Learning Object Properties Using Robot Proprioception via Differentiable Robot-Object Interaction

Peter Yichen Chen[1], Chao Liu[1], Pingchuan Ma[1], John Eastman[1],
Daniela Rus[1], Dylan Randle[2], Yuri Ivanov[2], Wojciech Matusik[1]

Abstract— Differentiable simulation has become a powerful tool for system identification. While prior work has focused on identifying robot properties using robot-specific data or object properties using object-specific data, our approach calibrates object properties by using information from the robot, without relying on data from the object itself. Specifically, we utilize robot joint encoder information, which is commonly available in standard robotic systems. Our key observation is that by analyzing the robot's reactions to manipulated objects, we can infer properties of those objects, such as inertia and softness. Leveraging this insight, we develop differentiable simulations of robot-object interactions to inversely identify the properties of the manipulated objects. Our approach relies solely on proprioception – the robot's internal sensing capabilities – and does not require external measurement tools or vision-based tracking systems. This general method is applicable to any articulated robot and requires only joint position information. We demonstrate the effectiveness of our method on a low-cost robotic platform, achieving accurate mass and elastic modulus estimations of manipulated objects with just a few seconds of computation on a laptop.

I. INTRODUCTION

When humans pick up objects, they can easily distinguish between heavy and light ones. When applying the same amount of torque at the elbow to lift an object, the elbow will rotate significantly more for a lighter object than a heavier one. Similarly, by squeezing an object, humans can assess its softness based on how much their finger joints flex. In both cases, humans rely solely on proprioceptive signals—internal sensing capabilities—to learn about the object. In this work, we ask: can robots learn about objects through interactions in the same way humans do?

Traditional methods for object parameter calibration often rely on external tools and sensors. To determine an object's softness, for example, robots might use specialized equipment like tensile testing machines [1]. While effective, this approach involves additional steps and is impractical in scenarios where direct access to the object is unavailable, such as when the object is inside a container (see Figure 1). Another common approach involves equipping robots with external sensors [2], such as cameras, to track object movement or force sensors to measure applied forces [3]. Although these sensors provide valuable data, they increase cost and are not universally available on all robots.

Our key insight is that a robot can infer object parameters by leveraging its own proprioceptive signals. During robot-object interactions, the robot's responses vary depending on the object's properties. By analyzing these responses, we can extract information about the object's characteristics, thus eliminating the need for external tools and simplifying the system identification process. The only signals required are the robot's proprioceptive data from the robot's internal sensors, such as joint encoders [4]. Therefore, this approach applies to any articulated robot without the need for advanced sensors. Similar to how humans can feel the weight of objects without visual cues, our approach also does not require vision-based supervision.

Previous approaches to object identification using proprioceptive signals often require multiple sensors, such as joint encoders and haptic sensors, necessitating sensor fusion [5], [6]. In contrast, our method relies solely on joint encoder information. A key aspect of our approach is using differentiable simulations of the robot and object dynamics, enabling us to inversely identify the object's characteristics using only a single motion trajectory.

In summary, we propose a novel framework for learning object parameters through robot-object interactions. Our contributions are threefold:

- We demonstrate that object parameters can be learned using robot proprioception in a highly data-efficient manner.
- We develop predictive simulations of robot-object interactions that are highly efficient and serve as key components in our system identification pipeline.
- We validate our approach to diverse robot-object interactions, successfully recovering both mass and elastic moduli from a single interaction trajectory.

II. RELATED WORK

System identification methods have been extensively used to calibrate the properties of both robots and manipulated objects. Traditional system identification techniques focus on identifying multiple inertial parameters of articulated robots by observing the robot's joint movements and employing linear regression [7]–[9]. More recently, researchers have explored alternatives to traditional differential-equation-based models, particularly data-driven approaches like neural networks, which serve as the underlying dynamics models for learning general nonlinear dynamics [10]–[14]. These learned models are crucial for downstream tasks such as model predictive control in dynamic manipulations [15]–[19], locomotion [20]–[22], and physical scene understanding [23]–[26]. Our work builds upon these robot system

[1]MIT CSAIL{`pyc, chaoliu, pcma, eastmanj, rus, wojciech`}`@csail.mit.edu`
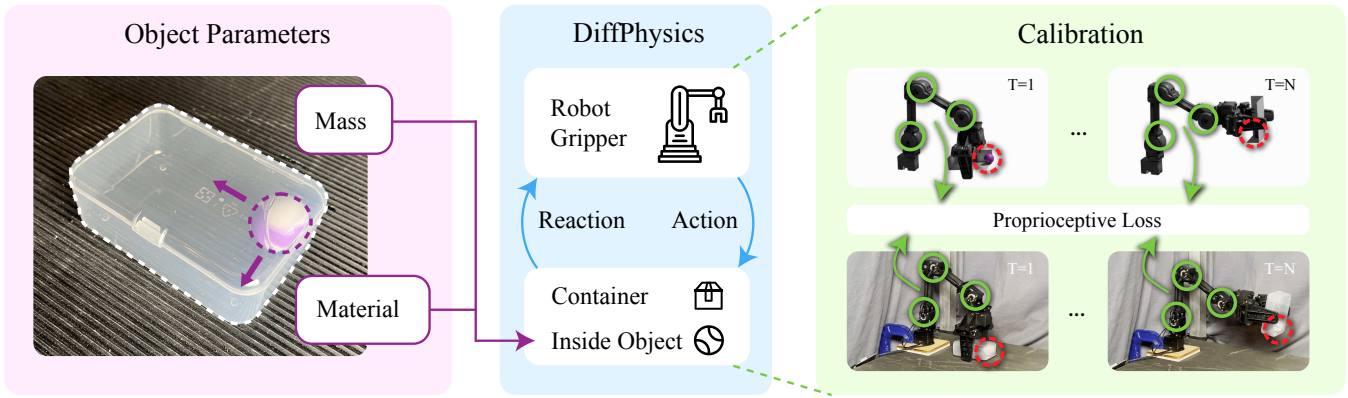[2]Amazon Robotics {`dylanran,yuriivan`}`@amazon.com`

Fig. 1. **Calibrating Object Parameters through Differentiable Physics Using Proprioceptive Signals.** Left: Our method aims to identify object parameters, such as the mass and material properties of the purple sphere. Middle: We utilize differentiable physics to simulate interactions between the robot and the object. Right: Object parameters are identified by supervising the differentiable physics simulation (top) using proprioceptive signals (joint positions, shown as green circles) from the real robot (bottom). Notably, our approach does not require tracking the object's trajectory (red circles); instead, it relies solely on the robot's internal sensors for the calibration process.

identification methods but diverges in its primary focus: rather than calibrating robot parameters, we aim to identify the properties of the object being manipulated.

Differentiable simulations have emerged as effective tools for system identification, leveraging easily available gradients to enable efficient gradient-based optimization for parameter identification. These simulations have been used to identify parameters of both rigid and soft robots by tracking the robot's movements [24], [25], [27]–[34]. Additionally, differentiable simulations have been employed to calibrate properties of objects manipulated by robots, involving complex interactions such as friction contacts and large deformations [35]–[38]. However, most of these approaches require direct access to the object's movement over time, often through vision-based tracking systems. Our approach also focuses on calibrating the properties of manipulated objects, but it does so by tracking the robot's movement over time, eliminating the need for direct observation of the object.

Proprioception plays a crucial role in human interactions with the environment, particularly when vision is insufficient [39]. Proprioceptive signals provide real-time feedback that is less susceptible to noise, can operate in various lighting conditions, and are inherently privacy-preserving. This has inspired the development of robotic actuators that perceive their own states—such as position, velocity, and force—without relying on external sensors [40]–[43]. Internal sensing capabilities have been increasingly leveraged for various purposes, including robot calibration [44], manipulated object identification [45], [46], parameter estimation [5], control [47], and manipulation of deformable objects [6]. These approaches often require multi-modal signals and sensor fusion algorithms. By contrast, our approach only utilizes proprioceptive signals from joint encoders in simple trajectories to efficiently estimate the physical properties of manipulated objects, bypassing the need for external sensing methods.

III. METHOD: DYNAMICS MODELS

This section outlines the dynamic models of both the robot and the manipulated objects, which serve as the "forward" simulation models. Section IV will describe how to use the corresponding "backward" models for parameter learning. These dynamic models take the joint torque $\boldsymbol{\tau}$ as input and predict the dynamic responses resulting from the robot-object interactions.

A. Robot

a) Equation of Motion: We model the robot's dynamics using articulated rigid body dynamics [48], [49]. The motion of these kinematic chains of multi-body systems can be described by the following ordinary differential equation (ODE):

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{f}^x) = \boldsymbol{\tau}, \qquad (1)$$

where $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$, and $\ddot{\boldsymbol{q}}(t)$ represent the joint positions, velocities, and accelerations, respectively. The matrix $\boldsymbol{H}$ is the generalized inertia matrix, and $\boldsymbol{C}$ is the bias force matrix that accounts for external forces $\boldsymbol{f}^x$, such as gravity. The term $\boldsymbol{\tau}(t)$ represents the generalized forces, such as torques applied at the joints. In the "forward" model of our experimental setup, we apply known $\boldsymbol{\tau}(t)$ to the system and aim to solve for $\boldsymbol{q}(t)$ over time.

b) Temporal Discretization: To solve Equation (1) over time, we discretize it temporally with a time step size Δt and a total of T time steps $\{t_n\}_{n=0}^T$. Using a semi-implicit Euler integration scheme [50], we arrive at:

$$\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}_n + \Delta t \ddot{\boldsymbol{q}}_{n+1}, \quad \boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta t \dot{\boldsymbol{q}}_{n+1}, \qquad (2)$$

$\forall n = 0, 1, \ldots, T-1$, where $\boldsymbol{q}_n \coloneqq \boldsymbol{q}(t_n), \dot{\boldsymbol{q}}_n \coloneqq \dot{\boldsymbol{q}}(t_n), \ddot{\boldsymbol{q}}_n \coloneqq \ddot{\boldsymbol{q}}(t_n)$. Semi-implicit time integration is a type of variational integrator that conserves energy, but it is not unconditionally stable and requires a sufficiently small time step to maintain stability. In our experiments, we simulate at 60 frames per second, using between 4 and 64 substeps to ensure accuracy and stability.

B. Object

Although we focus on articulated robots, the objects being manipulated can vary broadly and may not be limited to the dynamics described by Equation (1). Below, we outline several types of objects and their dynamic models.

a) Fixed Joint: The simplest case involves a robot holding an object modeled as a rigid body attached to the robot via an additional joint in the kinematic tree. For instance, assuming a tight grip, this additional joint can be modeled as a fixed joint (see Figure 2a). The inertial parameters of the object influence the generalized inertia matrix H in Equation (1), leading to different dynamic responses. This fixed joint model can be extended to other joint types, such as prismatic and universal joints [51]. Notably, contact and collision modeling are unnecessary as we assume the object is securely grasped by the robot.

b) Contacts and Collisions: The second case involves manipulating an object that *cannot* be modeled as part of the articulated system's kinematic tree. For example, while the robot grasps and moves a container, a small sphere inside the container is not part of the kinematic tree (see Figure 2b). The small sphere and the robot interact dynamically through contact and collision between the sphere and the container. The small sphere's rigid body dynamics are modeled separately from the robot's using its own rigid body equation of motion [48]:

$$H_o(q_o)\ddot{q}_o + C_o(q_o, \dot{q}_o, f^x{}_o) = \tau_o,$$

where the subscript "o" denotes the object's quantities.

The contact forces between the sphere and the robot's kinematic tree are modeled using penalty-based methods:

$$f_n = k_e d_n - k_d v_n,$$
$$f_t = -\min\left(k_f\|v_t\|, \mu\|f_n\|\right) \cdot \frac{v_t}{\|v_t\|},$$

where f_n and f_t represent normal and tangential contact forces, v_n and v_t are relative velocities in the normal and tangential directions, and d_n is the penetration depth. The constants k_e, k_d, k_f, μ are the normal contact stiffness, normal damping coefficient, friction stiffness, and friction coefficient, respectively [52]–[54]. These contact forces impact both the robot and the object's motion, contributing to the time-varying $\tau(t)$ and $\tau_o(t)$.

c) Deformables: While the previous cases assume rigid bodies, this scenario considers deformable bodies whose shapes change over time due to interactions and contacts with the robot (see Figure 2c). The dynamics of the deformable object are governed by the elastodynamics equation [55]:

$$\rho_0\ddot{\phi} = \nabla \cdot P + \rho_0 b, \tag{3}$$

where ϕ is the spatiotemporal deformation map, ρ_0 is the initial density, $P = \frac{\partial \Psi}{\partial F}$ is the first Piola-Kirchhoff stress, $F := \nabla\phi$ is the deformation gradient, and b represents the body force. The elastic energy Ψ is characterized by the stable Neo-Hookean model [56], [57]:

$$\Psi = \frac{k_\mu}{2}(I_C - 3) - k_\mu \log J + \frac{k_\lambda}{2}(\log J)^2, \tag{4}$$

where k_μ and k_λ are the first and second Lamé parameters, and I_C and J are invariants of the deformation gradient. We use the finite element method (FEM) to discretize and solve the elastodynamics equation [58]. In our FEM implementation, the deformable body is geometrically represented using tetrahedral elements. For objects with a rectangular shape, we divide them into multiple hexahedral cells, each of which is further decomposed into 5 tetrahedral elements.

IV. METHOD: SYSTEM IDENTIFICATION

Let θ represent the system parameters we aim to recover from the coupled dynamical system of the robot and the object (as defined in Section III). For instance, we may wish to determine the inertial parameters of a small sphere moving inside a box that is manipulated by the robot (see Figure 1 and Figure 2b). Given observations of the robot's motion over time, our goal is to learn the parameters θ by solving an inverse problem, formulated as the following optimization problem:

$$\min_{\theta} \mathcal{L}(\{q_n^{GT}\}_{n=0}^T, \{q_n\}_{n=0}^T), \tag{5}$$

where $\mathcal{L}$ is the loss function, specifically the mean squared error (MSE). The term q_n^{GT} denotes the ground truth joint positions obtained from the robot's joint encoder readings. By using a semi-implicit integration scheme as described in Equation (2), it is equivalent to supervise on joint position, velocity, or acceleration [30].

The loss function is constructed solely from the robot's kinematic quantities. In contrast, the system parameters θ that we aim to learn can encompass any aspect of the robot-object system, including the object's inertial properties and material properties such as the first and second Lamé parameters (see Section III-B), which quantify the object's softness. Fundamentally, our method relies exclusively on proprioceptive signals to calibrate the environment with which the robot interacts (see Figure 3).

We use first-order gradient-based optimization techniques to solve Equation (5). To efficiently compute the gradients with respect to the system parameters ($\frac{\partial \mathcal{L}}{\partial \theta}$), we utilize the differentiable simulation framework Nvidia Warp [59], [60]. For each "forward" dynamics model described in Section III, Warp automatically generates a corresponding "backward" adjoint model. Warp does it by leveraging the chain rule and Jacobian transpose products through built-in functions with corresponding adjoint methods. As such, Warp enables efficient gradient accumulation across long computational chains, accommodating the extensive number of time steps (Equation (2)) required for accurate simulation of robot-object interactions.

V. RESULTS

Section V-A to Section V-C validate our method across a wide range of manipulated object types, demonstrating that it can efficiently and accurately identify object parameters using the robot's proprioceptive system in real-world scenarios. Section V-D to Section V-F present ablation studies that show our method is as accurate as non-proprioception approaches
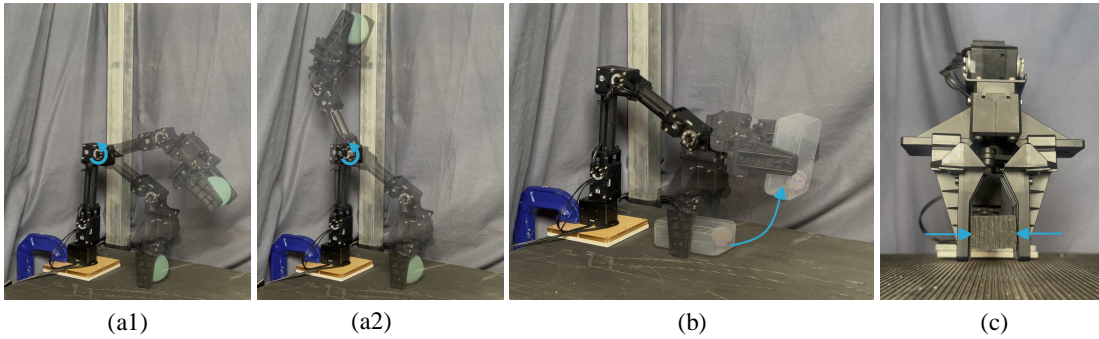
Fig. 2. **Learning Object Properties through Various Robot-object Dynamics Enabled by Joint Activations.** (a1)(a2) The object is attached to the robot as a fixed joint. The heavier ball on the left causes the robot to move less under the same joint torque. (b) The object interacts with the robot through contacts and collisions within a container. (c) The object deforms due to compression forces applied by the gripper.
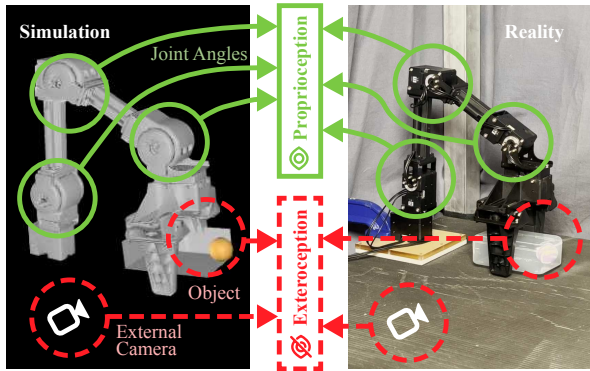

Fig. 3. **Proprioception—the robot's internal sensing capabilities.** Our approach constructs the optimization objective, i.e., Equation (5), using solely proprioceptive signals, which are directly available from the robot's internal encoders. This approach does not require exteroceptive signals, such as object motion tracking via external cameras.

that require external sensors and explicit tracking. All experiments were conducted using the Robotis OpenManipulator-X robot (see Appendix for details). The robot was operated in the current control mode, and for each experimental setup, we report the current applied to generate the torque τ. Additional optimization details are also provided in the appendix. The temporal evolutions of the dynamical system are best illustrated in the supplementary video.

A. Experiment: Fixed Joint

We aim to identify $\boldsymbol{\theta} = m_o$, where m_o is the mass of the object being manipulated. The object is held tightly by the robot's end effector and can be modeled through a fixed joint. We apply a constant current (300 mA) to generate torque τ at joint-2, applying the torque for 0.6 seconds while keeping the other joints fixed. Intuitively, under constant torque, heavier objects cause the robot to move more slowly (see Figure 2a1 and Figure 2a2). In this case, the supervision signal $\boldsymbol{q}_n^{GT}{}_{n=0}^{T}$ is the temporal evolution of joint-2's position.

Using only a single trajectory of 0.6 seconds for supervision might seem like very little data. However, due to the large number of time steps involved in constructing the loss (Equation (5)) and solving the dynamic equation

TABLE I
ACCURATE AND FAST SYSTEM IDENTIFICATION. OUR METHOD IDENTIFIES OBJECT MASS WITHIN A FEW SECONDS, WITH ERRORS RANGING FROM 0.002 TO 0.006 KG.

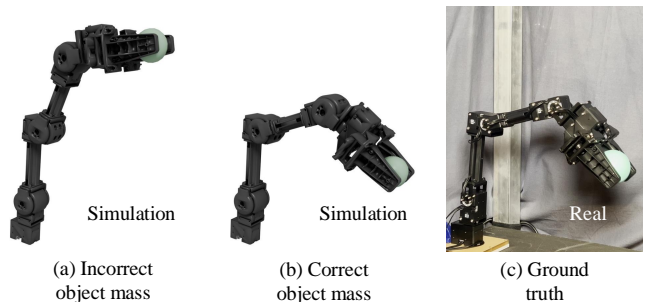|	Identified mass (kg)	Ground truth mass (kg)	Error (kg)	Computation time (s)
Heavy ball	0.056	0.050	0.006	2.696
Light ball	0.022	0.020	0.002	2.570
Heavy cube	0.020	0.023	0.003	2.546
Light cube	0.008	0.010	0.002	2.667

Fig. 4. **Closing the Sim-to-Real Gap through Accurate Object Property Identification.** With the correctly identified object mass, our simulation closely matches real-world observations.

(Equation (1)) at each time step, we effectively generate hundreds of unique and valid data points.

We tested four different objects: heavy ball, light ball, heavy cube, and light cube. Table I reports the masses identified by our algorithm. Figure 4 shows that with the correctly identified mass, the simulation closely matches the real-world robot's motion. Figure 6 demonstrates that our algorithm converges in just a few iterations, thanks to the accurate gradient computation enabled by differentiable simulation. Comparisons with gradient-free optimization will be discussed in Section V-E.

Instant calibration: Our algorithm runs on laptops (Mac-

(a) Stiff material under compression

(b) Soft material under compression

(c) Learned elastic modulus (stiff)
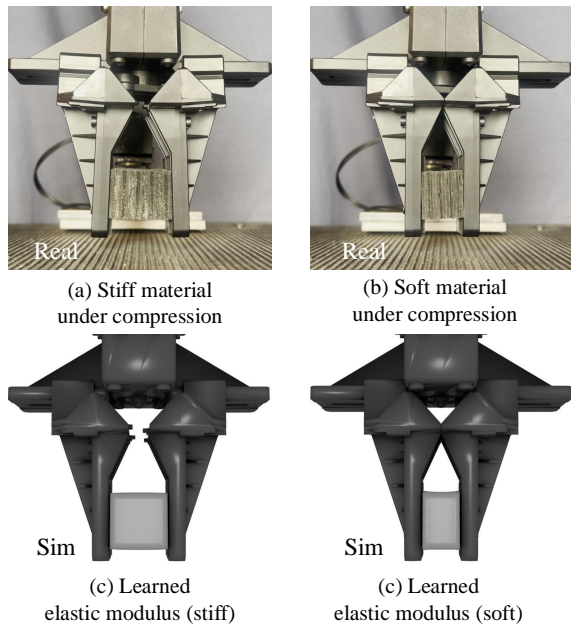
(c) Learned elastic modulus (soft)

Fig. 5. **Identifying Elastic Moduli.** Our approach identifies the elastic moduli of materials with varying stiffness. With the learned parameters, the simulation closely matches the experiments.

TABLE II

ABLATION STUDY: ROBUSTNESS UNDER DIFFERENT INITIAL SEEDS.

OUR APPROACH CONSISTENTLY CONVERGES TO ACCURATE PARAMETERS EVEN WITH INITIAL GUESSES THAT VARY BY ORDERS OF MAGNITUDE. ONLY EXTREME, UNREALISTIC INITIAL GUESSES LED TO UNDESIRABLE LOCAL MINIMA.

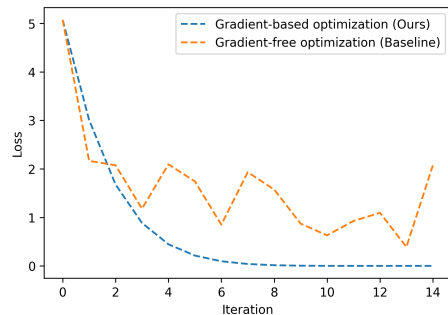| Initial guess (kg)	Converged mass (kg)	Ground truth (kg)
0.000	0.055	
0.001	0.056	
0.005	0.056	0.050
0.095	0.057	
0.191	0.058	
4.772	4.851	

Fig. 6. **Ablation Study: The Effectiveness of Gradients.** Differentiable simulation provides accurate gradients that enable gradient-based optimization, which is more efficient than gradient-free baselines.

Book Pro Apple M1 Max) and identifies the system parameters in just a few seconds (see timing in Table I).

B. Experiment: Contacts and Collisions

While the previous experimental setup provides direct access to the object under calibration, we now consider a more challenging scenario where explicit access to the object is not available. Specifically, we aim to calibrate the mass of a sphere inside a container *without* opening the container (See Figure 1), i.e., $\theta = m_s$ where m_s is the mass of the sphere.

Humans often estimate the properties of objects inside a container by shaking it [61]. Following this intuition, we use the robot gripper to shake the container holding the sphere. A constant current (100 mA) is applied to generate torque τ at joint-3 for 0.6 seconds, while the other joints remain fixed. The position of joint-3 serves as the supervision signal $\{q_n^{GT}\}_{n=0}^T$.

Under the influence of the moving container and gravity, the sphere rotates within the container, affecting the supervision signal. For contact and collision modeling, we scaled the default parameters from the Warp library to ensure stable robot simulations. Our algorithm identifies the sphere's mass as 0.018 kg, compared to a measured mass of 0.012 kg, yielding a similar error as the fixed joint experiment (see Table I).

C. Experiment: Deformable Bodies

Our model can also identify elastic moduli, specifically the first and second Lamé parameters, $\theta = k_\mu, k_\lambda$, as described in Equation (4). We use the heavy and light cubes whose masses were identified in Section V-A. The prismatic joint

at the robot's gripper is activated to compress the specimen. A constant current (300 mA) is applied to generate torque τ at joint-4 for 1.0 second, with all other joints held fixed. Intuitively, the softer the cube, the greater the compression it undergoes. The supervision signal in this case is the temporal evolution of the gripper's prismatic joint position. Similar to previous experiments, our method does not require tracking the soft cubes themselves.

Figure 5 shows the real experiment alongside the corresponding simulations. By solving Equation (5), we inversely identify the first and second Lamé parameters of the cubes. The lighter cube has smaller elastic moduli ($k_\mu = 749.6$ Pa, $k_\lambda = 264.3$ Pa) and undergoes greater deformation compared to the heavy cube ($k_\mu = 5097.6$ Pa, $k_\lambda = 5430.4$ Pa). These learned parameters allow the simulations to closely match observed behaviors (see Figure 5).

D. Ablation Study: Different Initial Seeds

To evaluate the robustness of our approach, we initialized the optimization solver (Equation (5)) with initial seeds spanning orders of magnitude differences. The task was to identify the mass of the aforementioned heavy ball. Table II shows that our method converges to the correct solution despite vastly different initial guesses. If the initial guess is extremely unrealistic (e.g., an excessively large mass), the simulation may yield divergent dynamics, causing the robotic arm to move incorrectly and trapping the optimization process in a local minimum. To mitigate this, we initialize our gradient-based algorithm with five different initial guesses
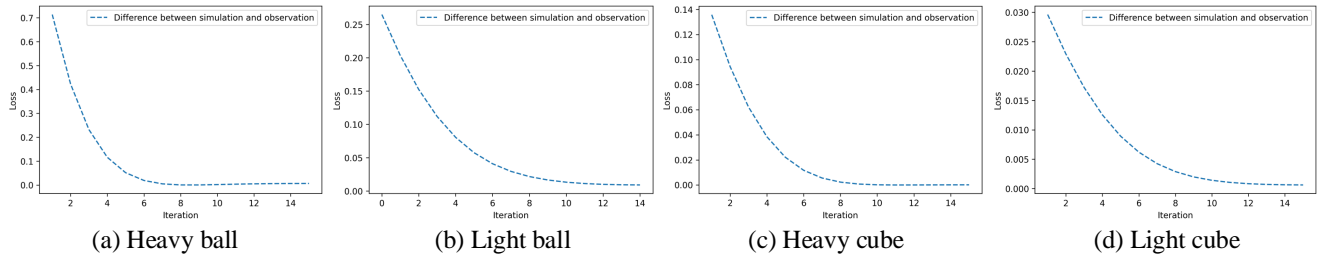
Fig. 7. **Optimization Landscape.** Our gradient-based, differentiable simulation-powered approach efficiently solves the system identification problem across various manipulated objects.

that span two orders of magnitude [62].

E. Ablation Study: Gradient-Free Optimization

In our work, we leverage differentiable simulations and gradient-based optimizers. As a comparison, gradient-free methods such as the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [63] can also be used. Figure 6 shows that our gradient-based approach converges faster, whereas gradient-free methods require more iterations and may struggle to find the global minimum.

F. Comparison: Non-Proprioception, Explicit-Tracking-Based Approaches

A key feature of our approach is that it senses the environment using only the robot's proprioceptive signals, without relying on external sensors. We compare our approach with differentiable simulation techniques that use additional supervision signals, such as from computer vision [51]. Specifically, for the sphere-container experiments (see Figure 1 and Section V-B), both the sphere positions and joint positions are used as supervision signals in $\{q_n^{GT}\}_{n=0}^{T}$. Under the same number of optimization iterations, this alternative approach identifies the sphere's mass as 0.016 kg, which is comparable to our method. Thus, compared to vision-based approaches, our method does not require explicit tracking of the object's movement while delivering similar performance.

VI. DISCUSSIONS AND CONCLUSIONS

We introduce a framework for object calibration using robot's proprioceptive sensing capabilities. Our approach leverages differentiable simulations of robot-object interactions to efficiently quantify object properties that influence the robot's responses over time. Notably, our method remains effective even in challenging scenarios where there is no direct access to the manipulated object (See Figure 1).

While we demonstrated three types of robot-object interaction modes in Section III-B, our approach can readily be generalized to handle more complex objects such as sloshing liquids and granular materials. Future extensions of this work could explore applications in other robotics systems, such as wheeled robots and soft robots [42], and incorporate probabilistic estimates of object properties [51].

Although our approach does not explicitly track object trajectories, it does assume knowledge of the objects' initial positions to set up the simulations. Future work could address this limitation by employing domain randomization techniques [37] to relax the requirement of known initial positions.

Proprioception is a fundamental way for humans to perceive and interact with the world. We envision a future where robots can also learn about their environment using proprioceptive signals. With an enhanced understanding of the world, robots can achieve more effective model predictive control, similar to how humans adjust their actions based on an object's known properties to optimize how they move and manipulate it.

APPENDIX

A. Hardware Details

Our hardware features a ROBOTIS OpenManipulator-X robotic arm powered by five high-precision DYNAMIXEL XM430-W350-T motors. These actuators have 12-bit absolute encoders for precise joint position measurements and torque control mode. Motor control is handled via the DYNAMIXEL SDK, allowing for communication at 40Hz to send commands and receive real-time motor status updates.

B. Optimizer Details

We utilize the Adam optimizer [64] combined with a cosine annealing learning rate scheduler [65]. An initial learning rate of 0.01 effectively minimizes the loss across all experiments. As shown in Figure 7, our method converges in just a few iterations.

ACKNOWLEDGMENT

References

[1] J. R. Davis, *Tensile testing*. ASM international, 2004.

[2] X. Zhang, Y. Song, Y. Yang, and H. Pan, "Stereo vision based autonomous robot calibration," *Robotics and Autonomous Systems*, vol. 93, pp. 43–51, 2017.

[3] C. Lebossé, P. Renaud, B. Bayle, and M. de Mathelin, "Modeling and evaluation of low-cost force sensors," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 815–822, 2011.

[4] P. Li and X. Liu, "Common sensors in industrial robots: A review," in *Journal of Physics: Conference Series*, vol. 1267, no. 1. IOP Publishing, 2019, p. 012036.

[5] D. Kubus, T. Kroger, and F. M. Wahl, "On-line rigid object recognition and pose estimation based on inertial parameters," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 1402–1408.

[6] M. C. Gemici and A. Saxena, "Learning haptic representation for manipulating deformable food objects," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 638–645.

[7] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.

[8] M. Gautier and W. Khalil, "A direct determination of minimum inertial parameters of robots," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 1682–1687.

[9] ——, "On the identification of the inertial parameters of robots," in *Proceedings of the 27th IEEE Conference on Decision and Control*, vol. 3. IEEE Piscataway, NJ, 1988, pp. 2264–2269.

[10] C.-M. Chow, A. G. Kuznetsov, and D. W. Clarke, "Successive one-step-ahead predictions in multiple model predictive control," *International journal of systems science*, vol. 29, no. 9, pp. 971–979, 1998.

[11] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *Proceedings of the 2004 American control conference*, vol. 3. IEEE, 2004, pp. 2214–2219.

[12] I. Lenz, R. A. Knepper, and A. Saxena, "Deepmpc: Learning deep latent features for model predictive control." in *Robotics: Science and Systems*, vol. 10. Rome, Italy, 2015, p. 25.

[13] E. Heiden, D. Millard, E. Coumans, Y. Sheng, and G. S. Sukhatme, "NeuralSim: Augmenting differentiable simulators with neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021. [Online]. Available: https://github.com/google-research/tiny-differentiable-simulator

[14] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot learning from randomized simulations: A review," *Frontiers in Robotics and AI*, vol. 9, p. 799893, 2022.

[15] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," *arXiv preprint arXiv:1810.01566*, 2018.

[16] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020.

[17] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3d neural scene representations for visuomotor control," in *Conference on Robot Learning*. PMLR, 2022, pp. 112–123.

[18] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 389–404, 2024.

[19] Z. Liu, G. Zhou, J. He, T. Marcucci, F.-F. Li, J. Wu, and Y. Li, "Model-based control with sparse neural dynamics," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[20] N. Sontakke, H. Chae, S. Lee, T. Huang, D. W. Hong, and S. Hal, "Residual physics learning and system identification for sim-to-real transfer of policies on buoyancy assisted legged robots," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 392–399.

[21] W. Yu, V. C. Kumar, G. Turk, and C. K. Liu, "Sim-to-real transfer for biped locomotion," in *2019 ieee/rsj international conference on intelligent robots and systems (iros)*. IEEE, 2019, pp. 3503–3510.

[22] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[23] Y. Li, T. Lin, K. Yi, D. Bear, D. Yamins, J. Wu, J. Tenenbaum, and A. Torralba, "Visual grounding of learned physical models," in *International conference on machine learning*. PMLR, 2020, pp. 5927–5936.

[24] D. Hahn, P. Banzet, J. M. Bern, and S. Coros, "Real2sim: Visco-elastic parameter estimation from dynamic motion," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019.

[25] Y. Hu, J. Liu, A. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik, "Chainqueen: A real-time differentiable physical simulator for soft robotics," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6265–6271.

[26] T. Du, K. Wu, P. Ma, S. Wah, A. Spielberg, D. Rus, and W. Matusik, "Diffpd: Differentiable projective dynamics," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 2, pp. 1–21, 2021.

[27] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, "End-to-end differentiable physics for learning and control," *Advances in neural information processing systems*, vol. 31, 2018.

[28] J. Degrave, M. Hermans, J. Dambre *et al.*, "A differentiable physics engine for deep learning in robotics," *Frontiers in neurorobotics*, p. 6, 2019.

[29] G. Sutanto, A. Wang, Y. Lin, M. Mukadam, G. Sukhatme, A. Rai, and F. Meier, "Encoding physical constraints in differentiable newton-euler algorithm," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 804–813.

[30] M. Lutter, J. Silberbauer, J. Watson, and J. Peters, "Differentiable physics models for real-world offline model-based reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4163–4170.

[31] Z. Huang, Y. Hu, T. Du, S. Zhou, H. Su, J. B. Tenenbaum, and C. Gan, "Plasticinelab: A soft-body manipulation benchmark with differentiable physics," *ICLR*, 2021.

[32] Y. Qiao, J. Liang, V. Koltun, and M. Lin, "Differentiable simulation of soft multi-body systems," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 123–17 135, 2021.

[33] E. Heiden, Z. Liu, V. Vineet, E. Coumans, and G. S. Sukhatme, "Inferring articulated rigid body dynamics from rgbd video," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8383–8390.

[34] M. Lutter, "A differentiable newton–euler algorithm for real-world robotics," in *Inductive Biases in Machine Learning for Robotics and Control*. Springer, 2023, pp. 9–34.

[35] Q. Le Lidec, I. Kalevatykh, I. Laptev, C. Schmid, and J. Carpentier, "Differentiable simulation for physical system identification," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3413–3420, 2021.

[36] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu, "Fast and feature-complete differentiable physics for articulated rigid bodies with contact," *arXiv preprint arXiv:2103.16021*, 2021.

[37] P. Ma, T. Du, J. B. Tenenbaum, W. Matusik, and C. Gan, "Risp: Rendering-invariant state predictor with differentiable simulation and rendering for cross-domain parameter estimation," *arXiv preprint arXiv:2205.05678*, 2022.

[38] P. Ma, P. Y. Chen, B. Deng, J. B. Tenenbaum, T. Du, C. Gan, and W. Matusik, "Learning neural constitutive laws from motion observations for generalizable pde dynamics," in *International Conference on Machine Learning*. PMLR, 2023, pp. 23 279–23 300.

[39] M. A. Heller, "Haptic dominance in form perception: Vision versus proprioception," *Perception*, vol. 21, no. 5, pp. 655–660, 1992, pMID: 1488268. [Online]. Available: https://doi.org/10.1068/p210655

[40] S. Seok, A. Wang, D. Otten, and S. Kim, "Actuator design for high force proprioceptive control in fast legged locomotion," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1970–1975.

[41] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *Ieee transactions on robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[42] J. Tapia, E. Knoop, M. Mutný, M. A. Otaduy, and M. Bächer, "Makesense: Automated sensor design for proprioceptive soft robots," *Soft robotics*, vol. 7, no. 3, pp. 332–345, 2020.

[43] D. Alatorre, D. Axinte, and A. Rabani, "Continuum robot proprioception: the ionic liquid approach," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 526–535, 2021.

[44] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *Experimen-*

tal Robotics: The 12th International Symposium on Experimental Robotics. Springer, 2014, pp. 211–225.

[45] T. Nakamura, T. Nagai, and N. Iwahashi, "Multimodal object categorization by a robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 2415–2420.

[46] J. Sinapov, T. Bergquist, C. Schenck, U. Ohiri, S. Griffith, and A. Stoytchev, "Interactive object recognition using proprioceptive and auditory feedback," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1250–1262, 2011. [Online]. Available: https://doi.org/10.1177/0278364911408368

[47] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak, "Coupling vision and proprioception for navigation of legged robots," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 273–17 283.

[48] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[49] E. Heiden, D. Millard, H. Zhang, and G. S. Sukhatme, "Interactive differentiable simulation," *arXiv preprint arXiv:1905.10706*, 2019.

[50] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, 1998.

[51] E. Heiden, C. E. Denniston, D. Millard, F. Ramos, and G. S. Sukhatme, "Probabilistic inference of simulation parameters via parallel differentiable simulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3638–3645.

[52] O. Reynolds, "Vi. on rolling-friction," *Philosophical Transactions of the Royal Society of London*, no. 166, pp. 155–174, 1876.

[53] D. C. Drucker, "Coulomb friction, plasticity, and limit loads," 1954.

[54] P. A. Cundall and O. D. Strack, "A discrete numerical model for granular assemblies," *geotechnique*, vol. 29, no. 1, pp. 47–65, 1979.

[55] O. Gonzalez and A. M. Stuart, *A first course in continuum mechanics*. Cambridge University Press, 2008, vol. 42.

[56] J. Bonet and R. D. Wood, *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[57] B. Smith, F. D. Goes, and T. Kim, "Stable neo-hookean flesh simulation," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 2, pp. 1–15, 2018.

[58] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2003.

[59] M. Macklin, "Warp: A high-performance python framework for gpu simulation and graphics," https://github.com/nvidia/warp, March 2022, nVIDIA GPU Technology Conference (GTC).

[60] ——, "Warp: Differentiable spatial computing for python," in *ACM SIGGRAPH 2024 Courses*, 2024, pp. 1–147.

[61] P. Güler, Y. Bekiroglu, X. Gratal, K. Pauwels, and D. Kragic, "What's in the container? classifying object contents from vision and touch," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3961–3968.

[62] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *International conference on machine learning*. PMLR, 2013, pp. 115–123.

[63] N. Hansen, "The cma evolution strategy: a comparing review," *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, pp. 75–102, 2006.

[64] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[65] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.