Proceedings of the ASME 2025
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE 2025
August 17-20, 2025, Anaheim, CA, USA

DETC2025/168666

REAL-TO-SIM PARAMETER LEARNING FOR DEFORMABLE PACKAGES USING HIGH-FIDELITY SIMULATORS FOR ROBOTIC MANIPULATION

Omey M. Manyar *
Realization of Robotics Systems Lab
University of Southern California
Los Angeles, CA, USA

Hantao Ye*
Realization of Robotics Systems Lab
University of Southern California
Los Angeles, CA, USA

Siddharth Mayya
Amazon Robotics
North Reading, MA, USA

Fan Wang
Amazon Robotics
North Reading, MA, USA

Satyandra K. Gupta
Realization of Robotics Systems Lab
University of Southern California
Los Angeles, CA, USA

ABSTRACT

Deformable packages are becoming increasingly prevalent in the logistics and warehouse industry, demanding robotic manipulation strategies that are robust and adaptive. Unlike rigid objects, these packages undergo significant shape changes under external forces, making their handling more complex. These deformable packages are often manipulated using suction cups and contain internal objects that shift during transport, introducing additional complexity. To ensure safe and efficient robotic manipulation at scale, high-fidelity simulation is crucial to support automated robot trajectory generation. In this work, we present a physics-driven simulation framework that accurately models multi-object deformable packages, capturing package deformations, suction-cup interactions, and internal object dynamics. Our approach employs a hyperelastic model to accurately simulate package-object interactions, including contact forces, suction-cup interactions, friction, and material deformation. By leveraging high-fidelity physics, we optimize simulation parameters using real-world trajectory and package deformation data, ensuring an accurate representation of package behavior. Running in real-time, our framework reduces real-to-sim discrepancies, making it viable for real-world deployment. Additionally, we develop a parallelized simulation environment for large-scale reinforcement learning and trajectory optimization, paving the way for scalable, efficient, and adaptable robotic manipulation of deformable packages in diverse warehouse settings. Our code, dataset, and videos are available on our project website: https://sites.google.com/usc.edu/deformable-sim/

1 Introduction

Deformable packages, such as padded mailers and polybags (Refer Fig. 1), are gaining widespread adoption in the e-commerce and logistics industry due to their flexibility, cost-effectiveness, and ability to accommodate small-sized orders. These packaging solutions are extensively used across various sectors, including fast-moving consumer goods and the textile industry, as they conform to different product shapes while optimizing storage and shipping efficiency. Fig. 1 illustrates common examples of such packages, which are designed to fulfill orders containing objects of varying sizes and masses. In warehouse operations, these packages must be picked from one location, transported, and packed into shipping containers. Currently, human workers perform these tasks in shifts, handling vast quantities of deformable

*Authors Contributed Equally, Listed Alphabetically

Copyright © 2025 by ASME

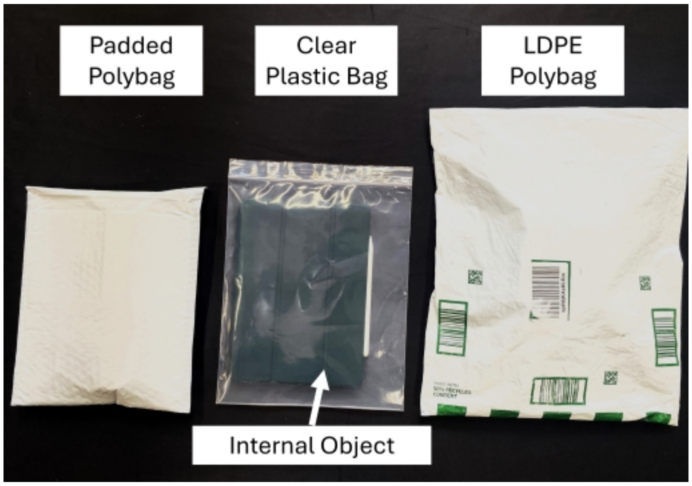


FIGURE 1: Examples of deformable closed-form packages made from various materials used in the fulfillment industry. The internal object, a tablet in this case, is enclosed within the package, demonstrating the challenge of handling such deformable structures.

packages daily. However, with increasing consumer demand and labor shortages, robotic automation has become critical for maintaining efficiency and scalability in warehouse operations.

Despite their potential, robotic systems face significant challenges in handling deformable packages due to their complex nature. In large-scale fulfillment centers, these packages are primarily handled using suction-based end effectors, as vacuum gripping is well-suited for their smooth, non-porous surfaces (Refer Fig. 2). The primary task in these environments involves picking up a package from a moving conveyor belt and placing it into a designated bin for sorting, packaging, or shipping. However, unlike rigid boxes, deformable packages undergo complex shape changes when manipulated, making traditional rigid-body suction-based grasping strategies ineffective.

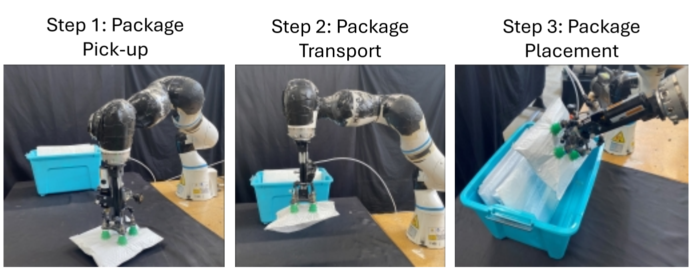


FIGURE 2: Example of a robotic cell using a suction-based tool to pick and place a deformable package into a bin, where failures can occur at any stage of the process.

Further complicating the problem, these packages often contain loosely packed internal objects that shift dynamically based on the robot's motion. This internal movement alters the package's center of mass and deformation characteristics, making it difficult to anticipate failure modes and ensure stable manipulation. As a result, robotic systems must account for the package's external deformability as well as the complex interplay between the internal object and the surrounding flexible material.

Although suction-based manipulation is effective in handling such complexities, several failure modes can still disrupt the process, leading to inefficiencies in the form of slower trajectories and the risk of potential package damage if trajectories are too fast. The three primary failure modes (Refer Fig. 3) encountered when handling deformable packages include: (1) Loading Failures: The suction cup fails to establish a secure grip, often due to insufficient suction or improper seal due to excessive surface irregularities; (2) Peeling failure: Wrinkles or folds in the package can cause premature loss of suction, resulting in the package detaching from the end effector and (3) Shear failure: Rapid movements or improper trajectory planning can lead to excessive lateral forces, breaking the suction seal and causing package drop. These failures primarily stem from the way the package deforms during manipulation and how the internal object motion further influences these deformations. The interaction between the shifting internal mass and the flexible packaging material introduces instability, making it challenging to predict and prevent failure modes.

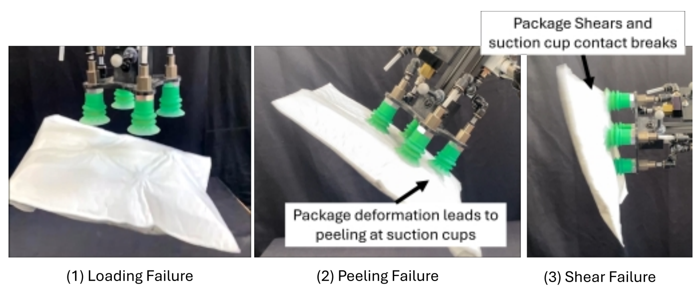


FIGURE 3: Illustration of three failure modes in package handling, primarily caused by wrinkles from package deformation and the dynamic motion of the internal object

Human operators effortlessly adapt to such variability in package shape, weight distribution, and internal motion, building an intuitive model of the deformable package-object system to manipulate it effectively. Similarly, enabling robots to autonomously handle such packages requires developing a precise model of this multi-object deformable system, a problem that remains largely unexplored in robotics. While challenging, creating such a simulation model could accelerate the large-scale

Copyright © 2025 by ASME

adoption of robotic automation for these tasks. A well-designed simulation can predict package deformations and internal object dynamics, allowing for the automated computation of optimal manipulation trajectories. These trajectories/policies must strike a balance between safety, minimizing failure risks leading to package drop, and efficiency by executing trajectories at maximum feasible speeds.

Recent advancements in learning-based manipulation have shown promise in equipping robots to handle the complexities of deformable package manipulation. These methods are particularly appealing for warehouse automation, where robots must adapt to diverse package conditions while ensuring safety and reliability. However, a major challenge of learning-based approaches is their reliance on large, high-quality datasets to provide informative inductive biases for robust policy training. Data collection in real-world warehouse environments is expensive and time-consuming due to constantly changing workspace layouts and package attributes. Even minor variations in package properties can necessitate extensive fine-tuning, further increasing the data burden. A common strategy to mitigate this challenge is training robotic policies in simulation, where data generation is scalable, controlled, and reproducible. However, to the best of our knowledge, no existing simulation framework captures both deformable package behavior and internal object dynamics, creating a significant gap in transferring learned policies to real-world settings.

Physics-based simulators are widely used for rigid-body manipulation, but modeling deformable packages, especially those with internal objects and coupled dynamics, remains challenging due to complex material properties, contact-rich interactions, and high computational costs. Soft-body simulation [1, 2] offers a promising solution, enabling high-fidelity physics modeling for efficient policy learning. Unlike conventional simulation methods, these simulators leverage structured physics priors, improving data efficiency, reducing sample complexity, and enhancing generalization. Building on these advancements and addressing the challenges of deformable package manipulation, we develop a physics-based simulation framework for multi-object deformable packages. This simulation is designed to compute optimal trajectories for safe and efficient robotic manipulation, ensuring seamless transfer to real-world conditions.

To enable efficient and realistic simulation of deformable package manipulation, we propose a hyperelastic model that captures both package deformation and internal object dynamics. Our model represents the package as a hyperelastic triangular mesh with bending and elastic properties, while the internal object exhibits interdependent motion within the deformable structure. We incorporate suction-based constraints to simulate interactions with a suction gripper, ensuring realistic manipulation dynamics. To bridge the sim-to-real gap, we collect real-world deformation data using a motion capture system for system identification, refining simulation parameters to match physical behavior. Additionally, we develop a parallelized simulation environment for scalable training of manipulation policies and trajectory optimization, enabling robots to adapt to diverse package dynamics. Our key contributions are

1. A high-fidelity simulation model of a deformable package with internal object dynamics
2. A simulation parameter learning framework that employs a structured loss to capture the nuances of the deformable package system accurately
3. A reinforcement learning gym environment for parallelized training of robotic manipulation policies in simulation

We validate our approach through real-world package transport experiments, demonstrating that our simulation model accurately captures package deformation and object dynamics. Our computationally efficient framework enables robots to adapt quickly to deformable packages in dynamic logistics settings. By integrating real-world deformation data, we enhance the development of safer and more efficient manipulation strategies. Additionally, in Section 8, we explore how this framework optimizes package handling efficiency and generates safer, more reliable trajectories for deformable object manipulation.

2 Related Works

2.1 Deformable Object Simulation

Researchers have successfully applied simulation across various deformable materials: 1D (rope) [3], 2D (fabric) [4, 5], 3D (elasto-plastic objects) [6, 7], and combinations of liquid, fabric, and elastoplastic objects [8–10]. While these works demonstrate a single-object system in simulation for various deformable materials, the specific challenges for simulating multi-object systems with constrained deformations, particularly packages with internal objects, remain underexplored. Our framework addresses this gap by developing a hyperelastic model that accurately captures the intricate dynamics of package-object interactions while maintaining computational efficiency.

Various numerical methods have been applied to the modeling of deformable objects, each with distinct advantages for particular material behaviors. The Material Point Method (MPM) has been employed for materials undergoing large deformations and topology changes [6, 11], while Position-Based Dynamics (PBD) offers computational efficiency and unconditional stability for real-time applications [3, 12]. The Finite Element Method (FEM) remains the gold standard for accurately modeling hyperelastic materials [7, 13], with implicit integration schemes providing superior stability for stiff materials. Our work employs implicit Euler integration with FEM to capture the complex interactions between deformable packages and internal objects while maintaining numerical stability.

The development of specialized software frameworks has been crucial for making simulation accessible and computation-

Copyright © 2025 by ASME

ally efficient, enabling researchers to implement complex physical models without rebuilding systems from scratch. Several notable frameworks [14–16] have emerged recently to address this need. For multi-physics simulation, GradSim [13] combines differentiable physics with differentiable rendering to jointly model scene evolutions and appearance. NVIDIA Warp [1], which we employ in our work, incorporates the computing engine from GradSim and offers a GPU-accelerated Python framework supporting both rigid and soft body simulation, with particular strengths in parallel computation and handling complex material models. We selected Warp for our implementation due to its efficient GPU utilization, flexible material modeling capabilities, and robust contact handling, which are essential for accurately simulating the interactions between deformable packages and internal objects at interactive rates.

2.2 System Identification and Policy Learning

System identification helps calibrate simulation parameters in mathematical models based on measured data. Differentiable simulation has been applied to estimate cloth stiffness and elasticity [4, 5, 13], though most studies validated parameter accuracy in simulated environments rather than real-world settings. Notable exceptions include DiffCloud [17], which utilized DiffSim [14] to perform parameter estimation for cloth in quasi-static states using real point clouds. In [18], the authors extended estimation to include cloth dynamics using point cloud data. Our work not only aims to calibrate package material parameters, but also to align internal object positions. To capture more comprehensive information, we employed motion capture systems to track the dynamics of both the package and its internal object.

Traditional simulation-based approaches for manipulation policy learning [19–22] have primarily relied on sampling-based data collection, with limited focus on multi-object systems. More recent work has explored the manipulation of elastoplastic objects [23–25] and cloth [5, 26–28], advancing simulation techniques for deformable materials. While studies have begun addressing deformable package simulation [29] and manipulation constraints [30], we extend these efforts by developing a high-fidelity multi-object simulation framework that captures real-world package interactions and enables policy learning through our high-performance simulation pipeline.

3 Problem Formulation

Consider an agent $\mathcal{A}$ manipulating a deformable thin-shell package $\mathcal{P}$. Let $\Psi_{package}$ be the set of parameters governing $\mathcal{P}$'s deformation and dynamic behavior. The package $\mathcal{P}$ contains an object $\mathcal{O}$, which is constrained to move within $\mathcal{P}$. The dynamics and motion of $\mathcal{O}$ are determined by its corresponding parameter set $\Psi_{\mathcal{O}}$. Thus, we define a multi-object system consisting of $\mathcal{P}$ and $\mathcal{O}$ manipulated by $\mathcal{A}$ and represented by the parameter sets $\{\Psi_{\mathcal{P}}, \Psi_{\mathcal{O}}\}$. This system as a whole is subjected to external forces F_{ext} and constraints C, which govern its evolution over time.

$\mathcal{P}$ is grasped by $\mathcal{A}$ via an interface $\mathcal{I}$, characterized by parameters ${}^{\mathcal{A}}\mu_{interface}$. This interface imposes constraints (C) on $\mathcal{P}$. In our case, $\mathcal{I}$ represents a suction cup interface between the robot and the package, but our formulation remains generalizable to non-suction-based interfaces. Similarly, we define ${}^{\mathcal{P}}\mu_{O}$ to describe the interface parameters governing interactions between $\mathcal{O}$ and $\mathcal{P}$. These interface parameters, i.e., μ, determine how the objects interact with each other (see Section 4.3 for details). Additionally, the agent $\mathcal{A}$ excites the package-object system by applying external control inputs U_{ext}, primarily through end-effector trajectories τ executed to manipulate $\mathcal{P}$ for a given task. Additionally, we have external forces F_{ext} that include gravitational forces acting on the system.

At a given time instance t, we define $X_t^{\mathcal{P}}$ and $X_t^{\mathcal{O}}$ as the states of the package and the object, respectively. Our objective is to learn the parameter sets Ψ and μ of a function Φ that models the behavior of the package-object system. Given an external control inputs U, forces F_{ext} and constraints C, this function should accurately predict $X_t^{\mathcal{P}}$ and $X_t^{\mathcal{O}}$. Thus, we formulate the following problem:

$$\Phi_{\Psi,\mu}(F_{ext}^t, U^t) \mapsto X_t^{\mathcal{P}}, X_t^{\mathcal{O}} \tag{1}$$

Where Φ represents the simulation model, U^t represents the control input from the commanded robot trajectory τ at the time t, and F_{ext} denotes the corresponding external forces acting on the system at the time t due to gravity and the robot's acceleration.

4 Methodology

4.1 Overview

To model the function Φ, our proposed methodology consists of four key components (Refer Fig. 4): (1) A Real-World Data Collection Stage, (2) Package-Object Representation, (3) The Simulation Framework, and (4) The Real-to-Sim Parameter Learning Framework. Our data collection process is detailed in Section 6, where we describe how data is collected in alignment with our system representation, as outlined in Section 4.2. We adopt a particle-based representation [4] for the deformable package $\mathcal{P}$, a widely used approach for deformable objects due to its ability to capture intricate deformations. Moreover, modeling a closed package containing another object introduces challenges that require careful considerations during initialization and simulation, as discussed in Section 4.2.

Our proposed framework has the potential to serve as a valuable tool for the robotic logistics industry and other applications involving deformable objects, where learning safe and efficient

Copyright © 2025 by ASME

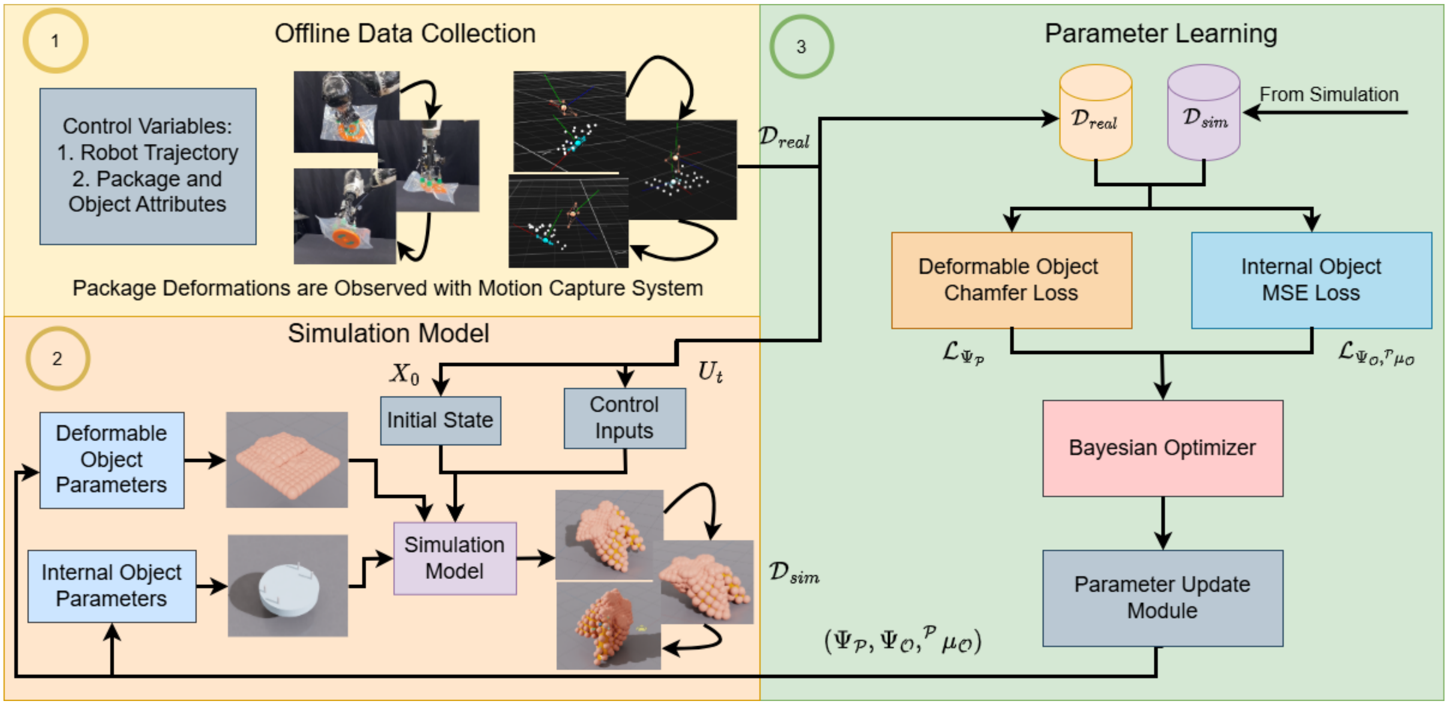


FIGURE 4: Overview of our proposed simulation parameter identification framework. As depicted, we have a real-world data collection phase, where we collect a set of robot trajectories and observe package deformation and object dynamics. Then we learn the corresponding parameters for our simulation to generate the model of our system.

policies is important. Therefore, the choice of the simulation environment is crucial, as both simulation fidelity and computational efficiency directly impact performance. Recent advancements in soft-body simulators have made them strong candidates for learning complex manipulation policies [9]. These simulators provide access to high-fidelity deformation information, enabling end-to-end learning of manipulation policies. As described in Section 4.3, we employ NVIDIA Warp as our simulation environment due to its GPU acceleration capabilities, which enhance computation efficiency in optimization while maintaining high fidelity through advanced representations of deformable objects.

A key component of our system is the Bayesian optimization-based parameter learning module, which, given real-world data, minimizes the real-to-sim gap by learning the desired parameter set Ψ of the function Φ. This is achieved by reducing the discrepancy between observed system states in the real world and their corresponding behavior in the simulation environment (Refer to Section 4.4). Through this approach, we construct a simulation framework that facilitates robotic manipulation of deformable packages containing dynamic objects. Additionally, we demonstrate how this simulation model can be leveraged to learn manipulation policies for package transport in Section 5.

4.2 System Representation

The package $\mathcal{P}$ is modeled as a two-layer hyperelastic shell structure with thickness t, discretized in a particle grid $M \times N \times 2$. This dual-layer approach effectively captures the physical structure of packaging materials with finite thickness, enabling the model to represent realistic deformation behavior of both the upper and lower surfaces during manipulation. The surface is tessellated into triangular elements, with each 2×2 particle cell subdivided into two triangles. As illustrated in Figure 5a, the first triangle is formed by connecting the bottom left particle with its right and upper neighbors, while the second triangle connects the top right particle with its left and lower neighbors. This tessellation approach ensures a balanced distribution of deformation forces across the surface. The three particles group is modeled as a triangular finite-element unit characterized by elastic stiffness $k_{tri,e}$, damping $k_{tri,d}$, and adhesion $k_{tri,a}$, which collectively form part of the parameter $\Psi_\mathcal{P}$ governing $\mathcal{P}$'s deformation. To capture out-of-plane deformation behavior, bending elements are introduced between adjacent triangular elements. The calculation of bending energy follows the formulation proposed by [31], which models the angular displacement between neighboring triangles, and the bending component is modeled as an edge and measured using elastic stiffness $k_{edge,e}$ and damping $k_{edge,d}$, which are remaining parts of the parameter set $\Psi_\mathcal{P}$. The edges connecting

Copyright © 2025 by ASME

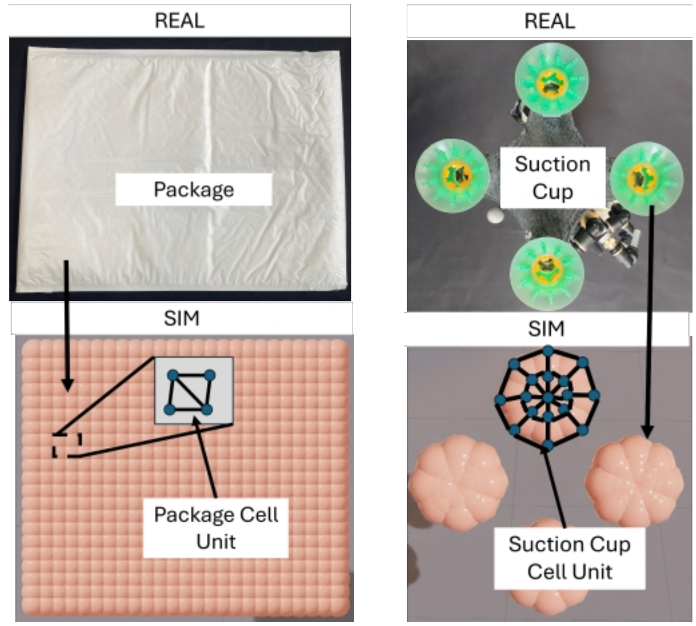
two layers also share the same structure of triangles and edges.

The interface $\mathcal{I}$ between agent $\mathcal{A}$ and package $\mathcal{P}$ consists of n suction cup units arranged in a circular configuration, where each suction cup is positioned at an angle θ_i, where $i \in \{1, 2, \ldots, n\}$, along a circle with a radius r_{cup} from the center. Each suction cup unit is modeled as a particle grid. The particle grid with dimensions $N_\theta^{cup} \times N_r^{cup}$ representing circumferential and radial discretization of the cup contact surface is shown in Figure 5b. This grid is represented as a hyperelastic shell structure parameterized by elastic stiffness $k_{cup,e}$, damping $k_{cup,d}$, and adhesion $k_{cup,a}$.

The vacuum-based adhesion phenomenon is represented by the interface parameters ${}^{\mathcal{A}}\mu_{interface}$ characterizing the interaction between the suction cups and package through a system of radially distributed springs connecting the cup's ring-shaped particles to the closest particles on the upper layer of the package. Each connection is characterized by a spring stiffness $k_{spr,e}$ and damping coefficient $k_{spr,d}$ as part of ${}^{\mathcal{A}}\mu_{interface}$. While this simplification does not explicitly model vacuum forces, it captures the essential mechanical coupling between the suction cup and package surface during manipulation.

The internal object $\mathcal{O}$ is represented at the center between the two deformable layers of $\mathcal{P}$. The interface parameters ${}^{\mathcal{P}}\mu_O$ describe the contact between $\mathcal{O}$ and the package layers, defined as soft contact characterized by contact stiffness $k_{con,e}$, damping $k_{con,d}$, and friction coefficient μ_{con}. This representation accounts for the challenges of modeling a closed package containing internal objects, particularly how these contents influence deformation behavior during manipulation.

This system representation forms the foundation for our simulation framework described in Section 4.3, enabling the implementation of the function $\Phi_{\Psi,\mu}$ that maps external forces F_{ext}^t and control inputs U^t to the states of the package $X_t^{\mathcal{P}}$ and object $X_t^{\mathcal{O}}$



(a) Package particle cell unit

(b) Suction cup unit

FIGURE 5: System Representation Overview: Each element of the package and suction cup is modeled using particles connected by edges to simulate elasticity and bending. Additionally, particle radius accounts for adhesion and compression in the deformable components.

4.3 Simulation Framework

Simulating a complex multi-object system with deformable components, as in this work, presents unique challenges. The model must accurately capture the nuanced deformations of a two-layered package under manipulation constraints, external forces, and gravity while also handling contact dynamics and suction-based coupling. In logistics and warehouse automation, where speed and efficiency are critical, high-speed simulation is essential for seamless deployment. This demands a high-fidelity framework that supports coupled multi-object interactions with diverse material properties. High-fidelity physics information enables safe trajectory optimization and policy learning, ensuring fast, safe, and reliable package handling. Additionally, the simulation must balance fidelity and computational efficiency to enable precise system identification and execution in practical robotics applications.

Soft-body simulators offer a crucial advantage over traditional physics engines by addressing key challenges in modeling complex deformable interactions. They achieve this through two main capabilities: (1) they enable efficient optimization by providing structured representations of system dynamics, allowing for more precise policy learning, trajectory optimization, and system identification compared to sampling-based and finite-difference approaches. (2) they leverage parallel computing architectures to achieve high-speed simulations, sometimes running significantly faster than real-time for simpler systems, even in scenarios involving intricate contact dynamics and deformable objects.

When evaluating potential simulation frameworks for our application, we considered several critical factors as mentioned above: high-fidelity physics, material model flexibility, computational efficiency, and extensibility for implementing custom physics for specialized components like suction cups. After comparing several options including DiffTaichi [15], MJX [32], and Nvidia Warp [1]. We selected Warp as our simulation framework due to its comprehensive fulfillment of these requirements.

While DiffTaichi offers flexibility for modeling dynamical systems, it lacks essential features such as URDF import and kinematic chain generation, requiring extensive custom implementation compared to general-purpose physics engines. Sim-

Copyright © 2025 by ASME

ilarly, MJX struggles with handling large numbers of contacts efficiently, as it precomputes all potential contacts rather than resolving them dynamically. In contrast, NVIDIA Warp leverages parallel computation to accelerate simulation while providing built-in physical models and integrators that streamline development. Its support for FEM techniques makes it well-suited for our package-object system, and its interoperability with PyTorch facilitates seamless integration with learning-based frameworks. By distributing particle updates, constraint solving, and contact resolution across GPU cores, Warp enables high-speed simulation, achieving rates exceeding 60 Hz even for complex systems with over 1,000 particles.

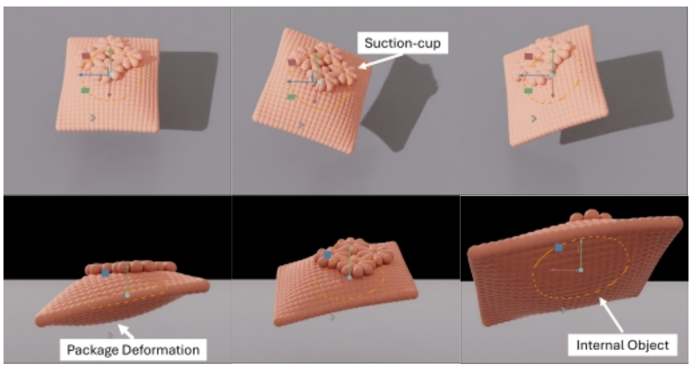


FIGURE 6: The simulation of the deformable package, suction cup, and internal object in our proposed simulation environment

Our implementation (Refer Fig. 6) leverages several key components of the Warp framework to realize the system representation described in Section 4.2. We use Warp's particle system to represent the vertices of both the package's two-layer structure and the suction cup array, constructing triangular meshes according to our earlier definitions. The force computation for hyperelastic materials and rigid bodies is handled by Warp's built-in semi-implicit Euler integration scheme. For actuation, trajectory control inputs are converted to timestamped velocity commands for the suction cups, which we implement through a customized kernel function.

Warp's GPU-accelerated computation, efficient contact handling, and soft-body physical modeling capabilities make it an ideal framework for simulating and optimizing our complex package-object system. Its ability to efficiently model large-scale deformable interactions enables accurate system identification and rapid policy learning, overcoming the computational bottlenecks of traditional simulation methods. This approach directly addresses the challenges of robotic manipulation in logistics, ensuring both high-fidelity simulation and scalable deployment.

4.4 Real-to-Sim Parameter Identification

The package ($\mathcal{P}$), Object ($\mathcal{O}$), and the suction cup system are characterized by a set of parameters, (Ψ, μ), as described in Section 4.2, that governs specific aspects of the system's response to external forces and constraints. For instance, the parameters $k_{trie,e}$ and $k_{tri,d}$ primarily influence the deformation behavior of the package, whereas $k_{con,e}$ predominantly affects the motion of the object $\mathcal{O}$ inside the package. Recognizing these distinctions is crucial, as it informs the design of our learning framework for parameter identification. Leveraging this understanding, we develop a method to infer these parameters from real-world data (see Section 6).

Now, given that our simulation model represents the deformable package $\mathcal{P}$ as a system of particles with grid size $M \times N \times 2$, it is crucial to collect real-world data that provides congruent information for learning the simulation parameters. Additionally, tracking the position of the internal object $\mathcal{O}$ is essential. Section 6 details our experimental setup and data-collection strategy to obtain this information. Specifically, we employ a motion capture system with reflective markers placed on the package, the object, and the suction tool to record timestamped trajectory data. Consequently, our observation set $\mathbf{O}$ consists of timestamped position data of the package particles, the object, and the robot end-effector, i.e., the suction tool's position and orientation. Notably, these markers are required only during the training phase; during execution, our model can infer this information seamlessly. Thus, no markers or motion capture system are needed for actual robot deployment (see Section 6).

From our data collection framework, we receive access to a dataset of timestamped trajectories $\mathcal{D}_{real} = \{\tau_1^{real}, \tau_2^{real}, \ldots.., \tau_n^{real}\}$. Where, $\tau_n^{real} = \{(X_0^{real}, U_0^{real}), (X_1^{real}, U_1^{real}), \ldots., (X_T^{real}, U_T^{real})\}$ is a timestamped trajectory consisting of information of system state X_t and the corresponding control input U_t at a given time instance t. Here $X_t = \{X_t^{\mathcal{P}}, X_t^{\mathcal{O}}\}$ is the state of package-object system. The control inputs U_t are the robot's end-effector position, orientation, velocity, and acceleration parameters at a given time t. Now, given this dataset $\mathcal{D}_{real}$, we use our simulation framework described in Sections. 4.2 and 4.3, where we initialize $\mathcal{P}$ and $\mathcal{O}$ with the corresponding initial states and constraints. During initialization, we can only control U_o, i.e., the control input, and the constraints, i.e., the contact between the suction cup and the package. Under the external forces, the package-object system assumes a state X_0^{sim}. We then simulate the entire trajectory τ_i^{real} in our simulation environment and observe the corresponding states at the timestamps $t = 0 : T$. This process constructs a corresponding dataset of simulated trajectories, $\mathcal{D}_{sim}$, which we use to learn the simulation parameters Ψ. Our parameter estimation framework is detailed in Algorithm 1, where we employ the Bayesian optimization [33] framework to optimize Ψ.

As discussed in Sections 4.1 and 4.2, each parameter in the

 Copyright © 2025 by ASME

Algorithm 1: Learning Simulation Parameters via Bayesian Optimization

```
Data: Real-world Trajectory Dataset: D_real
Input: N : Number of Optimization Iterations,
Φ: Deformable Simulation Model,
F_ext: External Force (Gravity),
G: Gaussian Process Surrogate Model
Result: Ψ*_P, Ψ*_O, ^P μ*_O: Optimized Simulation
        Parameters
Initialize: Initialize search space for Ψ_P, Ψ_O, ^P μ_O
Fit initial Gaussian Process G using random samples
i = 0
while i ≤ N do
    for τ^real_batch ∈ D_real do
        Sample candidate parameters Ψ_P, Ψ_O, ^P μ_O
          from acquisition function
        X^sim_batch = Φ(Ψ_P, Ψ_O, ^P μ_O, U^real_batch)
        L^batch_{Ψ_P} = CD(X^real_{P_batch} − X^sim_{P_batch})
        L^batch_{Ψ_O, ^P μ_O} = ||X^real_{O_batch} − X^sim_{O_batch}||_2^2
        L_batch = L^batch_{Ψ_P} + L^batch_{Ψ_O, ^P μ_O}
        Update G with new observation
          (Ψ_P, Ψ_O, ^P μ_O, L_batch)
    end
    Select the next parameters by optimizing the
      acquisition function over G
    if L_batch < L_min then
        L_min = L_batch
        Ψ*_P ← Ψ_P
        Ψ*_O ← Ψ_O
        ^P μ*_O ← ^P μ_O
    end
    i++
end
```

full parameter set Ψ contributes uniquely to the system's dynamics. To ensure our optimization framework captures this structured influence, we design distinct loss functions that appropriately guide the optimization process. Specifically, for optimizing the package parameters, we employ the Chamfer loss [34], defined as follows:

$$\mathcal{L}_{\Psi_\mathcal{P}} = \frac{1}{2|\mathcal{X}_\mathcal{P}^{real}|} \sum_{x_i^{real} \in \mathcal{X}_\mathcal{P}^{real}} min||x_i^{real} - \mathcal{X}_\mathcal{P}^{sim}||_2 + \frac{1}{2|\mathcal{X}_\mathcal{P}^{sim}|} \sum_{x_j^{sim} \in \mathcal{X}_\mathcal{P}^{sim}} min||x_j^{sim} - \mathcal{X}_\mathcal{P}^{real}||_2 \quad (2)$$

Chamfer loss is well-suited for training because it directly minimizes the discrepancy between simulated and real particle positions, ensuring that the simulated package deformation closely matches real-world behavior. By penalizing the distance between corresponding particle sets in a trajectory pair $(\tau_i^{sim}, \tau_i^{real})$, Chamfer loss enforces fine-grained spatial alignment without requiring explicit point correspondences. This makes it particularly effective for capturing complex deformations, as it naturally handles variations in particle distribution while maintaining robustness to minor noise in the dataset.

Similarly, for optimizing the object parameters $\Psi_\mathcal{O}$ and the interface parameters ${}^\mathcal{P}\mu_\mathcal{O}$, we utilize the Mean Square Error (MSE) loss function to effectively guide the optimization of these parameters:

$$\mathcal{L}_{\Psi_\mathcal{O}, {}^\mathcal{P}\mu_\mathcal{O}} = ||X_\mathcal{O}^{real} - X_\mathcal{O}^{sim}||_2^2 \quad (3)$$

We specifically rely on $\mathcal{O}$'s state observations for optimizing the interface parameters, as our experiments indicate that $\mathcal{O}$'s position is primarily influenced by these parameters, which govern contact forces and friction. Thus, our optimization process jointly refines the simulation parameters to enhance accuracy. The overall loss function, $\mathcal{L}_\mathcal{D}$ as defined in the Algorithm. 1 is the weighted sum of the individual loss terms in Eqs. 2 and 3, ensuring that each set of parameters is optimized accordingly. This structured approach allows us to learn simulation parameters that accurately reproduce the dynamics of our multi-object system. Furthermore, as outlined in Algorithm 1, we optimize the parameters using the Bayesian optimization framework over N epochs. Bayesian optimization excels in efficiently exploring the parameter space, especially when the objective function is complex, noisy, or expensive to evaluate. A key component of this framework is the acquisition function, which guides the search for optimal parameters by balancing exploration and exploitation. The acquisition function evaluates the utility of sampling a particular set of parameters based on the current probabilistic model of the objective function. By selecting the parameters that maximize the acquisition function, we ensure that each iteration improves our understanding of the parameter space and helps identify regions with high performance.

In our approach, the acquisition function helps us prioritize promising regions for parameter updates, ensuring that we avoid redundant evaluations and focus computational resources on the most informative areas. This leads to stable convergence with fewer iterations, making the optimization process computationally efficient while maintaining high-quality results. The combination of the probabilistic model and the acquisition function makes Bayesian optimization particularly powerful for parameter estimation tasks, enabling robust and reliable learning even with limited data.

Copyright © 2025 by ASME

5 Manipulation Policy Learning

The proposed simulation model is designed to support both online planning and offline learning of safe and efficient manipulation policies. Prior work [35] has demonstrated the importance of simulators in enhancing the learning of manipulation policies, particularly for complex robotic tasks, by providing valuable inductive biases. Our framework leverages these biases to enable both data-driven policy learning and model-based trajectory optimization. The ability to capture high-frame-rate deformation information further enhances the model's applicability to online planning, offering a powerful solution for tackling dynamic manipulation tasks with precision and efficiency.

One of the standout features of our approach is the use of NVIDIA Warp, a CUDA-based framework optimized for GPU computation. Massive parallelization is essential for reinforcement learning (RL) applications, as training requires extensive exploration across diverse scenarios. By leveraging GPU acceleration, we simulate multiple training instances simultaneously, drastically reducing policy optimization time. This level of parallelism is crucial for large-scale learning, where sample efficiency and computational speed are fundamental constraints.

In high-mix, high-volume environments such as logistics and warehouse automation, where speed and efficiency are crucial, this parallelism marks a significant advancement in policy learning. The ability to train and refine policies at scale enables the rapid deployment of learned behaviors in real-world applications. While the primary focus of this work is on our simulation model and parameter identification framework, we also highlight the practical applicability of our differentiable simulation system by implementing an RL Gym [36] environment. This open platform enables the robotics community to train their own RL-based manipulation policies efficiently and in a scalable manner. Additionally, our simulator facilitates model-predictive control (MPC), enabling the optimization of robot trajectories for safe and efficient motion generation. This capability is crucial for handling deformable objects in real-world applications, ensuring that the resulting motions adhere to safety constraints while maximizing efficiency.

As a demonstration, we train a Proximal Policy Optimization (PPO)-based policy for package transport. The primary objective of this policy is to minimize the internal object movement within the package and reduce its deformation curvature near the suction cup to ensure stable grasping. This example highlights how our simulator enables large-scale policy learning while maintaining physically meaningful constraints that are crucial for real-world deployment.

By integrating high-fidelity physics with reinforcement learning and model-based control, our approach bridges the gap between soft-body simulation and practical robotic policy deployment. The ability to train and optimize policies at scale makes it a valuable tool for advancing deformable object manipulation in industrial settings.

6 Experiments and Data Collection

6.1 Experimental Design

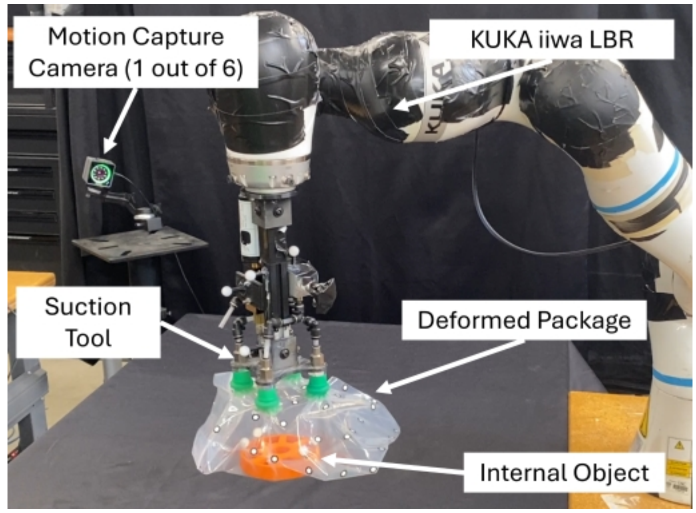


FIGURE 7: The experimental setup with a 7 DOF KUKA LBR iiwa Robot. A suction tool is attached to the robot that is connected to a 9 CFM vacuum pump. The suction cups selected in this task are rated to handle such objects specifically.

Our experimental setup consists of a 7-DOF KUKA LBR iiwa robot, chosen for its impedance control capabilities, which facilitate the handling of deformable packages during trials. The robot is equipped with a four-suction cup tool connected to a 9 CFM vacuum pump with a 1.5-micron ultimate pressure via a manifold (see Fig. 7). Suction control is handled by a relay operated through an Arduino microcontroller. To systematically study the effects of robot motion on both the package and its internal object, we designed a trajectory that captures a range of dynamic movements relevant to real-world logistical settings. The trajectory simulates a pick-and-place operation, where a package is lifted from a conveyor belt and placed into a bin. The final placement orientation varies from horizontal (0^o) to near-vertical, with three distinct angles: 30^o (low), 45^o (medium), and 60^o (high). The trajectory consists of three primary motions: a linear motion followed by a circular motion and an orientation change at the final placement stage. To further analyze motion effects, we vary joint velocity and acceleration, categorizing each into three levels: low, medium, and high. Each of these trajectory variations reflects the changes in U_{real}. The test package measures 10×12 inches and contains a 3D-printed internal disc of 160 mm diameter. The internal mass is also varied across three levels: 0.5 lbs (low), 0.75 lbs (medium), and 1 lbs (high). We collect a total of 27 distinct timestamped trajectory data points corresponding to 1 package, 3 placement orientations, 3 mass variations, and 3 ve-

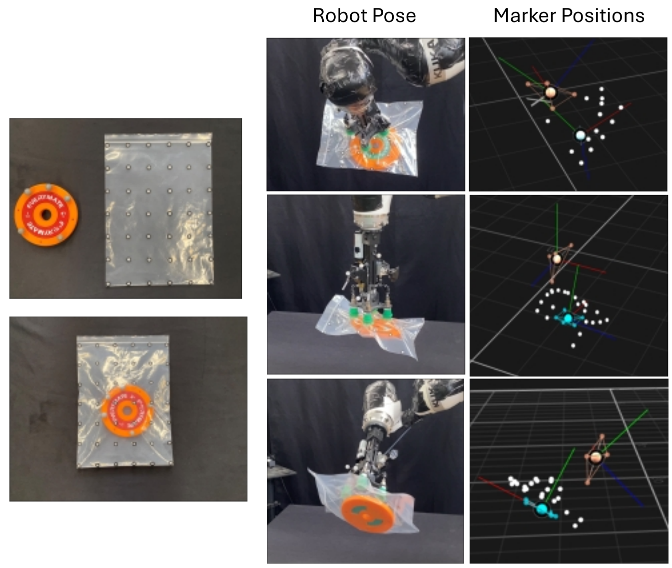
Copyright © 2025 by ASME

locity conditions. Each trajectory lasts an average of 6 seconds, and we neglected the failure scenarios since the focus was mainly on the dynamics. Our data acquisition framework operates at 250 Hz, providing high-resolution motion data that captures detailed package deformation over time. Despite the limited number of 24 trajectories after filtering the failure scenarios, the high temporal resolution of the data ensures sufficient detail for learning the parameters effectively.

A key component of our data collection framework is a 6-camera OptiTrack motion capture system, which is primarily used to track the deformation of the package. OptiTrack's Motive 2 software streams data from these cameras, allowing us to track the positions of individual reflective markers or defined rigid bodies and compute their corresponding $\mathbb{SE}(3)$ poses with a submillimeter accuracy in tracking. As discussed in Section 4.2, our simulation represents the package $\mathcal{P}$ as a particle-based system. To emulate this in the real world, we affix N 2D reflective markers to the package, effectively creating a particle-based representation. The (x,y,z) positions of these markers are tracked in Motive 2, providing high-fidelity deformation data. Thus, the state of the package at a time instance t is represented by the position of these markers $X_{\mathcal{P}} = [x_i, y_i, z_i]_{i=1}^{N}$. To ensure accurate tracking of both the package and its internal object, we make a design choice to use transparent packages. This allows us to track the internal object without additional engineering efforts for time synchronization. We acknowledge that real-world logistics settings typically involve non-transparent packages. However, our primary objective in this work is to establish a proof of concept demonstrating that our simulation framework can handle such systems. For non-transparent packages, an alternative approach involves using an external force-torque sensor to estimate the center of mass (CoM) of the internal object. This would require additional engineering, particularly for time synchronization and ensuring reliable CoM estimation. In this study, we opt for transparent packages to simplify data collection while maintaining the visibility of the internal object. For tracking the internal object, we use spherical reflective markers and define it as a rigid body in Motive 2 (see Fig. 8). This setup allows us to capture both position and orientation relative to the motion capture system's base frame, ensuring accurate and consistent tracking. For simplicity, we only consider the Cartesian position of the object, and thus $X_{\mathcal{O}} = [x_{\mathcal{O}}, y_{\mathcal{O}}, z_{\mathcal{O}}]$.

6.2 Processing of Motion Capture Data

The timestamped data collected from the motion capture system is inherently noisy, primarily due to marker occlusions and the presence of spurious reflections from other reflective surfaces in the camera's field of view. These spurious reflections can result in erroneous marker detections, which we mitigate through precise calibration. Specifically, during data collection, we execute predefined trajectories and manually mask out any spurious markers that appear. This ensures that reflective surfaces incorrectly tagged as markers are eliminated from the dataset. To further improve robustness, we enforce a minimum difference of 10 mm between the inter-marker distance of the 2D markers on the package and the spherical markers on the internal object. This prevents marker misidentification, where the system might erroneously associate one marker's identity with another.



(a) The Package and Object with reflective markers (b) The visualization of markers in Motive 2 Software

FIGURE 8: The package and the corresponding inside object with the motion capture markers. We also illustrate a sample trajectory that is collected from the motion capture. We can observe that at certain time instances, the package markers can disappear, thus entailing the adoption of advanced data-processing methods.

A key challenge in post-processing arises due to marker occlusion, where markers temporarily disappear from the tracking system. Since OptiTrack's Motive 2 software does not provide marker identification and tracking across trajectory for deformable bodies,[1] we introduce a comprehensive filtering methodology to estimate missing marker positions:

1. Noise Reduction: We first apply a low-pass Butterworth filter to the timestamped marker trajectories to eliminate high-frequency noise $X_f(t) = \frac{1}{\sqrt{1+\left(\frac{f_c}{f}\right)^{2n}}} X(t)$. where $X_f(t)$ is the filtered signal, $X(t)$ is the raw marker position, f_c is the cut-off frequency, and n is the filter order. The band gain is set to 1.
2. Hungarian Matching-Based Reassignment: If a marker dis-

[1] This functionality is available in Motive 3, but for the scale of deformations in our setup, it remains intractable.

Copyright © 2025 by ASME

appears, we check whether it has been reassigned a new ID by using the Hungarian algorithm for optimal matching. Specifically, we solve for the best match between the missing marker p_i and the detected markers at the time t, $\mathcal{M}(t)$, based on their available neighbor information. The assignment cost is minimized by:$\hat{p}_i = \arg\min_{p_j \in \mathcal{M}(t)} \|p_j - \hat{p}_i^{(t)}\|$, where $\hat{p}_i^{(t)}$ represents the set of neighbor positions associated with the missing marker at the current time step. If the cost is below a predefined threshold ε, the marker is reassigned based on the optimal matching result. If no neighbors are available, we discard the position information by assigning it a zero mask
3. Velocity-Based Interpolation: If reassignment fails, we estimate the missing position using first-order derivative-based interpolation under the assumption of local smoothness: $p_i(t) = p_i(t-1) + v_i(t-1) \cdot \Delta t$, where, $v_i(t-1) = \frac{p_i(t-1) - p_i(t-2)}{\Delta t}$ is the estimated velocity of the marker.
4. 1D Interpolation for Long Occlusions: If data is missing for a longer window, we perform 1D linear interpolation using adjacent marker positions, defined by $p_i(t) = p_i(t_1) + \frac{t-t_1}{t_2-t_1}(p_i(t_2) - p_i(t_1))$. where t_1 and t_2 are the nearest available timestamps.

If none of the above methods successfully reconstruct the missing marker positions, we discard the affected timestamps from our dataset. Ultimately, our filtering strategy enables us to retain 98% of the timestamped data from the collected dataset for downstream parameter learning while ensuring high-fidelity deformation tracking.

7 Results

In our experiments, we aimed to evaluate key aspects of our parameter learning framework to assess its effectiveness. Specifically, we sought to answer the following questions:

1. How accurately does our simulation framework predict real-world package deformation and internal object dynamics?
2. To what extent does our method generalize to variations in control parameters and object mass?
3. How well does our proposed simulation model perform on unseen scenarios in a held-out test dataset?

With these evaluations, we aim to provide a comprehensive assessment of our simulation framework's fidelity, robustness, and generalization capabilities, highlighting its potential for real-world deployment in robotic manipulation tasks.

7.1 Parameter Estimation Results

The dataset of 24 trajectories was preprocessed using our data-processing pipeline, as described in Section 6.2. We employed a subset sampling strategy to divide the data into training and testing sets to ensure a fair and representative evaluation. Specifically, for the test set, we aimed to assess the generalization of our method on unseen cases. To achieve this, we selected trajectories with medium velocity, medium acceleration, and medium orientation for a given package and mass combination. These conditions fall within the overall distribution of the training data but are not explicitly included, allowing us to fairly evaluate generalization. Based on this strategy, we allocated four trajectories for testing. This resulted in a final split of 20 trajectories for training and 4 for testing. All data was normalized using a min-max normalization approach before training. We implemented the Bayesian optimization framework with the Bayes-opt library [37]. We trained our model for 100 epochs and reported the best-performing parameter values on the training data in Table 1.

Dataset	Object			Package		
	MSE ↓			CD ↓		
	mean	max	std	mean	max	std
Train	0.0006	0.003	0.0006	0.032	0.036	0.003
Test	0.0008	0.001	0.0003	0.033	0.038	0.003

TABLE 1: The performance of the parameter estimation framework. The error values reported here are the chamfer distance of the package state and the mean-square error in object state prediction in meters.

Table 1 highlights the effectiveness of our parameter estimation framework in accurately predicting both the package and internal object positions. The error values reported confirm that our simulation model effectively captures the intricate interactions between the deformable package, the internal object, and the external constraints imposed by the robotic manipulator. The chamfer distance estimation for the package loss, which is 0.033 m, demonstrates that our model successfully captures the package deformation [38], especially for such a complex dynamic deformable object. Additionally, the mean square error of 0.0008 for the internal object shows that the interface contact and object parameters were appropriately learned. These results validate the learned parameters' accuracy and fidelity, suggesting our approach generalizes well to various control inputs and package configurations (Refer Fig. 9).

7.2 Optimization Performance Analysis

A key premise of our framework is that accurately modeling the deformable package system requires selecting optimal simulation parameters. If the loss landscape were relatively flat, ran-

Copyright © 2025 by ASME

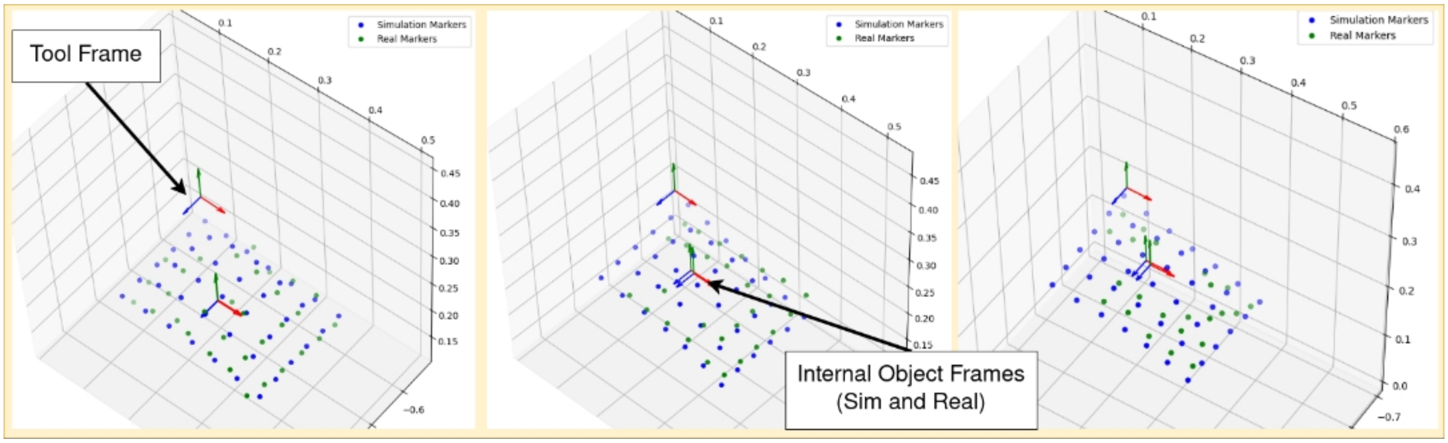


FIGURE 9: Qualitative comparison of simulation vs. real-world predictions. Blue markers represent simulated positions at a given time step, while green markers indicate corresponding real-world positions from our test dataset. The coordinate axes near the markers denote simulated and real object positions, with the topmost axis representing the tool frame.

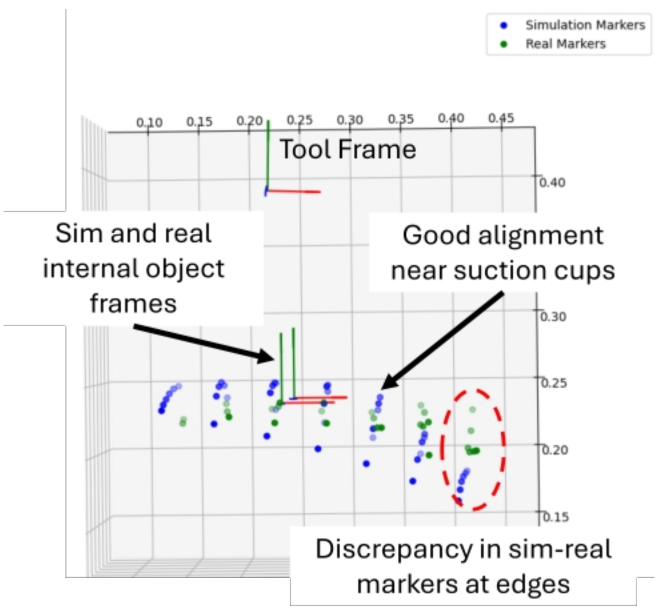


FIGURE 10: The marker prediction discrepancies are primarily at the edges, while accuracy remains high near the suction cups, a critical region for failure detection. This demonstrates that our parameter estimation is well-suited for computing safe and efficient trajectories, as most errors do not impact performance.

domly chosen parameters would yield similar results, reducing the need for optimization. However, in practice, random initialization often leads to poor predictions, as the loss landscape contains local minima that significantly impact performance. Our Bayesian optimizer effectively identifies an optimal set of parameters that enables accurate prediction of the package and internal object states, which are critical for learning safe and efficient manipulation policies.

To validate this, we perturb each parameter by a percentage of its optimized value and analyze its effect on prediction accuracy. Our sensitivity analysis shows that perturbations within 20% of the optimized value maintain stable performance with minimal impact on accuracy. However, beyond this threshold, the simulation becomes unstable, requiring reduced time steps for integration, which significantly slows down the overall simulation speed.

To validate that our prediction accuracy is sufficient for computing safe and efficient manipulation policies and trajectories, we analyze the simulation's performance. As shown in Fig. 10, the primary discrepancies in marker position predictions occur at the package edges, while markers near the suction cup engagement points, closer to the package center, maintain high accuracy (< 0.01m). This demonstrates that despite a chamfer distance of 0.03m, our simulation model remains effective for generating reliable and safe trajectories.

7.3 Simulation Performance Metrics

A critical aspect of our simulation framework is evaluating its computational efficiency in handling complex multi-object interactions. To ensure high fidelity while maintaining acceptable prediction accuracy, we systematically vary the number of particles representing the package and analyze its impact on both prediction performance and simulation frame rate. As shown in Table 2, increasing the particle count results in a comparable simulation frame rate but introduces a trade-off between computational efficiency and accuracy. Our findings suggest that practitioners should fine-tune the particle count based on their specific accuracy and performance requirements. Another key con-

Copyright © 2025 by ASME

sideration is hardware memory limitations. All our evaluations are conducted on an NVIDIA GeForce RTX 3060 12 GB GPU, where instantiating more than 1612 particles leads to memory overload. Furthermore, Fig. 11 demonstrates that reducing the particle count can lead to inaccurate package deformation, ultimately degrading prediction performance, which aligns with our ablation studies.

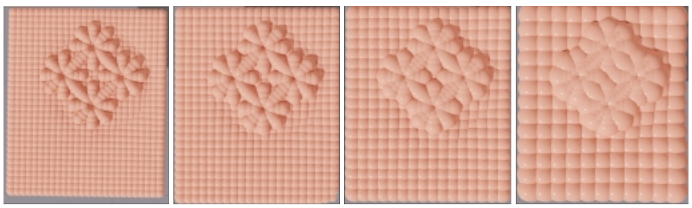


FIGURE 11: Difference in package shape in simulation with reducing number of particles. We can see that lesser particles can lead to poor performance in capturing the package deformation

Num Particles	Frame Rate (FPS) ↑	Package CD ↓	Object MSE ↓
1612	9	0.032	0.0006
1050	10	0.033	0.0007
608	10	0.035	0.0008
286	10	0.037	0.001

TABLE 2: Effect of particle count on simulation parameters and frame rate. The prediction performance depends on the number of particles. Our simulation runs in real-time, meaning it takes t seconds to simulate a trajectory of duration t seconds. We can see increasing num particles improves the loss without affecting the FPS, however, FPS remains the same. Increasing particles can cause issues with available GPU memory.

8 Discussions

While our primary focus in this work is on developing a deformable simulation model of the package-object system, its broader impact on package handling efficiency is crucial to consider. The proposed model significantly enhances the balance between safety and efficiency across key manipulation stages: pick-up, transport, and placement. Each of these steps demands precise trajectory computation to minimize failures that could result in package damage. In our prior [30], we explored trajectory optimization for fully filled packages, but it did not capture the complexities introduced by partially filled deformable packages with a moving object. In this work, our proposed multi-object model addresses this gap by providing a more adaptable and accurate simulation framework, enabling robots to handle a wider range of package conditions with improved reliability and efficiency in computation that translates to process efficiency.

Our GPU-based simulation framework implementation enables more efficient training times, with an observed 30% reduction compared to non-GPU-based models. By accelerating trajectory optimization and improving convergence speed, our approach reduces the number of iterations required to find an optimal solution. This computational improvement can potentially decrease overall process time by 15% [30]. Furthermore, the ability to explicitly model failure cases [29] enhances robustness, with a potential failure reduction of 20%. Internal object movement within the package plays a key role in suction-based grasp failures, and our approach allows for precise simulation and tracking of these dynamics. By optimizing trajectories to minimize internal object motion, we reduce instability and the risk of package damage. Unlike conventional rigid-body simulators, our approach dynamically adapts to variations in package shape, weight, and contents, allowing robots to respond to disturbances up to 2-5 times faster, ensuring more reliable real-world deployment without extensive manual tuning.

Although our simulation framework is designed to be differentiable, maintaining smooth and stable gradients within the NVIDIA Warp framework proves challenging. This difficulty arises from the inherent complexities of simulating deformable objects, where small numerical or algorithmic inconsistencies can introduce noisy gradients, particularly when accounting for non-linearities, contact dynamics, and material deformations. In high-dimensional, dynamic environments, where the relationships between simulation parameters and physical behaviors are intricate, the gradients may become unstable or imprecise. Such noisy gradients can degrade the performance of gradient-based optimization methods, leading to suboptimal solutions or slower convergence due to erratic updates.

In contrast, Bayesian optimization is particularly well-suited for these types of scenarios. Unlike gradient-based methods that depend on precise gradient information, Bayesian optimization treats the objective function as a black box and constructs a probabilistic model. This allows it to explore the parameter space effectively without relying on gradients, making it more resilient to noisy or unreliable signals. By incorporating prior knowledge and carefully balancing exploration with exploitation, Bayesian optimization can navigate the parameter space efficiently, even in the presence of noise or limited data. Furthermore, since Bayesian optimization updates its model iteratively based on fewer but more informative evaluations, it mitigates the impact of noisy gradients, leading to more stable optimization. Given these advantages, we have opted to rely on Bayesian optimization rather than exploiting the differentiability of our simulation.

 Copyright © 2025 by ASME

9 Conclusions

In this work, we introduced a high-fidelity deformable object simulation framework that captures the complex dynamics of a deformable package with an internally moving object. Our GPU-accelerated simulation provides accurate physics data for training manipulation policies, enabling more effective learning for package handling tasks. We proposed a Bayesian optimization approach to learn simulation parameters from real-world deformation data and developed a structured loss formulation to enhance parameter estimation. Additionally, we designed a comprehensive data collection testbed for parameter learning and created a Gym environment for training manipulation policies. We demonstrated the real-world applicability and fidelity of our simulation.

Although our framework shows promise for online applications, real-time performance for control remains a challenge, and failure cases such as suction cup bending were not explicitly modeled. In future work, we plan to address these limitations by incorporating solutions for suction-induced failures and optimizing real-time performance. We also aim to further explore the differentiability of our simulation framework by implementing custom gradient handling to overcome non-differentiability issues, thus advancing the use of differentiable simulations in robotic manipulation tasks.

Acknowledgment

This work is supported in part by Amazon Robotics. The opinions expressed are those of the authors and do not necessarily reflect the opinions of the sponsors.

References

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

[2] Hu, Y., Li, T.-M., Anderson, L., Ragan-Kelley, J., and Durand, F., 2019. "Taichi: a language for high-performance computation on spatially sparse data structures". *ACM Transactions on Graphics (TOG),* ***38***(6), p. 201.

[3] Liu, F., Su, E., Lu, J., Li, M., and Yip, M. C., 2023. "Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics". *IEEE Robotics and Automation Letters,* ***8***(7), pp. 3964–3971.

[4] Liang, J., Lin, M., and Koltun, V., 2019. "Differentiable cloth simulation for inverse problems". In Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc.

[5] Li, Y., Du, T., Wu, K., Xu, J., and Matusik, W., 2022. "Diffcloth: Differentiable cloth simulation with dry frictional contact". *ACM Trans. Graph.,* ***42***(1), oct.

[6] Huang, Z., Hu, Y., Du, T., Zhou, S., Su, H., Tenenbaum, J. B., and Gan, C., 2021. "Plasticinelab: A soft-body manipulation benchmark with differentiable physics". In International Conference on Learning Representations.

[7] Heiden, E., Macklin, M., Narang, Y. S., Fox, D., Garg, A., and Ramos, F., 2021. "DiSECt: A Differentiable Simulation Engine for Autonomous Robotic Cutting". In Proceedings of Robotics: Science and Systems.

[8] Lin, X., Wang, Y., Olkin, J., and Held, D., 2020. "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation". In Conference on Robot Learning.

[9] Chen, S., Xu, Y., Yu, C., Li, L., Ma, X., Xu, Z., and Hsu, D., 2023. "Daxbench: Benchmarking deformable object manipulation with differentiable physics". In The Eleventh International Conference on Learning Representations.

[10] Liu, M., Yang, G., Luo, S., and Shao, L., 2024. "Softmac: Differentiable soft body simulation with forecast-based contact model and two-way coupling with articulated rigid bodies and clothes". In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 12008–12015.

[11] Hu, Y., Fang, Y., Ge, Z., Qu, Z., Zhu, Y., Pradhana, A., and Jiang, C., 2018. "A moving least squares material point method with displacement discontinuity and two-way rigid body coupling". *ACM Transactions on Graphics (TOG),* ***37***(4), pp. 1–14.

[12] Macklin, M., Müller, M., and Chentanez, N., 2016. "Xpbd: position-based simulation of compliant constrained dynamics". In Proceedings of the 9th International Conference on Motion in Games, MIG '16, Association for Computing Machinery, p. 49–54.

[13] Murthy, J. K., Macklin, M., Golemo, F., Voleti, V., Petrini, L., Weiss, M., Considine, B., Parent-Lévesque, J., Xie, K., Erleben, K., Paull, L., Shkurti, F., Nowrouzezahrai, D., and Fidler, S., 2021. "gradsim: Differentiable simulation for system identification and visuomotor control". In International Conference on Learning Representations.

[14] Qiao, Y.-L., Liang, J., Koltun, V., and Lin, M., 2020. "Scalable differentiable physics for learning and control". In Proceedings of the 37th International Conference on Machine Learning, Vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 7847–7856.

[15] Hu, Y., Anderson, L., Li, T.-M., Sun, Q., Carr, N., Ragan-Kelley, J., and Durand, F., 2020. "Difftaichi: Differentiable programming for physical simulation". In International Conference on Learning Representations.

[16] Freeman, C. D., Frey, E., Raichuk, A., Girgin, S., Mordatch, I., and Bachem, O., 2021. "Brax - a differentiable physics engine for large scale rigid body simulation". In Thirty-fifth Conference on Neural Information Processing

Copyright © 2025 by ASME

Systems Datasets and Benchmarks Track (Round 1).
[17] Sundaresan, P., Antonova, R., and Bohgl, J., 2022. "Diffcloud: 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), pp. 10828–10835.
[18] Blanco-Mulero, D., Barbany, O., Alcan, G., Colomé, A., Torras, C., and Kyrki, V., 2024. "Benchmarking the sim-to-real gap in cloth manipulation". *IEEE Robotics and Automation Letters,* ***9***(3), pp. 2981–2988.
[19] Yan, W., Vangipuram, A., Abbeel, P., and Pinto, L., 2021. "Learning predictive representations for deformable objects using contrastive estimation". In Conference on Robot Learning, PMLR, pp. 564–574.
[20] Manyar, O. M., Desai, J., Deogaonkar, N., Joesph, R. J., Malhan, R., McNulty, Z., Wang, B., Barbič, J., and Gupta, S. K., 2021. "A simulation-based grasp planner for enabling robotic grasping during composite sheet layup". In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 930–937.
[21] Seita, D., Florence, P., Tompson, J., Coumans, E., Sindhwani, V., Goldberg, K., and Zeng, A., 2021. "Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks". In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 4568–4575.
[22] Manyar, O. M., Cheng, J., Levine, R., Krishnan, V., Barbič, J., and Gupta, S. K., 2022. "Physics informed synthetic image generation for deep learning-based detection of wrinkles and folds". *Journal of Computing and Information Science in Engineering,* ***23***(3), 12, p. 030903.
[23] Chen, S., Liu, Y., Yao, S. W., Li, J., Fan, T., and Pan, J., 2022. "Diffsrl: Learning dynamical state representation for deformable object manipulation with differentiable simulation". *IEEE Robotics and Automation Letters,* ***7***(4), pp. 9533–9540.
[24] Qi, C., Lin, X., and Held, D., 2022. "Learning closed-loop dough manipulation using a differentiable reset module". *IEEE Robotics and Automation Letters,* ***7***(4), pp. 9857–9864.
[25] Shi, H., Xu, H., Huang, Z., Li, Y., and Wu, J., 2024. "Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks". *The International Journal of Robotics Research,* ***43***(4), pp. 533–549.
[26] Kuroki, S., Guo, J., Matsushima, T., Okubo, T., Kobayashi, M., Ikeda, Y., Takanami, R., Yoo, P., Matsuo, Y., and Iwasawa, Y., 2024. "Gendom: Generalizable one-shot deformable object manipulation with parameter-aware policy". In 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 14792–14799.
[27] Zhang, Y., Liu, F., Liang, X., and Yip, M., 2024. "Achieving autonomous cloth manipulation with optimal control via differentiable physics-aware regularization and safety constraints". In 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 9931–9938.
[28] Chen, Y.-W., Joseph, R. J., Kanyuck, A., Khan, S., Malhan, R. K., Manyar, O. M., McNulty, Z., Wang, B., Barbič, J., and Gupta, S. K., 2022. "A digital twin for automated layup of prepreg composite sheets". *Journal of Manufacturing Science and Engineering,* ***144***(4), p. 041010.
[29] Manyar, O. M., Ye, H., Sagare, M., Mayya, S., Wang, F., and Gupta, S. K., 2024. "Simulation-assisted learning for efficient bin-packing of deformable packages in a bimanual robotic cell". In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 5211–5218.
[30] Shukla, R., Yu, Z., Moode, S., Manyar, O. M., Wang, F., Mayya, S., and Gupta, S. K., 2024. "Performing efficient and safe deformable package transport operations using suction cups". In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 11835–11842.
[31] Bridson, R., Fedkiw, R., and Anderson, J., 2002. "Robust treatment of collisions, contact and friction for cloth animation". *ACM Transactions on Graphics (TOG),* ***21***(3), pp. 594–603.
[32] Deepmind, 2023. Mujoco XLA (MJX). `https://mujoco.readthedocs.io/en/stable/mjx.html`. [Accessed 03-03-2025].
[33] Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., and Deng, S.-H., 2019. "Hyperparameter optimization for machine learning models based on bayesian optimization". *Journal of Electronic Science and Technology,* ***17***(1), pp. 26–40.
[34] Huang, T., Liu, Q., Zhao, X., Chen, J., and Liu, Y., 2024. "Learnable chamfer distance for point cloud reconstruction". *Pattern Recognition Letters,* ***178***, pp. 43–48.
[35] Srinivasan, K., Heiden, E., Ng, I., Bohg, J., and Garg, A., 2024. "Dexmots: Learning contact-rich dexterous manipulation in an object-centric task space with differentiable simulation". In International Symposium of Robotics Research (ISRR).
[36] Makoviichuk, D., and Makoviychuk, V., 2021. rl-games: A high-performance framework for reinforcement learning. `https://github.com/Denys88/rl_games`, May.
[37] Nogueira, F., 2014. Bayesian Optimization: Open source constrained global optimization tool for Python.
[38] Wu, T., Pan, L., Zhang, J., Wang, T., Liu, Z., and Lin, D., 2021. "Density-aware chamfer distance as a comprehensive metric for point cloud completion". In Proceedings of the 35th International Conference on Neural Information Processing Systems, pp. 29088–29100.

 Copyright © 2025 by ASME