(\mathbf{W}^{(l)} z^{(l-1)} + \mathbf{b}^{(l)}), \quad (6)$$

$$z^{(l)} = \mathbf{c} \cdot z'^{(l)} + \mathbf{d} \cdot (1 - z'^{(l)}). \quad (7)$$

The first step is a fully connected layer with weight $\mathbf{W}^{(l)}$, bias $\mathbf{b}^{(l)}$, and a \tanh activation function acting on the previous layer output $z^{(l-1)}$. The second step weights the output $z'^{(l)}$ with \mathbf{c} and \mathbf{d} from separate MLPs with a \tanh activation function, as in Fig. 1. Here, we propose the last fully connected layer unique to each dimension of \mathbf{y} and \mathbf{f} .

B. Incorporating Inductive Bias into Network Design

Many practical robotics systems have low-frequency signals, and building a network with bandwidth matching the underlying signal is essential for robustness to signal noise and successful training. We found that removing the input mappings helps the model be robust to measurement noise, whereas prior works in AI4science [22], [24] highlight the necessity of the input mappings, particularly when focusing on high-frequency signals.

C. Non-dimensionalization

Ensuring inputs and outputs have reasonable scales is crucial for successful training [24]–[26]. We normalize inputs to be within a unit cube $[-1, 1]^s$ and each dimension of the outputs to have similar scales. For instance, if we scale inputs in $[m]$ unit by some value k , we scale Young's modulus $[N/m^2]$ by $\frac{1}{k^2}$ and depth measurements in $[m]$ unit by k .

D. Loss Weighting

We found that an effective training strategy is to quickly regress to the partial measurements and then gradually solve for the forcing function. One way to control the convergence speed of each loss term is by updating the loss weights every hundred epochs using each loss term's gradients [24]; however, this approach is expensive in time and space, especially with highly nonlinear and high-order residual losses. Instead, we recommend fixed loss weightings satisfying $50\lambda_r \|\nabla_\theta \mathcal{L}_r\| \leq \lambda_{reg} \|\nabla_\theta \mathcal{L}_{reg}\|$, where $\nabla_\theta \mathcal{L}_{reg}$ and $\nabla_\theta \mathcal{L}_r$ are the gradients of the regression and residual loss.

IV. REAL-WORLD EXPERIMENTS

We present two real-world inverse source problems that vary in several key aspects: the order of the differential equations, the degree of observability (ranging from partial to full measurements), and the levels of noise in the data. Additionally, we demonstrate system parameter identification using the same framework.

A. Softbody Contact Problem

A softbody with linear elasticity interacting with a rigid environment is known as a Signorini Problem. This problem is defined in 3D coordinates \mathbf{u} of an undeformed softbody $\mathbf{x} \in \mathbb{R}^3$. The quantity of interest $\mathbf{u}(\mathbf{x}) \in \mathbb{R}^3$ is the deformation field, and the forcing function $f(\mathbf{x}) \in \mathbb{R}^3$ is contact pressure applied perpendicular to the surface. Following [27], our problem is:

$$\Phi : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3, \quad \mathbf{x} \mapsto \mathbf{u}(\mathbf{x}), \quad (8a)$$

$$\sigma(\mathbf{x}) \mathbf{n} = f(\mathbf{x}) \geq 0 \quad \text{on } \Omega_b, \quad (8b)$$

$$\text{div } \sigma(\mathbf{x}) + \rho \mathbf{g} = \mathbf{0} \quad \text{in } \Omega, \quad (8c)$$

$$\sigma(\mathbf{x}) = \mathbf{D} \varepsilon(\mathbf{x}), \quad \varepsilon(\mathbf{x}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T). \quad (8d)$$

where $\sigma(\mathbf{x})$ is the stress tensor, \mathbf{n} is the normal vector at the query, ρ is the density, \mathbf{g} is the gravitational constant, \mathbf{D} is the elasticity tensor, and $\varepsilon(\mathbf{x})$ is the strain tensor. Domain Ω is the entire volume of interest with the boundary ω_b . The domain Ω_s is the four sides of the cubed sponge, where we assume no contact for simplification. Eq. 8 represent the force equilibrium at the contact and the infinitesimal volume at \mathbf{x} , respectively. $f(\mathbf{x})$ is always greater than 0 because normal tractions can only be compressive.

Dataset: We select 20 examples from [13]'s real-world dataset, where each data sample consists of a single deformed sponge's partial and noisy measurement $(\mathbf{P}_m, \mathbf{w}_m)$. The sponge is a 46 mm cube with a Young's modulus $E = 1.1 \times 10^4$ and a Poisson's ratio $\nu = 0.1$. The sponge was attached to the Pandas Franka Emika robot with a force-torque sensor (ATI-Gamma) mounted at the wrist. This dataset includes ground truth contact locations for evaluation, obtained by directly observing the contact location through a transparent acrylic plate.

Training: Our model takes the wrist wrench and partial point clouds as input, and it predicts the full deformation and contact pressure. If we define the entire 3D query space as

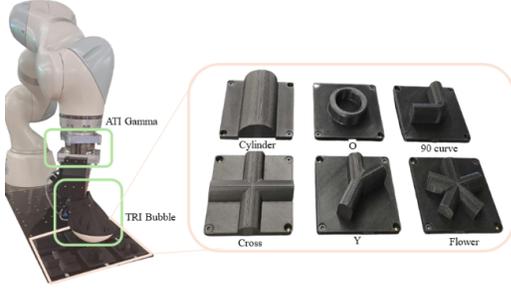


Fig. 2: Soft Bubbles test data collection setup with 6 objects.

$\Omega = [-0.43, 0.43]^3$, all six surfaces as Ω_b , and four sides as Ω_s , the loss function is defined as:

$$\begin{aligned} \mathcal{L}(\theta) = & \lambda_1 CD(P_m, P_\theta) + \lambda_2 \|\mathbf{w}_m - \mathbf{w}_\theta\|^2 \\ & + \frac{\lambda_3}{|\Omega_s|} \sum_{x \in \Omega_s} \|f(x)\|^2 + \frac{\lambda_4}{|\Omega|} \sum_{x \in \Omega} \sigma_n(x) u(x) \cdot n \\ & + \frac{\lambda_5}{|\Omega_b|} \sum_{x \in \Omega_b} \|\sigma(x) n - f(x)\|^2 \\ & + \frac{\lambda_6}{|\Omega|} \sum_{x \in \Omega} \|\text{div } \sigma(x) + \rho g\|^2. \end{aligned} \quad (9)$$

where CD is a single directional Chamfer Distance from P_m to estimated deformed surface pointcloud $P_\theta = \{\mathbf{x} + \mathbf{u}(\mathbf{x}) | \mathbf{x} \in \Omega_b\}$ and \mathbf{w}_θ is a wrist wrench estimation calculated from the estimated contact pressure as in [28]. The third term represents an inductive bias that there are no contacts at the sides, and the fourth term represents a Signorini constraint, where $\sigma_n = \mathbf{n} \cdot \sigma(\mathbf{x})\mathbf{n}$. The last two terms are residual losses from Eq. 9.

B. Membrane-based Tactile Sensor

Here, we emphasize the capacity of our method to solve high-order and complex differential equations' inverse source problems. We show our model's ability to perform system identification with the same framework and robustness to partial and noisy observations unlike [9], [10].

Thin membrane system [29] is defined in 2D in-plane coordinates α and β . The quantity of interest is the 3D deformation $[\mathbf{u}, w]$ corresponding to the undeformed in-plane coordinates of the membrane. The forcing function $f(\mathbf{x}) \in \mathbb{R}$ represents the contact pressure applied perpendicular to the surface normal. The governing equation and the boundary conditions are given by:

$$\begin{aligned} \mathbf{x} = (\alpha, \beta) \in \mathbb{R}^2, \quad \Phi : \mathbf{x} \mapsto (\mathbf{u}, w) \in \mathbb{R}^3, \\ D \Delta^2 w - t \partial_\beta [\sigma_{\alpha\beta}(\mathbf{x}) \partial_\alpha w] - p = f(\mathbf{x}), \quad (10) \\ \text{with } w = 0, \mathbf{u} = \mathbf{0}, f(\mathbf{x}) = 0 \text{ on } \Omega_b. \end{aligned}$$

where $D = Et^3/12(1 - \nu^2)$ is a constant of flexural rigidity consisting of Young's modulus E and Poisson's ratio ν , t is the plate thickness, the biharmonic operator Δ^2 is $\frac{\partial^4}{\partial \alpha^4} + \frac{\partial^4}{\partial \beta^4} + 2 \frac{\partial^4}{\partial \alpha^2 \partial \beta^2}$, $\sigma_{\alpha\beta}$ is the shear stress, $\mathbf{u} \in \mathbb{R}^2$ represents the in-plane displacement, w denotes the out-of-plane deflection, and p is the traverse load per area from air

pressure and gravity. The membrane is subject to a Dirichlet boundary condition in the domain Ω_b .

Dataset: Our dataset $\mathcal{D} = \{P_m, p_m\}$ consists of partial and noisy point clouds P_i and internal air pressures from the pressure sensor p_m . We recorded 24 data samples from 6 different geometries with 4 interactions per object, as shown in Fig. 2. For each interaction, the Soft Bubble randomly rotates in yaw $\in [-90^\circ, 90^\circ]$ and lowers until the contact force reaches $8 \sim 11$ N. We use a Kuka arm equipped with an ATI-gamma F/T sensor mounted on the wrist to move the Soft Bubble. The reaction wrench measurement from the F/T sensors is used only for evaluation purposes of our contact pressure estimation. The Soft Bubble's membrane has a thickness of $t = 0.45$ mm and is an ellipse with major axis $a = 0.06$ m and minor axis $b = 0.04$ m.

System Identification: Our network can also perform system identifications via optimizing for unknown parameters. When we use an inflated sensor pointcloud without contact, the sourcing term is all zero $f_\theta = 0$ and Young's modulus is the only unknown parameter. The loss function becomes

$$\begin{aligned} \min_{E, \theta} \lambda_1 CD(P_m, P_\theta) + \lambda_2 \sum_{x \in \Omega} g_{E, \theta}(x)^2 / |\Omega| \\ + \lambda_3 \sum_{x \in \Omega_b} [w_\theta(x)^2 + \|\mathbf{u}_\theta(x)\|^2 + f_\theta(x)^2] / |\Omega_b|. \end{aligned} \quad (11)$$

where $g_{E, \theta}$ is the left hand side of Eq. 10, $\Omega = \{(\alpha, \beta, w) | (\frac{\alpha}{a})^2 + (\frac{\beta}{b})^2 \leq 1\}$, Ω_b is the boundary of Ω , CD is one-directional Chamfer Distance from noisy partial pointcloud measurement P_m to our predicted full pointcloud $P_\theta = (\alpha, \beta, 0) + (\mathbf{u}_\theta, w_\theta) | (\alpha, \beta) \in \Omega$.

Training: Given a pointcloud under contact P_m , the loss function for contact pressure estimation is

$$\begin{aligned} \min_{\theta} \lambda_1 CD(P_m, P_\theta) + \lambda_2 \sum_{x \in \Omega} [g_\theta(x) - f_\theta(x)]^2 / |\Omega| \\ + \lambda_3 \sum_{x \in \Omega_b} [w_\theta(x)^2 + \|\mathbf{u}_\theta(x)\|^2 + f_\theta(x)^2] / |\Omega_b|. \end{aligned} \quad (12)$$

The only difference between Eq. 11 is non zero f_θ and that the Young's modulus is not updating.

V. RESULTS

A. Softbody Contact Problem

Results: We consider a point to be in contact when the estimated contact pressure exceeds 1,500 Pa, found to be optimal through grid search. The average Chamfer Distance between the ground truth and the estimated contact patch is 45 mm^2 (Ours), whereas the baseline achieves 32 mm^2 .

Baseline: Our SOTA baseline [13] tackles this ill-posed problem by training a neural network on a dataset comprising a total of 3,000 sponges-environment interactions. One-third of the data includes sponge-box interactions, similar to their real-world dataset's sponges-tables interactions. They utilize Isaac Gym's softbody simulation [30], based on finite element method (FEM), which is non-differentiable and can

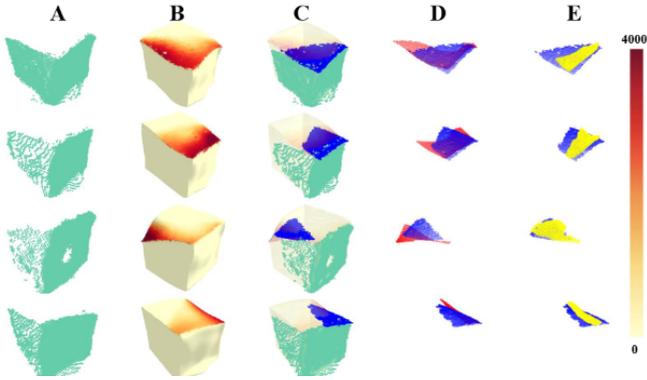


Fig. 3: A) Partial pointcloud measurement (green), B) reconstructed deformed geometry where the color represents predicted contact pressure ranging from 0 to 4,000Pa, C) ground truth contact (blue) and partial observation (green), D) estimated contact location (red) overlaid with the ground truth (blue), E) baseline’s contact location (yellow) overlaid with the ground truth (blue).

only address forward problems. The baseline [13] takes wrist wrench, partial point clouds, and a trial code as inputs, and produces signed distances and binary contact as outputs.

Result: We demonstrate that our results are both qualitatively and quantitatively comparable to the SOTA baseline [13] (Fig. 3). This is impressive considering that our model had to infer from an infinite number of contact possibilities from partial and noisy measurements and physics equations, whereas the baseline [13] relies on strong priors obtained from the data it was trained on.

B. Membrane-based Tactile Sensor

System Identification: The resulting Young’s modulus is $E = 341,260Pa$ when we used $\nu = 0.5$ like typical rubber [9] and $p = 103,320Pa$ from the pressure sensor measurement.

Metrics: We evaluate the predicted contact pressure with 1) contact patch and 2) net contact force estimation. For contact patch evaluations, we use Intersection over Union (IoU) and bidirectional Chamfer Distance (CD). We classify a point in contact if the predicted contact pressure satisfies $f_\theta(\mathbf{x}) > \max(2500, \frac{2}{|\Omega|} \int_\Omega f_\theta d|A|)$, where $\frac{1}{|\Omega|} \int_\Omega f_\theta d|A|$ is the average contact pressure across the entire bubble surface area [9]. Here, the ground truth is the intersection of the Soft Bubble pointcloud and the object mesh, using known transformations. For contact force evaluations, we use L2 norm between the estimated wrench at the wrist (Fig. 2).

Baselines: Our SOTA baseline [9] is a Finite-element (FE) based approach, highly specialized in the Soft Bubble’s 3D contact force and contact location estimation. Unlike our approach, [9] requires correspondence tracking between mesh before and after contact, a special system identification process requiring a total of 205 data points, and a result refinement step for handling partial and noisy observations using a convex optimization (CVXPY). We utilized their finest mesh resolution with 749 vertices along with their best-identified model calibration results.

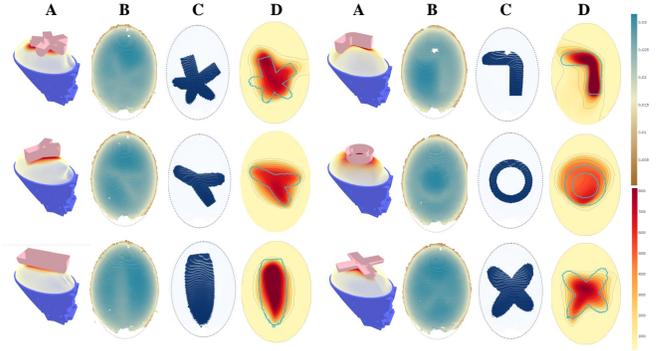


Fig. 4: A) Visualization of real-world Soft Bubble and object (pink) interactions. B) Real-world Soft Bubble point cloud measurement with holes and occlusions. The color represents heights from the x-y plane. C) Ground truth contact mask prediction (navy). D) Estimated contact pressure overlaid with the ground truth contact location indicated in outlines (blue).

Results: Fig. 4 shows examples of bubble-object interactions and the resulting contact pressure estimation from our model. Tab. I indicates that our method excels at contact patch predictions for all shapes except for Y, producing 15% higher IoU and 22.2% smaller chamfer distances when compared to the baseline, highlighting that our predicted contact masks are much smoother, less noisy, and more precise at following the shapes of the objects. Although the predicted contact pressures outperformed results on contact patch estimation, the predicted contact force produced a 1.083 N higher contact force L2 error, which is about 10.8% of the average contact force scale, 9.99 N.

Shape	Ours			[9]		
	IoU \uparrow	CD \downarrow	Force \downarrow	IoU \uparrow	CD \downarrow	Force \downarrow
Flower	0.604	1.000	1.900	0.557	1.026	0.678
O	0.376	2.191	2.056	0.334	2.357	0.706
Y	0.541	1.120	1.741	0.581	0.933	0.357
Cylinder	0.647	0.903	2.758	0.553	1.083	1.642
Cross	0.680	0.635	1.350	0.431	1.750	0.873
90 Curve	0.597	0.911	1.514	0.511	1.514	0.563
Average	0.574	1.127	1.886	0.494	1.444	0.803

TABLE I: Soft Bubble’s contact-patch and contact-force estimation on six shapes, comparing our method vs. the FEM-based baseline [9]. Force is the ℓ_2 norm of the contact-force error.

VI. CONCLUSION AND LIMITATIONS

While our approach has shown effective in solving inverse source problems in some real-world robotics applications, we have yet to explore real-time inference capacity and representing multiple DE solutions with single networks. Interesting future work is to parameterize multiple DE solutions, similar to methods used in [13], [16], [31], and integrate the latent space inference for potential real-time applications. Additionally, while our method relies on the known governing equations of the system, an exciting direction involves replacing these known differential equations with surrogate models identified from data [32]–[34].

ACKNOWLEDGMENT

This work is supported by the NSF CAREER #2337870 and NRI #2220876 grants. The views expressed here are only of the authors and do not reflect that of the NSF. We thank the members of the Manipulation and Machine Intelligence (MMINT) Lab for their support and feedback.

REFERENCES

- [1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.
- [2] V. Isakov, "Inverse source problems," 1990. [Online]. Available: <https://api.semanticscholar.org/CorpusID:118715557>.
- [3] A. V. Smirnov, M. V. Klibanov, and L. H. Nguyen, "On an inverse source problem for the full radiative transfer equation with incomplete data," *SIAM Journal on Scientific Computing*, vol. 41, no. 5, B929–B952, 2019.
- [4] C.-S. Liu and L. Qiu, "Solving the 2d and 3d nonlinear inverse source problems of elliptic type partial differential equations by a homogenization function method," *Numerical Methods for Partial Differential Equations*, vol. 39, no. 2, pp. 1287–1298, 2023.
- [5] P. Zhang, P. Meng, W. Yin, and H. Liu, "A neural network method for time-dependent inverse source problem with limited-aperture data," *Journal of Computational and Applied Mathematics*, vol. 421, p. 114 842, 2023.
- [6] T. Fan, K. Xu, J. Pathak, and E. Darve, *Solving inverse problems in steady-state navier-stokes equations using deep neural networks*, 2020. arXiv: 2008.13074 [math.NA].
- [7] I. Depina, S. Jain, S. M. Valsson, and H. Gotovac, "Application of physics-informed neural networks to inverse problems in unsaturated groundwater flow," *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, vol. 16, pp. 21–36, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:244243653>.
- [8] Y. Chen, L. Lu, G. E. Karniadakis, and L. Dal Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials," *Optics express*, vol. 28, no. 8, pp. 11 618–11 633, 2020.
- [9] J.-C. Peng, S. Yao, and K. Hauser, "3d force and contact estimation for a soft-bubble visuotactile sensor using fem," *arXiv preprint arXiv:2310.11372*, 2023.
- [10] N. Kuppuswamy, A. Castro, C. Phillips-Grafflin, A. Alspach, and R. Tedrake, "Fast model-based contact patch and pose estimation for highly deformable dense-geometry tactile sensors," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1811–1818, 2019.
- [11] P. Sundaresan, R. Antonova, and J. Bohgl, "Diff-cloud: Real-to-sim from point clouds with differentiable simulation and rendering of deformable objects," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 10 828–10 835.
- [12] 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*, vol. 7, no. 3, pp. 5819–5826, 2022. DOI: 10.1109/LRA.2022.3157377.
- [13] M. Van der Merwe, Y. Wi, D. Berenson, and N. Fazeli, "Integrated object deformation and contact patch estimation from visuo-tactile feedback," *arXiv preprint arXiv:2305.14470*, 2023.
- [14] M. Van der Merwe, D. Berenson, and N. Fazeli, "Learning the dynamics of compliant tool-environment interaction for visuo-tactile contact servoing," in *Conference on Robot Learning*, PMLR, 2023, pp. 2052–2061.
- [15] Y. Wi, P. Florence, A. Zeng, and N. Fazeli, "Virdo: Visio-tactile implicit representations of deformable objects," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 3583–3590.
- [16] Y. Wi, A. Zeng, P. Florence, and N. Fazeli, "Virdo++: Real-world, visuo-tactile dynamics and perception of deformable objects," in *6th Annual Conference on Robot Learning*.
- [17] C. Higuera, S. Dong, B. Boots, and M. Mukadam, "Neural contact fields: Tracking extrinsic contact with tactile sensing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 12 576–12 582.
- [18] C. Higuera, J. Ortiz, H. Qi, L. Pineda, B. Boots, and M. Mukadam, "Perceiving extrinsic contacts from touch improves learning insertion policies," *arXiv preprint arXiv:2309.16652*, 2023.
- [19] Y. Narang, B. Sundaralingam, M. Macklin, A. Mousavian, and D. Fox, "Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6444–6451. DOI: 10.1109/ICRA48506.2021.9561969.
- [20] M. A. Alvarez, D. Luengo, and N. D. Lawrence, "Linear latent force models using gaussian processes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2693–2705, 2013.
- [21] S. Särkkä, M. A. Alvarez, and N. D. Lawrence, "Gaussian process latent force models for learning and stochastic control of physical systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2953–2960, 2018.
- [22] S. Wang, S. Sankaran, and P. Perdikaris, "Respecting causality is all you need for training physics-informed

- neural networks,” *arXiv preprint arXiv:2203.07404*, 2022.
- [23] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient flow pathologies in physics-informed neural networks,” *SIAM Journal on Scientific Computing*, vol. 43, no. 5, A3055–A3081, 2021.
 - [24] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, “An expert’s guide to training physics-informed neural networks,” *arXiv preprint arXiv:2308.08468*, 2023.
 - [25] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings*, 2010, pp. 249–256.
 - [26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
 - [27] A. Rashid, “Finite element modeling of contact problems,” Ph.D. dissertation, Linköping University Electronic Press, 2016.
 - [28] Y. Wi, J. Lee, M. Oller, and N. Fazeli, “Neural inverse source problem,” in *8th Annual Conference on Robot Learning*.
 - [29] T. Von Kármán, *Festigkeitsprobleme im maschinenbau*. Teubner, 1910.
 - [30] V. Makoviychuk, L. Wawrzyniak, Y. Guo, *et al.*, “Isaac gym: High performance gpu-based physics simulation for robot learning,” *arXiv preprint arXiv:2108.10470*, 2021.
 - [31] 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.
 - [32] Z. Li, N. Kovachki, K. Azizzadenesheli, *et al.*, “Fourier neural operator for parametric partial differential equations,” *arXiv preprint arXiv:2010.08895*, 2020.
 - [33] L. Lu, P. Jin, and G. E. Karniadakis, “DeepOnet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators,” *arXiv preprint arXiv:1910.03193*, 2019.
 - [34] J. Seo, “Solving real-world optimization tasks using physics-informed neural computing,” *Scientific Reports*, vol. 14, no. 1, p. 202, 2024.