Contact-Aware Neural Dynamics

Changwei Jing[1], Jai Krishna Bandi[1], Jianglong Ye[1], Yan Duan[2],
Pieter Abbeel[2], Xiaolong Wang[1†], Sha Yi[1†]

[1]UC San Diego, [2]Amazon FAR (Frontier AI & Robotics), [†]Equal advising

https://changwei-jing.github.io/neural-physics/

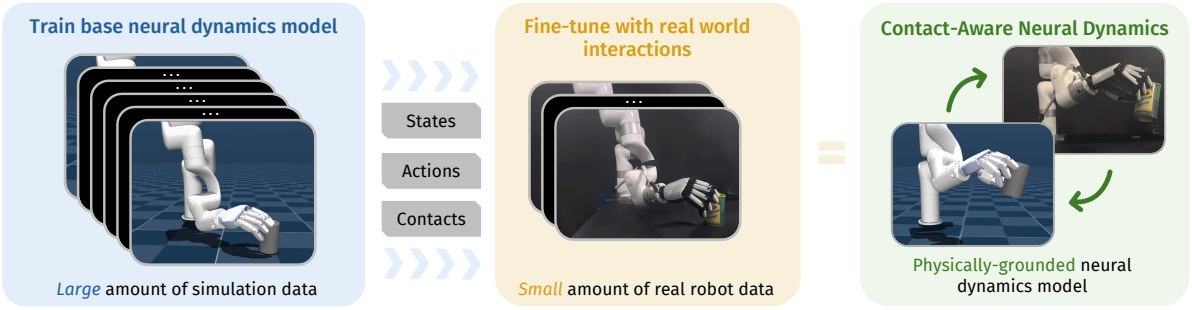

Figure 1. We present an implicit sim-to-real alignment framework for robot manipulation that uses contact information from tactile sensors. We first train a base neural dynamics model using large-scale simulation data, allowing the network to learn contact-induced physical behaviors from histories of states, actions, and contact information. We then fine-tune the model with a small amount of real-world interaction data, including tactile information, enabling it to better capture real contact patterns and dynamics in contact-rich manipulation tasks. This yields a **contact-aware neural dynamics** model that performs consistently across both simulation and the real world.

Abstract

High-fidelity physics simulation is essential for scalable robotic learning, but the sim-to-real gap persists, especially for tasks involving complex, dynamic, and discontinuous interactions like physical contacts. Explicit system identification, which tunes explicit simulator parameters, is often insufficient to align the intricate, high-dimensional, and state-dependent dynamics of the real world. To overcome this, we propose an implicit sim-to-real alignment framework that learns to directly align the simulator's dynamics with contact information. Our method treats the off-the-shelf simulator as a base prior and learns a contact-aware neural dynamics model to refine simulated states using real-world observations. We show that using tactile contact information from robotic hands can effectively model the non-smooth discontinuities inherent in contact-rich tasks, resulting in a neural dynamics model grounded by real-world data. We demonstrate that this learned forward dynamics model improves state prediction accuracy and can be effectively used to predict policy performance and refine policies trained purely in standard simulators, offering a scalable, data-driven approach to sim-to-real alignment.

1. Introduction

Although recent advances in teleoperation and human-in-the-loop data collection have enabled impressive robot demonstrations of dexterous manipulation, simulation-based approaches for training and evaluation provide significantly greater scalability and diversity. Learning purely from real-world teleoperation is expensive, time-consuming, difficult to iterate, and limited to only human intuition. This makes accurate physics simulation crucial for developing generalizable manipulation policies. However, the gap between simulated and real-world dynamics is still big, particularly for contact-rich manipulation tasks. Policies trained in simulation have transferred successfully to a range of locomotion problems, where dynamics are only needed for robots and a static ground, so that modeling errors are often forgiving. Directly deploying sim-trained policies for in-hand or contact-rich manipulation is far less reliable. Small discrepancies in contact geometry, frictional modeling, compliance, or simulation integration timing can drastically change object motion and stability, making standard rigid-body simulators insufficient for faithfully capturing the delicate dynamics for dexterous manipulation.

A dominant way to mitigate this gap is *explicit* system identification: improving geometric parameters [1, 9], tun-

ing friction and mass [36], or optimizing a small set of physical parameters so that simulated rollouts match real trajectories [8]. However, this approach fundamentally assumes that a low-dimensional parametric correction is good enough, which is not sufficient for contact-rich manipulation tasks. Many sources of error are high-dimensional (complex contact formulation involving damping and restitution), state-dependent, or simply due to the discrete integration of the simulation. Attempts to compensate purely via domain randomization or parameter sweeps often trade accuracy for robustness and still fail to capture the non-smooth transitions [34] that arise from contact. In parallel, vision-focused sim-to-real techniques, such as rendering [24, 38] and aggressive domain randomization [18], primarily close the perception gap, enabling robust visual policies while leaving the underlying contact dynamics model largely unchanged.

In parallel, recent work has explored *implicit* alignment via learning. These methods include neural residual models atop analytical dynamics [15, 49], differentiable simulators with learned components [20], and neural dynamics models trained on simulation [44] or real-world [51] data. However, models trained on purely simulation data often lack transferability, while those using only real data are inefficient to collect. Many of these approaches treat the simulator as a prior and use a neural network to model the residual errors. A significant limitation is that these methods are often contact-agnostic, i.e., treating discontinuities as noise, or rely solely on kinematic and proprioceptive signals. Consequently, they under-utilize one of the richest signals available in manipulation: contact information itself. High-bandwidth tactile sensing, in contrast, provides a fast and responsive signal to guide the modeling process and future dynamics predictions.

In this work, we propose *contact-aware neural dynamics*: an implicit sim-to-real alignment framework that leverages contact information in both simulation and the real world. We first train a neural forward dynamics model in simulation using large-scale rollouts of a dexterous hand interacting with diverse objects under extensive domain randomization. The model conditions on the robot and object states, robot actions, and rendered contact information. This neural dynamics model learns to predict multi-step rollouts for both successful and failed manipulation trajectories. We then collect corresponding real-world trajectories, again including both successes and failures, augmented with tactile sensor readings, and fine-tune the simulation-only model with real-world data. By co-training rather than fitting a separate correction stage, the learned dynamics implicitly align simulated and real-world states in a shared representation based on contact events. This gives a contact-aware forward model that utilizes the diversity and efficiency in simulation while inheriting fidelity from real-world robot data, enabling more accurate robot policy behavior in contact-rich manipulation.

2. Related Work

Bridging the sim-to-real gap remains central in robotic learning. Simulators enable scalable and safe data collection but diverge from reality due to mismatches in interaction dynamics. Domain randomization perturbs simulation parameters such as masses and friction to account for the sim2real mismatch [3, 35, 40], yet small errors in contact parameters often yield unrealistic dynamics. Complementary approaches either attempt to tune simulator parameters using data collected in the real world [7, 32, 47] or infer latent physical variables online for policy conditioning [48]. Recent real-to-sim pipelines generate physics-aware assets or photorealistic reconstructions for policy transfer [13, 18, 36]. Broader world-model frameworks integrate generative video modeling with physical scene evolution for control [2, 45, 53]. Despite progress, most sim2real methods treat contact indirectly—by randomizing few parameters or adapting rigid-body models that struggle with stiff, discontinuous interactions [34]. Our work instead learns a dynamics model grounded in tactile feedback, bridging the sim-to-real gap without the need for explicit parameter tuning.

System Identification. System identification estimates parameters linking simulation and reality. For manipulation, robot dynamics are usually known, leaving object properties as the main uncertainty. Classical methods recover mass or inertia from excitation trajectories or force–torque sensing [26, 28], but rely on precise calibration. Automated pipelines such as [36] and active exploration frameworks [27] use robot interaction to infer geometry and inertial parameters, while vision-language models estimate material properties from appearance [50]. Yet these approaches depend on user-specified parameter sets and often fail to generalize when unmodeled effects (compliance or anisotropic friction) dominate.

Vision-Based Methods. Vision-based dynamics models aim to infer physics directly from images or videos. Early methods estimated static object properties such as mass or volume from RGB-D data [30, 39], but many quantities (e.g., stiffness or friction) remain visually unobservable. Deep video-prediction approaches forecast future frames from actions [14, 33], and object-centric architectures such as [6] model pairwise interactions. Latent world models like [16, 17] use compact visual dynamics for long-horizon control, while [12] applies such models to real manipulation. More recent works leverage large-scale video and video-language models for planning or imitation [4, 10, 11, 22, 23]. Although visually coherent, these models often lack physical grounding and predicted trajectories often violate contact realism under occlusion or multi-

object interaction. Physics-aware visual frameworks such as [18, 45] highlight the value of coupling perception with physics; our work follows this principle but grounds learning in tactile rather than visual signals.

Neural Dynamics and World Models. Neural simulators learn forward dynamics directly from data, offering differentiable, adaptable alternatives to analytical physics replacing analytical contact solvers with learned modules for articulated bodies [44]. Material-conditioned approaches such as [31, 51] generalize across deformable materials, and particle- or point-cloud world models such as [21] capture multi-object interactions. Hybrid schemes such as [51] combine particle and grid representations to model deformable-object behavior, while [25] reconstructs geometry and physical properties from sparse videos. Large-scale embodied world models such as [2, 53] merge generative vision and physics, pointing toward unified physical AI generation. However, most rely on full-state or visual supervision and remain weakly grounded in real contact physics. Our approach complements these efforts by learning a tactile-aware dynamics model that aligns simulated trajectories with real contact behavior.

3. Method

We formulate sim-to-real alignment for contact-rich manipulation as a *conditional dynamics prediction* problem, where we learn to model contact-dependent object motion using multimodal observations from simulation and reality. At each time step t, the state of the object is defined by its pose $\mathbf{s}_t \in \mathrm{SE}(3)$, represented with translation and rotation. The robot hand joint configuration is denoted as $\mathbf{q}_t \in \mathbb{R}^{d_q}$, and it interacts with the object with its action $\mathbf{a}_t \in \mathbb{R}^{d_q}$. We model the contact between robot hand and object with a binary indication as $c_t \in \{0, 1\}$. $c_t = 1$ if *any* fingertip is in contact with the object, and zero otherwise. The object geometry is represented by point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$. The point cloud is obtained by sampling the object mesh surface in the beginning, and transformed by the object pose throughout the entire trajectory. We introduce four more definitions as follows:

History window. The input to our neural dynamics model is a sequence of past states and actions. We use a fixed-length observation history over $K+1$ past steps:

$$\mathcal{H}_t = \left\{ \mathbf{s}_{t-K:t},\ \mathbf{a}_{t-K:t},\ \mathbf{q}_{t-K:t},\ c_{t-K:t},\ \mathcal{P} \right\}, \quad (1)$$

which provides geometric, kinematic, control, and contact information for predicting future motion. To improve generalization to real-world dynamics from simulation data, we generate trajectories with domain randomization. This includes adding small Gaussian noise to each control command as well as perturbing object and hand poses at random intervals, ensuring that the model is exposed to diverse motion patterns and is robust to actuation noise and state estimation uncertainty.

Prediction goal. Given the history $\mathcal{H}_t$, the neural dynamics model aims to predict a future horizon of hand–object contacts and state trajectories:

$$\begin{aligned} \hat{c}_{t+1:t+H} &= f_\phi(\mathcal{H}_t), \\ \Delta\hat{\mathbf{s}}_{t+1:t+H} &= g_\theta(\mathcal{H}_t,\ \hat{c}_{t+1:t+H}). \end{aligned} \quad (2)$$

Here, f_ϕ denotes the contact-prediction module that infers future binary contact events from the historical observations, while g_θ denotes the state-prediction module that generates future pose trajectories conditioned on both the history and the predicted contact sequence.

Contact representation. The contact information in the simulation and real-world may differ greatly, especially when the tactile sensors on the robot hardware are inaccurate and noisy. For simplicity and robustness, we use a *binary, hand-level contact* signal $c_t \in \{0, 1\}$. The contact predictor is supervised using a binary cross-entropy (BCE) objective, and the predicted probabilities are re-encoded as a low-dimensional contact feature that conditions the pose dynamics, ensuring contact-aware motion generation.

Neural dynamics representation. We represent the system at time t by the object pose $\mathbf{s}_t$, a fixed-length observation history $\mathcal{H}_t$, and the object point cloud $\mathcal{P}$. The history $\mathcal{H}_t$ contains object poses $\mathbf{s}_{t-K:t}$, robot actions $\mathbf{a}_{t-K:t}$, joint values $\mathbf{q}_{t-K:t}$, and binary contact states $c_{t-K:t}$. In the next section, we introduce the framework of using this formulation to train a contact-aware neural dynamics model to automatically align simulation and real-world distribution for contact-rich robot manipulation tasks.

3.1. Model Framework and Architecture

Our model consists of two coupled modules: (i) a *Contact Predictor* module that predicts the future contact probabilities $\hat{c}_{t+1:t+H}$, and (ii) a *Diffusion Pose Predictor* that generates future pose differences conditioned on the given state and action history, and the predicted contact information.

Multimodal encoders and fusion. As shown in Fig. 2, the temporal history information, including the object pose $\mathbf{s}_{t-K:t}$, action $\mathbf{a}_{t-K:t}$, and robot joint configuration $\mathbf{q}_{t-K:t}$, is stacked together along the history dimension as inputs to the neural network. The contact sequences $c_{t-K:t}$ are encoded by an individual module. The static object point cloud $\mathcal{P}$ is processed by a PointNet encoder [37], yielding a geometry embedding $\mathbf{f}_\mathcal{P}$. All modality embeddings are concatenated and fused through a lightweight MLP to obtain a shared latent feature $\mathbf{z}_t \in \mathbb{R}^{512}$. This latent feature serves as the input to our two-stage dynamics modeling pipeline, where Stage I predicts future contact events and Stage II performs contact-conditioned pose diffusion.

Stage I: Contact Predictor. Given $\mathbf{z}_t$, an MLP predicts an

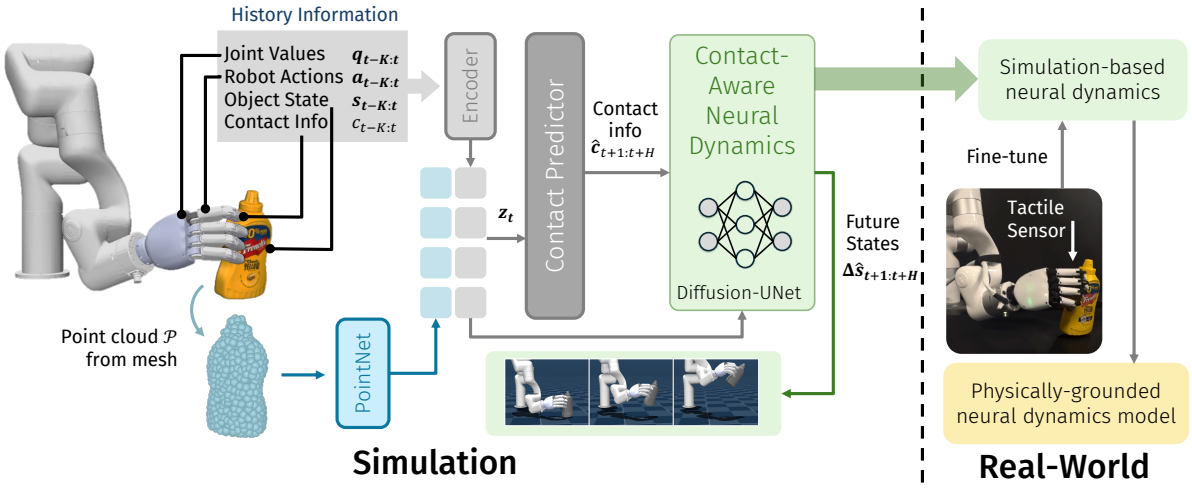

Figure 2. **Overview of the proposed contact-aware neural dynamics framework.** The model takes as input a multimodal history $\mathcal{H}_t = \{\mathbf{s}_{t-K:t}, \mathbf{q}_{t-K:t}, \mathbf{a}_{t-K:t}, c_{t-K:t}, \mathcal{P}\}$, including past object poses, joint values, robot actions, binary contact signals, and the object point cloud. A temporal encoder extracts features from the state–action–contact sequence, while a PointNet encoder processes the geometry $\mathcal{P}$. Their fused latent representation $\mathbf{z}_t$ is used by a contact prediction module to infer future contacts $\hat{c}_{t+1:t+H}$, which then condition a diffusion-based pose predictor that outputs future pose increments $\Delta\hat{\mathbf{s}}_{t+1:t+H}$. The model is first trained on large-scale simulation data and subsequently fine-tuned with a small amount of real-world interaction data, enabling implicit alignment of simulated and physical contact dynamics.

H-step contact probability sequence:

$$\hat{c}_{t+1:t+H} = \sigma(\mathbf{W}_c \mathbf{z}_t + \mathbf{b}_c), \tag{3}$$

$$\mathcal{L}_{\text{cnt}} = \text{BCE}(\hat{c}_{t+1:t+H},\, c_{t+1:t+H}), \tag{4}$$

where $\sigma(\cdot)$ denotes the logistic sigmoid. The predicted sequence $\hat{c}_{t+1:t+H}$ is further projected through a small MLP to obtain a compact contact feature $\mathbf{f}_c \in \mathbb{R}^{d_c}$, where $d_c = 64$ denotes the dimensionality of the contact embedding vector.

Finally, we form the dynamics condition vector by concatenation:

$$\mathbf{h}_t = [\,\mathbf{z}_t\,;\,\mathbf{f}_c\,]. \tag{5}$$

Stage I explicitly augments this latent representation with the predicted contact feature $\mathbf{f}_c$, producing the contact-conditioned vector $\mathbf{h}_t$ that serves as the input to Stage II.

Stage II: Diffusion Pose Predictor.

We model future pose changes with respect to the previous timestep. Let $\mathbf{x}_0 = \Delta\mathbf{s}_{t+1:t+H}$ denote the sequence of H pose increments, each represented in a 6D minimal form $\Delta\mathbf{s}_{t+k} = [\,\Delta\mathbf{p}_{t+k},\, \boldsymbol{\omega}_{t+k}\,]$. The translation delta is defined as

$$\Delta\mathbf{p}_{t+k} = \mathbf{p}_{t+k} - \mathbf{p}_{t+k-1}, \tag{6}$$

and the rotation increment $\boldsymbol{\omega}_{t+k} \in \mathbb{R}^3$ maps to a relative rotation through the exponential map

$$\mathbf{R}_{t+k} = \exp(\widehat{\boldsymbol{\omega}}_{t+k})\,\mathbf{R}_{t+k-1}, \tag{7}$$

where $\widehat{\boldsymbol{\omega}}$ denotes the skew-symmetric matrix of $\boldsymbol{\omega}$.

To model the distribution over $\mathbf{x}_0$, we use a conditional denoising diffusion model, whose forward process is

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\, (1-\bar{\alpha}_t)\mathbf{I}\big), \tag{8}$$

where $\mathbf{x}_t$ is the noisy sample and $\bar{\alpha}_t$ is the cumulative noise schedule.

The reverse denoising process is parameterized by a 1D U-Net, introduced here as the noise predictor:

$$\boldsymbol{\epsilon}_\theta = \text{UNet}_{\text{1D}}(\mathbf{x}_t,\, t,\, \mathbf{h}_t), \tag{9}$$

with FiLM-modulated conditioning on $\mathbf{h}_t$ applied at all layers.

The training objective is

$$\mathcal{L}_{\text{diff}} = \mathbb{E}\big[\,\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{h}_t)\|_2^2\,\big], \tag{10}$$

and the trajectory is reconstructed as

$$\hat{\mathbf{s}}_{t+1:t+H} = \mathbf{s}_t \oplus \hat{\mathbf{x}}_0, \tag{11}$$

where $\oplus$ applies the recovered incremental poses $\hat{\mathbf{x}}_0$ to $\mathbf{s}_t$ over the prediction horizon.

Joint objective. The overall training loss combines both stages:

$$\mathcal{L} = \mathcal{L}_{\text{cnt}} + \lambda\,\mathcal{L}_{\text{diff}}, \tag{12}$$

which jointly optimizes contact forecasting and contact-conditioned dynamics, yielding stable, physically consistent, and long-horizon motion predictions.

3.2. Contact Modeling

In manipulation tasks, contact dynamics between the fingers and the object are highly non-smooth: force spikes and velocity discontinuities occur at the moment of touch, making continuous contact quantities difficult to model reliably. Instead of regressing continuous contact forces or distributions, we adopt a more robust and learning-friendly *binary, hand-level* contact representation that focuses on the structural signal of whether contact occurs. In practice, continuous contact measurements from tactile or force sensors are noisy, sensitive to calibration, and often exhibit small fluctuations even when the qualitative contact state does not change. Using a binary signal reduces the impact of these fluctuations and gives cleaner supervision for training. Moreover, the binary labels can be consistently derived from both simulation (via collision detection) and real hardware (via tactile sensor readings), which helps align the contact representation across simulation and real-world. This design choice also complements the smooth nature of neural networks: instead of forcing the network to fit high-frequency variations in contact magnitude, it learns to predict the underlying discrete event of contact, which is then used as a stable conditioning signal for the downstream dynamics model.

Simulation. In simulation, fingertip and object collision meshes in MuJoCo [41] are used to compute binary contact signals. A contact is labeled as $c_t{=}1$ if any fingertip mesh intersects the object mesh at time t, and vice versa.

Real world. In real experiments, the XHand fingertips are equipped with force-based tactile sensors, with more detailed information in Section 4.1. A fingertip is considered in contact when its measured normal force exceeds a threshold τ_{force}. The global contact label is set to $c_t{=}1$ if any fingertip surpasses this threshold. This threshold-based definition provides a consistent and robust binary contact representation across simulation and the real system.

3.3. Implicit Sim-to-Real Alignment with Contacts

We begin by training our dynamics model entirely in simulation, following the architecture illustrated in Fig. 2. The model takes as input the recent history of object states, joint values, robot actions, and binary contact signals, together with the object point cloud. A temporal encoder processes the state–action–contact history, while a PointNet-based geometry encoder extracts shape features from the point cloud. Their outputs jointly condition the denoising diffusion model, which predicts future contact events and object-relative state transitions.

To further enhance the model's representation of contact-induced dynamics, we perform fine-tuning using simulation data only. This stage leverages a larger and more diverse set of simulated trajectories while maintaining the same contact modeling framework to refine the model. For the single-object setting, we use the mustard bottle from the YCB [5] dataset with 8,000 simulated trajectories. For the multi-object setting, we employ 15,000 simulated trajectories covering 40 YCB objects, with randomized physical and contact parameters to improve robustness and generalization.

All object point clouds are uniformly sampled from their corresponding meshes to ensure consistent geometric representations. Fine-tuning continues from the pretrained weights of the base dynamics model and adopts a lower learning rate to stabilize optimization and further refine the contact-conditioned latent representation. Although no real-world data are used at the first stage, the simulation base model serves as a strong prior, providing a solid foundation for subsequent co-training with real data and improving consistency with tactile contact behaviors observed in real experiments.

4. Results

We evaluated our contact-aware neural dynamics model in both simulation and real-world settings to assess its capability to bridge the sim-to-real gap and accurately model contact-induced motion in robot manipulation tasks. Our experiments are designed to examine: (i) long-horizon forward prediction accuracy, (ii) generalization across diverse objects and contact configurations, and (iii) the impact of contact conditioning on sim-to-real transfer.

Overall, the results demonstrate that incorporating explicit contact representations significantly enhances both the physical realism and temporal consistency of the predicted trajectories. Compared with previous neural dynamics baselines [52], our model achieves lower trajectory prediction errors, more stable multi-step rollouts, and improved alignment with real tactile signals. Qualitatively, it produces physically plausible object motions that closely match observed contact transitions, while quantitatively achieving substantial gains in MSE and ADD-S metrics across both single-object and multi-object scenarios.

These findings highlight that contact-aware learning provides an effective and scalable framework for aligning simulated and real-world dynamics, enabling robust forward prediction and reliable policy evaluation in contact-rich manipulation tasks.

4.1. Experiment Setup

The real-world setup is shown in Figure 3. Our system consists of an XArm7 robotic arm equipped with the XHand. A collection of everyday objects is used to evaluate contact-rich grasping and manipulation tasks. Each fingertip of the XHand is equipped with a force-based tactile sensor, and contact is detected when the measured normal force exceeds a predefined threshold, providing reliable binary contact signals during real-world experiments.

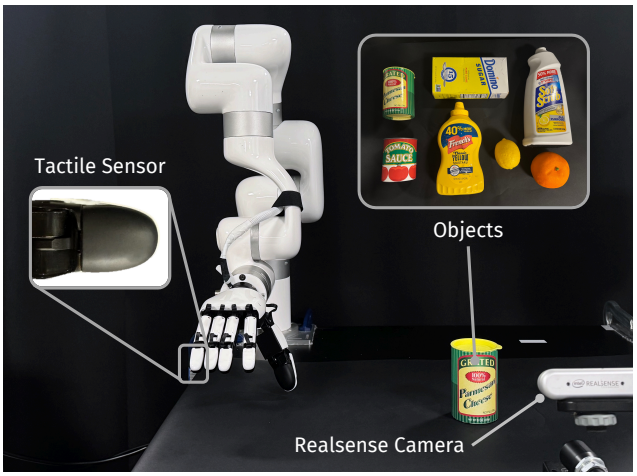

Figure 3. Real-world setup with an XArm7 robot arm and an XHand equipped with tactile sensors for contact detection. A Realsense camera captures visual observations, and everyday YCB objects are used for grasping.

To obtain the real-world object poses, we used FoundationPose [42] as the pose estimation backbone. To mitigate the residual noise and drift introduced by vision-based estimation, we adopt a lower control frequency and apply small random perturbations to the measured poses during training, which effectively regularizes the dynamics model and reduces the impact of pose errors.

4.1.1. Tactile Sensor Setup

The XHand is equipped with five fingertip modules-one per finger - each consisting of an array of tri-axial tactile sensors capable of capturing detailed contact forces along the x, y, and z axes at around 120 uniformly distributed points on the fingertip with a minimum resolution of 0.05 N. These tactile arrays enable fine-grained perception of local contact distributions, making them suitable for manipulation tasks requiring precise force feedback. During operation, the XHand communicates via an RS485 interface with the control and sensing loop running at approximately 80–85 Hz. For each fingertip sensor, the high-level computed force vector $F_{\text{calc}} = [F_x, F_y, F_z]$, representing the aggregated 3D contact force at the fingertip is measured.

Before data collection, all sensors undergo a reset and calibration routine to remove static offsets and ensure consistent baselines. The calibration procedure records the mean force over a stationary 3 s window and subtracts it from subsequent readings:

$$\mathbf{F}_{\text{calibrated}} = \mathbf{F}_{\text{calc}} - \mathbf{F}_{\text{offset}}. \tag{13}$$

A lightweight contact detection heuristic identifies contact events based on the cumulative force magnitude:

$$c_i = \begin{cases} 1, & \text{if } |F_x| + |F_y| + |F_z| > 0.3N, \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

yielding a binary contact flag c_i for each fingertip.

4.2. Qualitative Results

We present qualitative evaluations to illustrate how our contact-aware neural dynamics model bridges the sim-to-real gap and improves long-horizon prediction fidelity.

As qualitatively demonstrated in Fig. 4, our method produces simulated rollouts exhibiting high fidelity to real-world observations. It faithfully captures object motion and abrupt contact transitions—nuances standard simulators often miss due to simplified dynamics. Even under significant variations in contact geometry, our model preserves structural integrity and generates physically plausible states, highlighting robustness against domain randomization and parameter mismatch.

Figure 5 compares multi-step 3D trajectory rollouts between ground truth and predictions. The proposed two-stage architecture is critical: by conditioning diffusion dynamics on inferred contact cues, it maintains spatial consistency and models contact-driven discontinuities. In contrast, single-step predictors neglecting contact suffer from compounding errors, leading to significant drift. Our model adaptively switches motion regimes, resembling a simulator with collision detection. This capability is evident in slippage scenarios, where the model detects contact loss and adjusts the trajectory, yielding stable, physically consistent predictions.

Overall, these results validate that integrating contact-aware representations bridges the sim-to-real gap. Our approach generalizes effectively, ensuring synthesized predictions remain physically grounded and temporally coherent in contact-rich manipulation tasks.

4.3. Performance Comparisons

We further provide quantitative comparisons in Table 1, evaluating different dynamics models under both single- and multi-object settings across three data regimes (simulation, real, and co-training). Performance is measured by Mean Squared Error (MSE) and the **area under the curve (AUC) of ADD-S** [43]. Hereafter, we simply refer to this metric as **ADD-S**. Higher is better. Lower MSE and higher ADD-S indicate better prediction accuracy. Intuitively, **ADD-S** captures how often the predicted and ground-truth trajectories stay within distance thresholds in 3D space, reflecting how well the model preserves the object's geometric consistency during long-horizon motion prediction. As shown in the table, our *Diffusion-UNet with contact* achieves the best performance across all regimes,

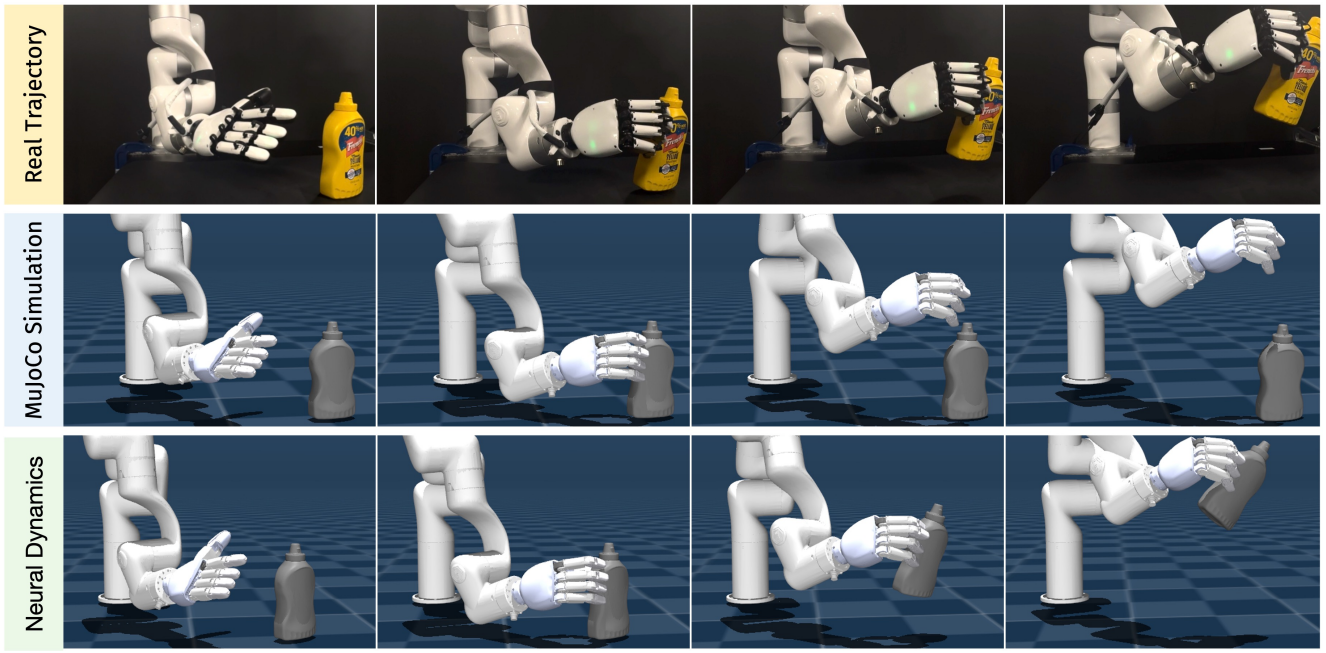

Figure 4. **Qualitative comparison between real, simulated, and our contact-aware neural dynamics results.** The first row shows real-world rollouts, the second shows standard MuJoCo simulations, and the third shows our model predictions. Standard simulation often yields unstable or incorrect contacts and physically implausible object motion due to limited contact modeling and the sim-to-real gap. In contrast, our model produces smoother, more realistic trajectories aligned with real-world motion. When co-trained with a small amount of real data, it further improves temporal stability and contact consistency, demonstrating stronger sim-to-real transfer.

Table 1. Quantitative comparison of single-object and multi-object tasks under different training regimes (simulation only and simulation+real co-training). Metrics: mean squared error (MSE↓) and **AUC of ADD-S** (% ↑). Hereafter we abbreviate it as **ADD-S**. Our model outperforms neural dynamics baselines in both settings, with further gains after limited real-world fine-tuning, especially for multi-object tasks, highlighting the strong generalization of our contact-aware representation and its ability to reduce the sim-to-real gap.

	Single object						**Multiple objects**					
	Sim data		Real data		Real-Finetune		Sim data		Real data		Real-Finetune	
Method	MSE↓	ADD-S↑	MSE	ADD-S	MSE	ADD-S	MSE	ADD-S	MSE	ADD-S	MSE	ADD-S
Baseline [52]	0.016	71.01	0.0194	68.72	—	—	0.0159	61.60	0.0161	62.98	—	—
MLP	0.026	62.58	0.0130	78.12	0.0110	77.43	0.0150	60.11	0.0082	71.76	0.0069	73.43
UNet	0.022	65.86	0.0150	68.45	0.0130	70.11	0.0170	67.74	0.0084	72.09	0.0085	74.12
Diffusion-UNet	0.021	69.12	0.0098	80.03	0.0091	82.45	0.0120	69.95	0.0083	73.04	0.0065	75.82
Diffusion-UNet w/ Contact	0.015	68.12	0.0094	81.34	**0.0082**	**88.23**	0.0100	69.34	0.0075	73.33	**0.0058**	**79.12**

consistently yielding lower prediction errors and higher spatial accuracy.

In particular, the model exhibits significant gains under the co-training setup, which achieves **0.0082 MSE** and **88.23% ADD-S** in single-object tasks, while also maintaining strong performance in the more complex multi-object setting. These results highlight that incorporating contact-aware representations not only improves physical realism but also enhances sim-to-real transfer and generalization.

4.4. Applications

Neural forward dynamics models have shown broad applicability in capturing complex physical interactions. Recent works [19, 29] demonstrate that such forward models can not only accurately predict future object motion, but also infer latent physical properties and provide differentiable structure for downstream decision-making. By learning contact-induced dynamics directly from data, these models can replace or augment traditional simulators, enabling more precise dynamics prediction and control in challeng-

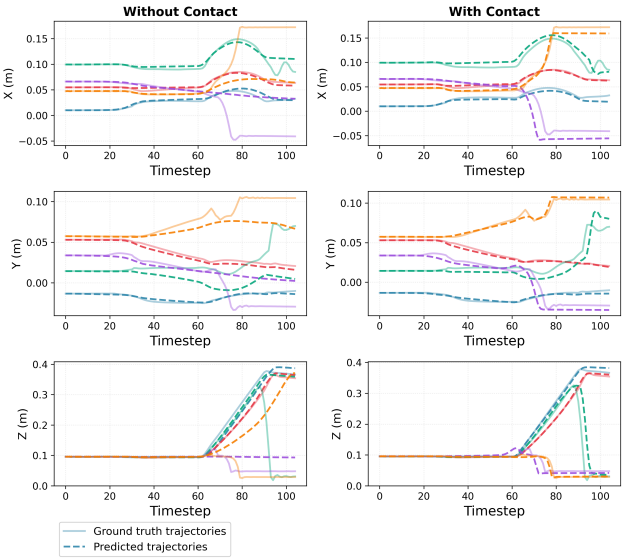

Figure 5. Comparison of multi-trajectory rollouts: the left panel shows predictions from our two-stage contact-aware model versus ground truth, while the right panel shows rollouts from a direct dynamics predictor compared with the same ground truth.

Table 2. Task success rate under single-object and multi-object settings. Success is defined as the percentage of rollouts whose final predicted object position deviates from the real trajectory endpoint by less than 5 cm.

Method	**Single-object (%)**	**Multi-object (%)**
Real-only	52.6	47.1
Sim+Real	73.7	64.7

ing manipulation scenarios. Beyond prediction, neural forward models serve as strong priors for control and policy learning: they can be integrated into model-based planning frameworks or used to fine-tune purely simulation-trained policies (e.g., large-scale dexterous manipulation policies such as Dex1B [46]), allowing better adaptation to real-world friction, compliance, and contact patterns. Overall, neural dynamics models offer a flexible and unified foundation for simulation, inference, and control, and hold significant promise for contact-rich robotic manipulation. An example application we performed is to use this neural physics model to evaluate and filter a manipulation policy trained only in simulation.

Task Success Rate Evaluation. We evaluated our neural dynamics model using the *task success rate*, defined as the percentage of rollouts whose final predicted object position deviates from the real-world trajectory endpoint by less than 5 cm. This metric quantifies whether the model can maintain accurate and drift-free long-horizon predictions that remain consistent with real object motion.

We evaluate two training regimes: (1) **Real-only**, trained solely on real-world trajectories; and (2) **Sim+Real w/ Contact**, which is pretrained on large-scale simulation data and subsequently fine-tuned using real-world contact observations. As shown in Table 2, the Sim+Real w/ Contact model achieves significantly higher success rates in both single-object and multi-object scenarios, reaching **73.7%** and **64.7%**, respectively. In contrast, the Real-only model suffers from accumulated prediction drift over long horizons. Leveraging structured priors from simulation and refining them with real contact supervision enables our model to effectively bridge the sim-to-real gap.

These results demonstrate that incorporating contact-aware dynamics not only improves physical prediction fidelity but also leads to substantially higher robustness and success in downstream manipulation tasks.

5. Conclusions and Limitations

This paper presents a contact-aware neural dynamics model for contact-rich manipulation tasks. By combining a unified binary contact representation, a geometry-aware point cloud encoder, and a conditional diffusion structure, our framework enables stable prediction of future object motion across multi-object and multi-contact scenarios. Large-scale simulation data provide a robust prior for contact-induced dynamics, while limited real-world co-training further improves the model's adaptability to real friction, sensing noise, and non-ideal contact behaviors. Experimental results demonstrate that our method significantly outperforms existing neural dynamics baselines in both single-object and multi-object settings, effectively narrowing the sim-to-real gap and offering a promising direction for building generalizable manipulation models.

Despite these strengths, our approach has several limitations. First, the model relies on object states estimated by FoundationPose during data collection, whose accuracy may degrade under occlusion, clutter, or multi-object stacking, leading to accumulated errors during prediction. Second, while binary contact signals are stable and easy to learn, they cannot fully capture richer real-world contact attributes such as contact area, slip direction, or force distribution. Third, achieving broad generalization across diverse motions and manipulation tasks requires a large and varied dataset, which limits scalability when data collection is costly or task coverage is limited. Finally, although the model performs well over short prediction horizons, long-horizon rollouts still suffer from compounding errors, particularly under frequent contact switching or rapid object motion, constraining its applicability in long-term planning or large-span dynamic prediction. Future work may incorporate more accurate tracking, richer contact representations, and more data-efficient or structured dynamics models to improve long-horizon stability and generalization.

Acknowledgment

This work was supported, in part, by NSF CCF-2112665 (TILOS), and gifts from Amazon.

References

[1] Jad Abou-Chakra, Krishan Rana, Feras Dayoub, and Niko Suenderhauf. Physically embodied gaussian splatting: A visually learnt and physically grounded 3d representation for robotics. In *8th Annual Conference on Robot Learning*, 2024. 1

[2] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 2, 3

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 2

[4] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *Conference on Robot Learning (CoRL)*, 2025. 2

[5] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 510–517, 2015. 5

[6] Michael B. Chang, Tomer Ullman, Antonio Torralba, and Joshua B. Tenenbaum. A compositional object-based approach to learning physical dynamics. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017. ICLR. 2

[7] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019. 2

[8] Peter Yichen Chen, Chao Liu, Pingchuan Ma, John Eastman, Daniela Rus, Dylan Randle, Yuri Ivanov, and Wojciech Matusik. Learning object properties using robot proprioception via differentiable robot-object interaction. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5997–6004. IEEE, 2025. 2

[9] Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024. 1

[10] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2023. 2

[11] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. In *International Conference on Learning Representations (ICLR)*, 2024. 2

[12] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. In *arXiv preprint arXiv:1812.00568*, 2018. 2

[13] Yu Fang, Yue Yang, Xinghao Zhu, Kaiyuan Zheng, Gedas Bertasius, Daniel Szafir, and Mingyu Ding. Rebot: Scaling robot learning with real-to-sim-to-real robotic video synthesis. *arXiv preprint arXiv:2503.14526*, 2025. 2

[14] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 64–72, Red Hook, NY, USA, 2016. Curran Associates Inc. 2

[15] Junpeng Gao, Mike Y Michelis, Andrew Spielberg, and Robert K Katzschmann. Sim-to-real of soft robots with learned residual physics. *IEEE Robotics and Automation Letters*, 2024. 2

[16] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019. 2

[17] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020. arXiv preprint arXiv:1912.01603. 2

[18] Xiaoshen Han, Minghuan Liu, Yilun Chen, Junqiu Yu, Xiaoyang Lyu, Yang Tian, Bolun Wang, Weinan Zhang, and Jiangmiao Pang. Re^{3}Sim: Generating high-fidelity simulation data via 3D-photorealistic real-to-sim for robotic manipulation. *arXiv preprint arXiv:2502.08645*, 2025. 2, 3

[19] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *ICML*, 2022. 7

[20] Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S Sukhatme. Neuralsim: Augmenting differentiable simulators with neural networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9474–9481. IEEE, 2021. 2

[21] Suning Huang, Qianzhong Chen, Xiaohan Zhang, Jiankai Sun, and Mac Schwager. Particleformer: A 3d point cloud world model for multi-object, multi-material robotic manipulation. *arXiv preprint arXiv:2506.23126*, 2025. 3

[22] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Jay Kundalia, et al. Dreamgen: Unlocking generalization in

robot learning through video world models. In *Proceedings of The 9th Conference on Robot Learning*, pages 5170–5194. PMLR, 2025. 2

[23] Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. *arXiv preprint arXiv:2410.22325*, 2024. 2

[24] Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation. *arXiv preprint arXiv:2510.20813*, 2025. 2

[25] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. *arXiv preprint arXiv:2503.17973*, 2025. 3

[26] Wisama Khalil, Maxime Gautier, and Philippe Lemoine. Identification of the payload inertial parameters of industrial manipulators. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4943–4948. IEEE, 2007. 2

[27] Andrej Kruzliak, Jiri Hartvich, Shubhan P Patni, Lukas Rustler, Jan Kristof Behrens, Fares J Abu-Dakka, Krystian Mikolajczyk, Ville Kyrki, and Matej Hoffmann. Interactive learning of physical object properties through robot manipulation and database of object measurements. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7596–7603. IEEE, 2024. 2

[28] Quentin Leboutet, Julien Roux, Alexandre Janot, Julio Rogelio Guadarrama-Olvera, and Gordon Cheng. Inertial parameter identification in robotics: A survey. *Applied Sciences*, 11(9):4303, 2021. 2

[29] Xueyi Liu, He Wang, and Li Yi. Dexndm: Closing the reality gap for dexterous in-hand rotation via joint-wise neural dynamics model, 2025. 7

[30] Nikos Mavrakis and Rustam Stolkin. Estimation and exploitation of objects' inertial parameters in robotic grasping and manipulation: A survey. *Robotics and Autonomous Systems*, 124:103374, 2020. 2

[31] Himangi Mittal, Peiye Zhuang, Hsin-Ying Lee, and Shubham Tulsiani. Uniphy: Learning a unified constitutive model for inverse physics simulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16208–16218, 2025. 3

[32] Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. Data-efficient domain randomization with bayesian optimization. *IEEE Robotics and Automation Letters*, 6(2): 911–918, 2021. 2

[33] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, page 2863–2871, Cambridge, MA, USA, 2015. MIT Press. 2

[34] Mihir Parmar, Mathew Halm, and Michael Posa. Fundamental challenges in deep learning for stiff contact dynamics. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5181–5188. IEEE, 2021. 2

[35] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018. 2

[36] Nicholas Pfaff, Evelyn Fu, Jeremy Binagia, Phillip Isola, and Russ Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups. *arXiv preprint arXiv:2503.00370*, 2025. 2

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 3

[38] M Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhisesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6502–6509. IEEE, 2025. 2

[39] Trevor Standley, Ozan Sener, Dawn Chen, and Silvio Savarese. image2mass: Estimating the mass of an object from its image. In *Conference on Robot Learning*, pages 324–333. PMLR, 2017. 2

[40] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 2

[41] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 5

[42] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024. 6

[43] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 6

[44] Jie Xu, Eric Heiden, Iretiayo Akinola, Dieter Fox, Miles Macklin, and Yashraj Narang. Neural robot dynamics. *arXiv preprint arXiv:2508.15755*, 2025. 2, 3

[45] Yu Yang, Zhilu Zhang, Xiang Zhang, Yihan Zeng, Hui Li, and Wangmeng Zuo. Physworld: From real videos to world models of deformable objects via physics-aware demonstration synthesis. *arXiv preprint arXiv:2510.21447*, 2025. 2, 3

[46] Jianglong Ye, Keyi Wang, Chengjing Yuan, Ruihan Yang, Yiquan Li, Jiyue Zhu, Yuzhe Qin, Xueyan Zou, and Xiaolong Wang. Dex1b: Learning with 1b demonstrations for dexterous manipulation. In *Robotics: Science and Systems (RSS)*, 2025. 8

[47] Jianglong Ye, Lai Wei, Guangqi Jiang, Changwei Jing, Xueyan Zou, and Xiaolong Wang. From power to precision:

Learning fine-grained dexterity for multi-fingered robotic hands. *arXiv preprint arXiv:2511.13710*, 2025. 2

[48] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017. 2

[49] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, 2020. 2

[50] Albert J Zhai, Yuan Shen, Emily Y Chen, Gloria X Wang, Xinlei Wang, Sheng Wang, Kaiyu Guan, and Shenlong Wang. Physical property understanding from language-embedded feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28296–28305, 2024. 2

[51] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. *Robotics: Science and Systems (RSS)*, 2024. 2, 3

[52] Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. Particle-grid neural dynamics for learning deformable object models from rgb-d videos. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. 5, 7

[53] Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: learning 4d embodied world models. *arXiv preprint arXiv:2504.20995*, 2025. 2, 3