Tac2Real: Reliable and GPU Visuotactile Simulation for Online Reinforcement Learning and Zero-Shot Real-World Deployment

Ningyu Yan[1,2*], Shuai Wang[1*], Xing Shen[1], Hui Wang[1,4], Hanqing Wang[1†], Yang Xiang[2,3†], and Jiangmiao Pang[1†]

[1] Shanghai Artificial Intelligence Laboratory.
[2] The Hong Kong University of Science and Technology.
[3] Algorithms of Machine Learning and Autonomous Driving Research Lab, HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute.
[4] Shanghai Jiao Tong University

Abstract. Visuotactile sensors are indispensable for contact-rich robotic manipulation tasks. However, policy learning with tactile feedback in simulation, especially for online reinforcement learning (RL), remains a critical challenge, as it demands a delicate balance between physics fidelity and computational efficiency. To address this challenge, we present Tac2Real, a lightweight visuotactile simulation framework designed to enable efficient online RL training. Tac2Real integrates the Preconditioned Nonlinear Conjugate Gradient Incremental Potential Contact (PN CG-IPC) method with a multi-node, multi-GPU high-throughput parallel simulation architecture, which can generate marker displacement fields at interactive rates. Meanwhile, we propose a systematic approach, TacAlign, to narrow both structured and stochastic sources of domain gap, ensuring a reliable zero-shot sim-to-real transfer. We further evaluate Tac2Real on the contact-rich peg insertion task. The zero-shot transfer results achieve a high success rate in the real-world scenario, verifying the effectiveness and robustness of our framework. The project page is: https://ningyurichard.github.io/tac2real-project-page/

Keywords: Visuotactile Simulation · Online RL · Sim-to-Real Transfer

1 Introduction

Tactile sensing is a cornerstone of human dexterity and has long motivated the design of robotic manipulation systems [33, 36, 40]. Our sense of touch enables us to perceive contact forces, surface textures, and local geometries, which is essential for contact-rich behaviors such as compliant grasping, precise assembly, and hand manipulation, especially under degraded or ambiguous visual feedback

*Equal contribution. † Corresponding authors.

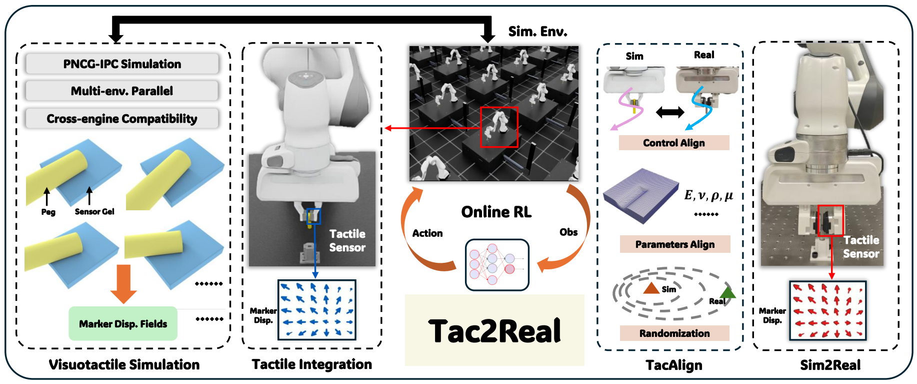

Fig. 1: Tac2Real Framework. Tac2Real contains a visuotactile simulator, which can be seamlessly integrated into existing physics engines in a highly parallel way. Tac2Real supports online large-scale RL training. Through the systematic approach introduced in the TacAlign framework, the sim-to-real gap will be significantly reduced, and Tac2Real guarantees reliable real-world policy deployment.

[39,41,47]. In robotics, vision-based tactile sensors (VBTSs) such as the GelSight family have become a widely adopted sensing modality [1,42,43]. By capturing high-resolution tactile images and marker displacement fields, VBTSs translate low-level contact physics into rich perceptual signals that align naturally with modern computer vision and learning-based frameworks.

However, integrating VBTS into data-driven policy training represents a key challenge for online RL. The fundamental limitation lies in efficiently synthesizing high-fidelity simulated tactile signals. While some penalty-based [2,35] approaches have been successfully applied to scalable tactile data generation, they fall short of modeling soft deformation and multi-phase contact dynamics that underpin realistic VBTS sensing. In contrast, high-fidelity physics-based alternatives such as material point methods (MPM) [3,7,17,30] can capture deformations with greater accuracy, yet often struggle with shear response and numerical stability under large deformations [16,22,45], making them unsuitable for complex contact-rich tasks. Furthermore, few existing simulation pipelines are architected for multi-GPU acceleration, restricting their use in large-scale parallelization for online RL. Against this backdrop, the zero-shot sim-to-real transfer [11,27,46,49] for tactile-based RL remains largely unresolved, with limited successful demonstrations on real hardware.

In this work, we present **Tac2Real**, a high-fidelity visuotactile simulation framework designed to address these limitations, as shown in Fig. 1. At its core, Tac2Real uses the Preconditioned Nonlinear Conjugate Gradient Incremental Potential Contact (PNCG-IPC) solver [28], which builds typical IPC method [10,15,19,20] and guarantees physically consistent elastomer simulation with robust contact handling. By deploying this solver on a multi-node, multi-GPU architecture, we achieve the simulation throughput that enables large-scale online RL. To mitigate uncertainties and high-dimensional cost introduced by optical rendering, Tac2Real directly outputs the marker displacement field as tactile representation and is designed as a lightweight plugin, allowing seamless integration with mainstream robot simulators such as Isaac Lab [25] and MuJoCo [34].

Beyond simulation, we identify that reliable sim-to-real transfer requires a structured strategy to reduce the reality gap [4, 13, 18, 48]. This gap includes both structured discrepancies, such as mismatched robot dynamics, material parameters, and contact models, and stochastic variations from unmodeled noise and environmental uncertainty. To close this gap, we introduce **TacAlign**, a unified, general calibration pipeline for tactile tasks. It consists of four sequential stages: (i) low-level robot controller calibration via trajectory alignment; (ii) identification of material parameters through both baseline and task-engaged tests; (iii) task-specific fine-tuning of friction and contact parameters; and (iv) domain randomization during RL training.

We evaluate Tac2Real on the challenging blind peg-in-hole task with random initial orientations, a benchmark that exemplifies the inherent difficulties of contact-rich manipulation. Experiments show that policies trained entirely in simulation transfer zero-shot to a real robot with high success rates, outperforming baselines built on TacSL [2] and Tacchi [6]. These results highlight the importance of both high-fidelity tactile simulation and reality-gap mitigation for the real-world deployment of tactile-based RL policies.

To summarize, this work makes the following contributions:

- **Tac2Real**, a high-performance visuotactile simulation framework based on PNCG-IPC, which supports multi-GPU parallelization and enables seamless integration with mainstream physics engines for RL.
- **TacAlign**, a systematic methodology for both reducing structured and stochastic sim-to-real gaps in tactile contact-rich manipulations.
- Experimental validations on real-world peg insertion tasks, demonstrating superior zero-shot transfer performance and robustness over existing state-of-the-art approaches.

We believe that Tac2Real provides a robust foundation for scaling up tactile-based policy learning, and takes a significant step toward the long-standing goal of reliable, real-world dexterous manipulations.

2 Related Work

Research on visuotactile simulation has evolved through three distinct stages: standalone sensor-object modeling, closed-loop policy generation (with or without learning) and sim-to-real trial, and holistic system integration and real-world deployment. Initially, methods like Tacchi [6] and Tacchi 2.0 [32] utilized the Material Point Method (MPM) to accurately model elastomer deformation and generate tactile images, though they often struggled with friction modeling and numerical stability; this prompted a shift toward the more robust Incremental

Table 1: Comparison of proposed Tac2Real with existing methods. S, O, and R denote sensor, object, and robot, respectively. The ranking in contact fidelity is based on suitability for online tactile RL, considering deformation realism, frictional contact handling, and numerical robustness. BC is behavior cloning.

Method	Scene	Sim. Method/ Contact Fidelity	Tactile Representation	Policy Learning	Sim-to-Real
Tacchi [6]	S-O	MPM / medium	RGB	×	×
Tacchi 2.0 [32]	S-O	MPM / medium	Markers	×	×
TacIPC [9]	S-O	IPC / high	RGB	×	×
Difftactile [29]	S-O	MPM+FEM / medium	Markers/RGB	×	✓
Sun *et al.* [31]	S-O	MPM / medium	Markers/RGB	RL	✓
Chen *et al.* [5]	S-O	IPC / high	Markers	RL	✓
TacSL [2]	S-O-R	Non-physics / low	Markers/RGB	RL/BC	✓
TacFlex [44]	S-O-R	FEM / high	Markers/RGB	BC	✓
Taccel [21]	S-O-R	IPC / high	Markers/RGB	×	✓
Tac2Real (ours)	**S-O-R**	**IPC / high**	**Markers**	**RL**	✓

Potential Contact (IPC) method, leading to simulators like TacIPC [9]. The second stage extended these capabilities to policy generation and sim-to-real transfer through differentiable optimization (Difftactile [29]) or reinforcement learning integration [5,31], yet these approaches typically lacked a unified, physically consistent simulation of the entire robot–object–sensor system. Currently, the third class achieves holistic integration within scalable environments like Isaac Gym [24], where frameworks such as TacSL [2], TacFlex [44], and Taccel [21] combine penalty-based [37, 38], Finite Element Method (FEM) [8, 50], or IPC-based tactile models with robot dynamics to support large-scale parallel training and reliable real-world deployment (summarized in Tab. 1).

Our proposed Tac2Real utilizes the PNCG-IPC method [28], an enhanced IPC solver specifically designed to shorten convergence time. By leveraging multi-node and multi-GPU acceleration and seamless integration with existing physics engines, Tac2Real enables large-scale online RL while maintaining high visuotactile fidelity. Also, Tac2Real incorporates a systematic strategy for mitigating sim-to-real gaps. We validate the robustness of our framework through zero-shot real-world deployment.

3 Tac2Real Simulation Framework

3.1 PNCG-IPC Method

Our tactile simulation is built upon the PNCG-IPC solver [28], which strikes an effective balance between physical accuracy and computational efficiency for contact simulation. Additionally, the traditional PNCG-IPC solver has been enhanced to support frictional contact handling.

IPC formulates elastodynamic contact simulation as a variational optimization problem. Using implicit Euler integration, the positions $\mathbf{x}^{t+1}$ at each time step are obtained by minimizing:

$$E(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{M}(\mathbf{x} - \hat{\mathbf{x}}) + h^2\Psi(\mathbf{x}) + B(\mathbf{x}) + D(\mathbf{x}), \quad (1)$$

where the four terms represent the inertia potential, the hyperelastic energy Ψ, and the log-barrier contact potential $B(\mathbf{x})$ and frictional potential $D(\mathbf{x})$, respectively. Crucially, we extend the original PNCG-IPC formulation—which lacks the frictional term—by incorporating $D(\mathbf{x})$ to explicitly handle frictional contact. Further implementation details of our solver are provided in the supplementary material. The standard IPC pipeline employs Newton's method with continuous collision detection (CCD) based line search, which yields high per-iteration accuracy but is computationally expensive and difficult to parallelize on GPUs. PNCG-IPC replaces Newton's method with the nonlinear conjugate gradient algorithm, which directly solves the nonlinear optimization without assembling or factorizing the Hessian matrix. The algorithm only requires gradient evaluations, diagonal Hessian entries, and vector-vector dot products—all highly GPU-parallelizable operations. A CCD-free line search strategy further eliminates the expensive collision detection per iteration by deriving an analytical step-size upper bound.

This design philosophy trades per-iteration accuracy for dramatically faster iteration throughput: while each conjugate gradient step is less precise than a Newton step, the extremely low per-iteration cost on GPUs allows PNCG-IPC to converge to sufficient accuracy within only tens of iterations. For tactile simulation where visually plausible and physically consistent deformation matters more than machine-precision convergence, this trade-off is highly favorable.

Moreover, our PNCG-IPC based method is implemented as a lightweight solver based on the Taichi programming language [14], which compiles to high-performance GPU kernels from concise Python code. This compact implementation makes it straightforward to integrate PNCG-IPC as a plug-in module into diverse simulation pipelines, as detailed in Sec. 3.3.

3.2 Tactile Representation

In this work, we mainly focus on simulating the tactile representations of GelSight Mini sensor, which can output either an RGB tactile image (320×240) or a marker displacement field (9×7) depending on the sensor gel type. Marker displacement fields excel at capturing diverse contact states, which is critical for robots contact-rich manipulation tasks, whereas RGB tactile images are useful for texture identification. To select the appropriate representation, we conduct a simple collision test. We mounted tactile sensors on the gripper of a Franka Panda robot, then grasped a peg to perform various contact interactions with sockets. We recorded both RGB outputs and 2D marker displacement fields under static, press-down, move-forward, and move-backward conditions. As shown in Fig. 2, the marker displacement fields exhibit significant variations across different collision states, while RGB images show only subtle differences. This superior sensitivity of markers to distinct contact modes, combined with their lower-dimensional representation, may prove crucial for large-scale RL training by enhancing regularity and efficiency. In our simulation, a mapping between marker and initial IPC mesh nodes can be constructed using the k-nearest neighbor method; then, the displacements of marker are interpolated using weighted interpolation from mesh node positions in the deformed state.

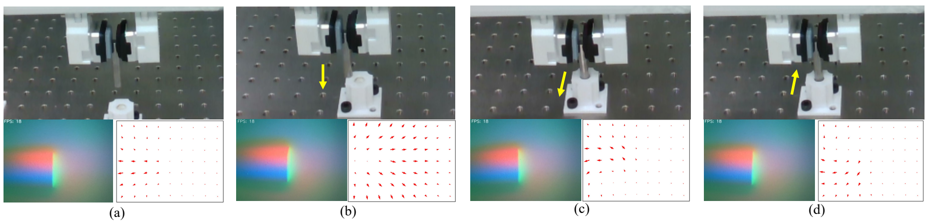

Fig. 2: Tactile feedback under 4 different contact modes in the peg insertion case. (a) stationary state; (b) press-down state; (c) move forward state; (d) move backward state. Tactile marker displacement fields (low dimension) exhibit greater sensitivity to contact than tactile RGB images (high dimension), which may enhance the regularity and efficiency of RL training.

3.3 Integrating to Physics Engine

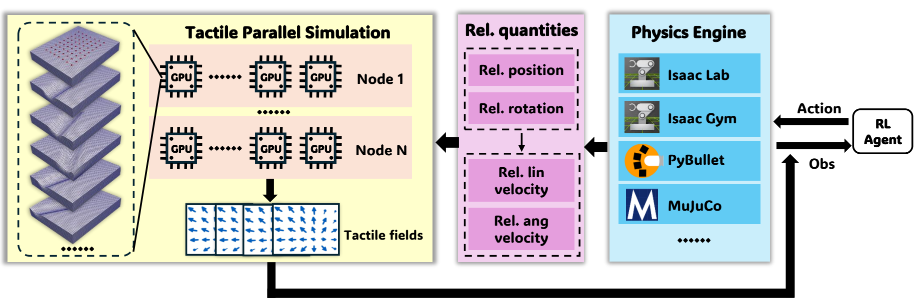

Fig. 3: Tactile Simulation Framework. Our tactile simulation is an external interface outside the physics engine. Receiving relative quantities from physics environments, Tac2Real starts tactile simulation in a multi-node, multi-GPU parallel way, with each GPU assigned tactile tasks from multiple environments.

The integration of our tactile simulation into existing physics engine for RL training follows a plugin-based architecture. As shown in Fig. 3, the RL agent first sends the action command, triggering one simulation step in the physics engine. We then extract the relative position and orientation between the grasped object and the tactile sensor during this step, and compute the corresponding linear and angular velocities which are used in the following tactile simulation. Finally, the output marker displacement fields, together with the baseline observation in the physics engine, form the combined observation that is fed back to the RL agent to continue training. This integration method has the following main properties:

Cross-engine compatibility. Our tactile simulation only requires relative physical quantities between the sensor elastomer and the interacting objects, so this interface can be embedded within the environment file layered on top of the physics engine, for example, the _get_observations() function in Isaac Lab [25], or play_steps_rnn() function in rl-games library [23].

Multi-node Multi-GPU parallelization. Benefiting from the parallelizability of the PNCG-IPC and Taichi language, **Tac2Real** leverages multi-node, multi-GPU parallelization to enable robust and efficient RL training. Specifically, we construct a Ray [26] cluster containing multiple nodes, each equipped with multiple GPUs. A Ray-wrapped tactile simulation class is instantiated for each GPU, which manages tactile simulation for a set of environments. During the online RL roll-out process, the simulation function within the Ray-wrapped class is invoked iteratively: it takes the relative pose quantities as input and returns the marker displacement fields after simulation. Finally, Ray-based distributed communication across multiple nodes is used to gather simulation results from all GPUs.

4 TacAlign to Narrow Sim-to-Real Gap

To systematically mitigate the sim-to-real gap, we propose **TacAlign** within the **Tac2Real** framework, addressing both structured and stochastic discrepancies, as illustrated in Fig. 4.

4.1 Robot Control Alignment

In our framework, the Franka robot in simulation and real world both adopt Cartesian impedance control. The target force applied at the end-effector is given by:$\mathbf{F}^{targ} = \mathbf{k_p} * (\mathbf{p}^{targ}(\mathbf{a}) - \mathbf{p}^{ee}) - \mathbf{k_d} * \mathbf{v}^{ee}$, where $\mathbf{k_p}$ and $\mathbf{k_d}$ denote the proportional and damping gains respectively, with $\mathbf{k_d} = 2\sqrt{\mathbf{k_p}} \in \mathbb{R}^6$ to ensure critical damping. Here, $\mathbf{p}^{targ}(\mathbf{a}) \in \mathbb{R}^6$ is the target end-effector pose determined by the action $\mathbf{a}$, $\mathbf{p}^{ee}$ is the current end-effector pose, and $\mathbf{v}^{ee}$ is the end-effector velocity. * is the element-wise multiplication.

By this controller, the end-effector trajectories in simulation and reality can be expressed as:

$$\mathbf{x}_{t+1}^{\text{sim}} = f_{\text{sim}}(\mathbf{x}_t, \mathbf{a}_t; {\mathbf{k_p}}^{\text{sim}}), \quad \mathbf{x}_{t+1}^{\text{real}} = f_{\text{real}}(\mathbf{x}_t, \mathbf{a}_t; {\mathbf{k_p}}^{\text{real}}), \tag{2}$$

where f_{sim} and f_{real} represent the closed-loop system dynamics induced by the impedance controller. An intuition is that reducing the discrepancy between

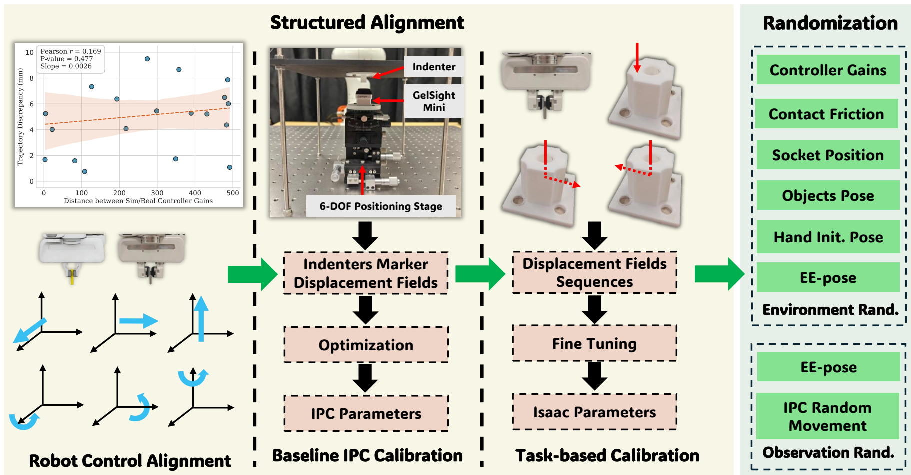

Fig. 4: TacAlign Framework. TacAlign narrows the sim-to-real gap from both structured and stochastic perspectives. TacAlign is composed by robot control alignment, baseline IPC calibration, task-based calibration and randomization.

${\mathbf{k_p}}^{\text{sim}}$ and ${\mathbf{k_p}}^{\text{real}}$ would synchronously reduce the sim-to-real trajectory gap. In practice, however, this relationship is highly non-linear and non-monotonic due to unmodeled dynamics, actuator delays, frictional effects, and contact nonlinearities. We empirically validate this phenomenon in the upper left part of Fig. 4, where we randomly choose 20 sim-real $\mathbf{k_p}$ pairs and visualize the trajectory discrepancy. Notably, the trajectory discrepancy does not necessarily decrease as $|{\mathbf{k_p}}^{\text{sim}} - {\mathbf{k_p}}^{\text{real}}|$ becomes smaller, and exhibits no significant linear correlation. This observation suggests that simply matching controller parameters is insufficient to minimize the sim-to-real gap; instead, trajectory-level alignment is required. We therefore formulate control alignment as the problem of minimizing trajectory discrepancy:$\mathcal{D}({\mathbf{k_p}}^{\text{sim}}, {\mathbf{k_p}}^{\text{real}}) = \frac{1}{T}\sum_{t=1}^{T} \left\|\mathbf{x}_t^{\text{sim}} - \mathbf{x}_t^{\text{real}}\right\|^2$, where the discrepancy is computed over six canonical end-effector motions (single-axis translation and rotation) as shown in Fig. 4 . Rather than treating either domain as ground truth, we jointly calibrate both simulation and reality using alternating minimization:

$$
{\mathbf{k_p}}^{\text{sim},(k+1)} = \arg\min_{{\mathbf{k_p}}^{\text{sim}}} \mathcal{D}({\mathbf{k_p}}^{\text{sim}}, {\mathbf{k_p}}^{\text{real},(k)}), \tag{3}
$$

$$
{\mathbf{k_p}}^{\text{real},(k+1)} = \arg\min_{{\mathbf{k_p}}^{\text{real}}} \mathcal{D}({\mathbf{k_p}}^{\text{sim},(k+1)}, {\mathbf{k_p}}^{\text{real}}). \tag{4}
$$

This bidirectional procedure iteratively reduces trajectory divergence and alleviates controller-induced sim-to-real mismatches.

4.2 Baseline IPC Calibration

Material parameters in PNCG-IPC simulation, specifically Young's modulus E, Poisson's ratio ν, density ρ, and friction coefficient μ, play a critical role in reducing the sim-to-real gap. To ensure that the mechanical response of the elastomer in tactile simulation closely matches physical reality, we conduct a baseline calibration experiment. First, we use the apparatus shown in Fig. 4, which contains a 6-DOF positioning stage, a GelSight Mini sensor and several 3D-printed indenters, to record sequences of marker displacement fields under different indenters and deformation modes. Let $\mathbf{u}_{k,i}^{\mathrm{real}}$ and $\mathbf{u}_{k,i}^{\mathrm{sim}}(\boldsymbol{\theta})$ denote the marker displacement fields for the i indenter at frame k in the real and simulated settings respectively, where $\boldsymbol{\theta} = [E, \nu, \rho, \mu]^\top$ is the vector of material parameters. We then replicate these deformation modes in the IPC simulator and define the mean squared error (MSE) between simulated and real displacement fields as the loss function $\mathcal{L}(\boldsymbol{\theta})$:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{K \cdot N} \sum_{k=1}^{K} \sum_{i=1}^{N} \left\| \mathbf{u}_{k,i}^{\mathrm{sim}}(\boldsymbol{\theta}) - \mathbf{u}_{k,i}^{\mathrm{real}} \right\|_2^2, \tag{5}$$

where K is the total number of frames, and N is the number of indenters. The calibration is formulated as an optimization problem to find the optimal parameter set $\boldsymbol{\theta}^*$ that minimizes this discrepancy:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}). \tag{6}$$

To solve this gradient-free nonlinear optimization efficiently, we adopt the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [12].

4.3 Task-based Calibration

Given the complex interactions encountered in real-world contact-rich tasks, the baseline calibration alone may not capture the full range of sensor deformation. Therefore, we further perform task-aware fine-tuning on simulation parameters of Isaac Lab, such as friction coefficient μ_{isaac} and contact stiffness s_{isaac}, under both static and dynamic contact conditions. Four representative contact states are considered: stationary grasping, press-down, forward and backward collisions as illustrated in Fig. 4. We record the marker displacement field sequences for both simulation and reality, with the real measurements used as the ground-truth reference. Then, we fine-tune the μ_{isaac} and s_{isaac} until the MSE of displacement fields between the closest-matching frames falls below a predefined threshold.

4.4 Randomization

We model the sim-to-real discrepancy as parametric and observational uncertainty. Accordingly, we randomize physical parameters (controller gains, friction), geometric configurations (object and socket poses), and sensing channels (end-effector pose noise and IPC perturbations). This stochastic regularization encourages the policy to be invariant to low-level dynamics mismatches and contact variations. Domain randomization complements our structured alignment by narrowing the stochastic gap that is difficult to resolve through deterministic calibration alone. Details are provided in the supplementary material.

The first two modules of TacAlign (control alignment and baseline IPC calibration) depend only on robot and the tactile sensor, requiring one-time calibration for identical hardware. The task-based calibration essentially builds an "alignment dataset" for a specific task. For a new task, the marker field sequences under static and random contact states need to be collected to cover the manipulation process as much as possible; the four settings adopted in the paper are representative.

5 Experimental Results

5.1 PNCG-IPC Simulation Results

Reliability. We evaluate PNCG-IPC (Tac2Real) against Tacchi and TacSL using a cube indenter rotation task. As shown in Fig. 5(a), Tac2Real and Tacchi both yield realistic deformations consistent with real-world data, whereas TacSL exhibits significant discrepancies due to its non-physical, penalty-based approach that only models interpenetration regions. Under large rotational deformations (Fig. 5(b)), Tacchi suffers from particle splashing and numerical instability caused by adhesion deficiencies. In contrast, Tac2Real robustly handles large rotations and slips, with the sensor reliably recovering its original state post-contact, demonstrating superior stability for contact-rich RL tasks. More simulation results and comparisons are provided in supplementary material.

Parallel Performance. Benchmarked on a node with 16 RTX 4090 GPUs, Tac2Real achieves exceptional scalability (Fig. 5(c)). At 4,096 environments,

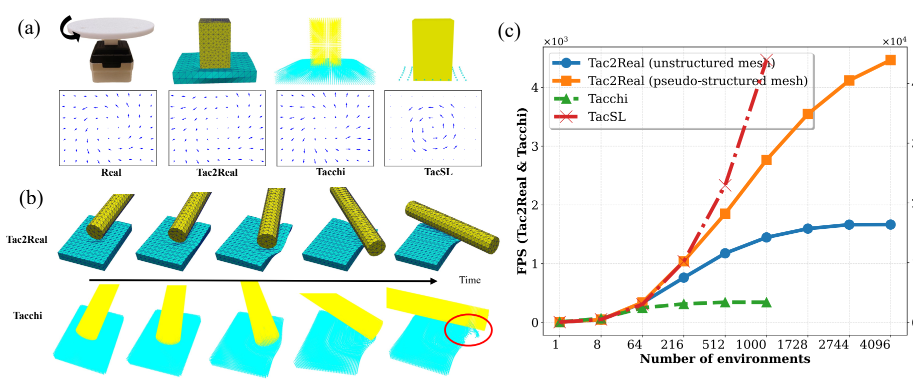

Fig. 5: Comparison of different simulation methods. (a) Comparison among Tac2Real, Tacchi, TacSL and real reference in cube indentation test; (b) Comparison of Tac2Real and Tacchi in large rotation deformation; (c) Parallel performance comparison.

it reaches 4,465 FPS (pseudo-structured mesh) and 1,665 FPS (unstructured mesh), outperforming Tacchi. While TacSL achieves higher raw FPS by relying on simple SDF queries, it sacrifices physical fidelity. Tacchi's lower parallel efficiency stems from the overhead of constructing large background grids for multiple environments on a single GPU. Consequently, Tac2Real offers the optimal balance of high-throughput parallelism and physical accuracy required for online reinforcement learning.

5.2 TacAlign Results

Control Alignment. We employ the alternating optimization approach described above to search for an optimal combination of controller gains for both simulation and reality. We start with the original value of ${\mathbf{k_p}}^{\text{sim}} = (100, 30)$ in Isaac Lab, which is an abbreviation for $(100, 100, 100, 30, 30, 30)$, and keep ${\mathbf{k_p}}^{\text{real}}$ to be the same. Thresholds for the averaged trajectory discrepancy of translation $\bar{\mathcal{D}}_{\text{trans}}$ and rotation $\bar{\mathcal{D}}_{\text{rot}}$ along the 3 coordinate axes are set to 3 mm and 0.5° respectively. As shown in Tab. 2, the initial averaged trajectory discrepancy is relatively large, with a translational mismatch of 11.11 mm recorded. This large discrepancy makes it impossible to complete the peg insertion task, as the diameter of the socket hole in this task is only approximately 8 mm. By the optimization, the trajectory discrepancy is reduced to 2.521 mm (translation) and 0.454° (rotation). The corresponding optimized controller gains are ${\mathbf{k_p}}^{\text{sim}} = (600, 50)$ and ${\mathbf{k_p}}^{\text{real}} = (400, 20)$, which are fixed for all subsequent alignment procedures and RL training.

Table 2: Controller Alignment Performance.

${\mathbf{k_p}}^{\mathbf{sim}}$	${\mathbf{k_p}}^{\mathbf{real}}$	$\bar{\mathcal{D}}_{\mathbf{trans}}$ (mm)	$\bar{\mathcal{D}}_{\mathbf{rot}}$ (degree)
(100, 30)	(100, 30)	11.11	2.635
(300, 30)	(500, 40)	7.381	1.091
(600, 50)	**(400, 20)**	**2.521**	**0.454**

Baseline IPC Calibration Results. Four indenters of distinct shapes, including the cube, cylinder, moon and triangle, are used in the baseline calibration. For each indenter, we performed three deformation interactions with the GelSight Mini sensor's elastomer: pressing, sliding, and rotating. The magnitudes of these three deformation modes are set to 1 mm, 1 mm and 2°, with step increments of 0.1 mm, 0.1 mm and 0.5°, respectively. We also constructed an identical calibration environment (comprising the sensor elastomer and indenters) using PNCG-IPC, simulating the entire deformation process to support the CMA-ES optimization. The final results are presented in Fig. 6, where only the results for the cube indenter at the final frame are illustrated. As evident from the results, the mechanical behavior of the elastomer in simulation closely approximates that in the real world. More optimization information is provided in supplementary material.

Task-based Calibration Results. We further use the marker displacement field sequences from the static state and three typical contact events in real-world peg-insertion tasks 4.3 to fine-tune the physics parameters in Isaac Lab. We find that the contact friction coefficient μ_{isaac}, together with the use of compliant contact in Isaac Lab, plays a critical role in reducing the discrepancy between IPC-simulated and real-world marker displacement fields during contact. The closest fields between reality and simulation for these four task-based scenarios after calibration are shown in Fig. 6, demonstrating strong alignment.

5.3 Online RL Learning Results in Simulation

We first evaluate Tac2Real using two typical contact-rich tasks, where tactile marker displacement fields are essential to estimating the relative pose and identifying the complex contact interactions:

- **Random Orientation Peg Insertion.** The Franka robot is required to fully insert a cylindrical peg held in its gripper into a fixed cylindrical socket. Notably, the entire insertion process, especially after the peg enters the socket, must not rely on gravity; that is, the insertion is achieved entirely through the robot's active control rather than gravitational effects, as illustrated in Fig. 7 (a). The initial orientation of the peg in the gripper is randomly sampled from the range of $[-35°, 35°]$, and both the peg and socket hole have a diameter of 8 mm, which is smaller than the real-world setting used in TacSL.
- **Random Orientation Nut Threading.** The Franka robot must place a nut (also held with a random initial orientation from $-35°$ to $35°$) onto a fixed bolt and fasten the nut onto the bolt via gripper rotation, as shown in Fig. 7 (b). Task success is defined as the nut is threaded into the bolt to a depth of 1.5 pitches.

We formulate peg-in-hole as a Markov decision process (MDP). To emphasize tactile reliance, the policy observes only the end-effector pose ($\mathbf{p}^{ee} \in \mathbb{R}^7$), the

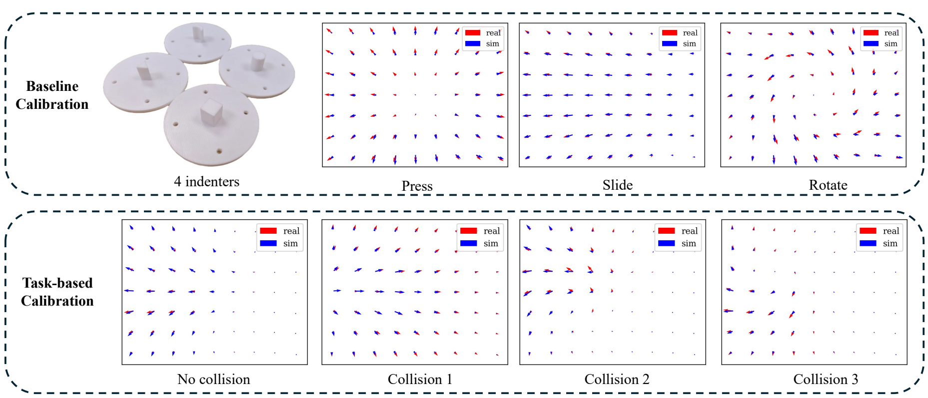

Fig. 6: Simulation parameter calibration.

tactile marker displacement field for a single finger ($\mathbf{u} \in \mathbb{R}^{7\times 9\times 2}$), and the previous action ($\mathbf{a}_{t-1}$), explicitly excluding object pose or visual data. Actions consist of incremental Cartesian updates executed via an impedance controller. The reward function balances task success through keypoint alignment and sparse bonuses for engagement and insertion, while penalizing excessive contact forces to prevent collisions. More training settings are given in supplementary material.

We set the total number of environments to 512, evenly distributing the tactile simulation tasks across four computing nodes, each equipped with 16 GPUs. We use the off-the-shelf Proximal Policy Optimization (PPO) implementation from the rl-games library in Isaac Lab, where the agent adopts a shared actor–critic architecture. For a fair comparison, we also integrate Tacchi and TacSL into Isaac Lab using integration schemes analogous to Tac2Real; further details are provided in the supplementary materials. Fig. 7 shows the average learning curves over three random seeds for the three simulation methods, as well as the setting without tactile observations and the snapshots for the two tasks. We also report the average success rate over 256 random initial configurations in Tab. 3. Among these, training based on Tac2Real and TacSL exhibit similar performance in simulation environments, both significantly outperforming Tacchi and the tactile-free case. This demonstrates the effectiveness of our tactile simulation framework and the critical role of tactile sensing in contact-rich tasks, particularly under nearly blind operating conditions.

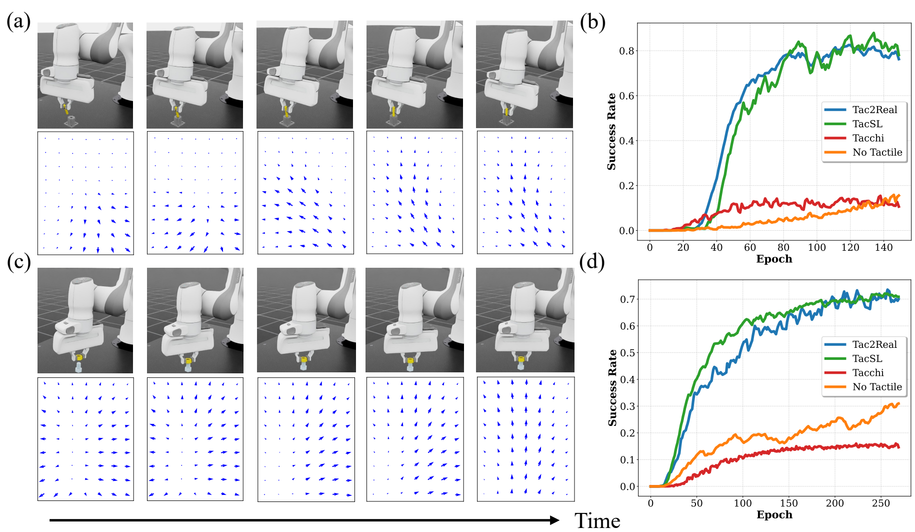

Fig. 7: Contact-rich tasks in simulation. (a) and (c) are snapshots of inference in peg insertion and nut threading tasks with tactile simulation; (b) and (d) are comparisons of the learning curves among Tac2Real, TacSL, Tacchi, and no tactile feedback.

Table 3: Policy Learning Results in both the Simulation and Real-world

	Env.	TacAlign level	Tac2Real	TacSL	Tacchi	No Tactile
Peg Insertion (8mm)	sim	-	0.776	0.789	0.173	0.168
Peg Insertion (12mm)	sim	-	0.831	-	-	-
Peg Insertion (16mm)	sim	-	0.857	-	-	-
Nut Threading	sim	-	0.702	0.708	0.152	0.313
Peg Insertion (8mm)	real	1,2,3,4	**0.917**	0.150	0.083	0.067
Peg Insertion (8mm)	real	2,3,4	0.533	0.033	0.050	0.017
Peg Insertion (8mm)	real	1,2,4	0.250	0.150	0.016	0.067
Peg Insertion (8mm)	real	1,2,3	0.767	0.100	0.100	0.017
Peg Insertion (12mm)	real	1,2,3,4	0.933	-	-	-
Peg Insertion (16mm)	real	1,2,3,4	0.933	-	-	-

5.4 Real-world Deployment

The policy trained in simulation is also deployed in the real world in a zero-shot manner for the peg insertion task. We mounted two GelSight Mini sensors on the Franka grippers to maintain force equilibrium, while using only the tactile marker displacement field from the right finger for inference. Given the sensor's gel pad vulnerability to damage, we implement a protective mechanism in practice. Specifically, when the mean squared error (MSE) difference between successive marker displacement fields exceeds a predefined threshold, we halt inference, move the end-effector back one step, and resume inference from this adjusted position. We initialize the peg's initial orientation to $0°$ and $\pm 15°$, and conduct a total of 60 trials with 20 trials for each peg orientation. As shown in Tab. 3, we recorded 55 successful insertions, achieving an approximate zero-shot sim-to-real transfer success rate of 91.7%. In Fig. 8, some inference snapshots of the 0-degree situation are illustrated. More snapshots are provided in supplementary material. Unlike the results in the simulation, the deployment of TacSL only attains a 15% success rate, attributable to the substantial discrepancy between simulated and real marker displacement fields (see Fig. 5). The Tacchi-based policy achieves

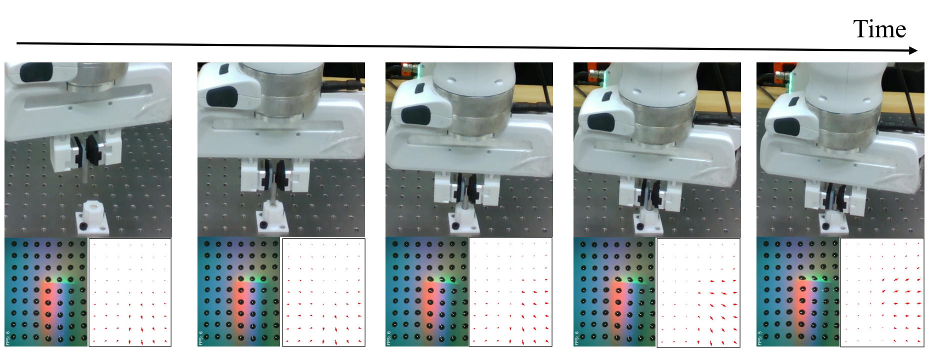

Fig. 8: Peg Insertion Real-world Deployment Snapshots.

few successful insertions because of the large amount of useless tactile feedback caused by MPM numerical instability. Additionally, the policy without tactile observations achieved a real-world success rate of only 6.7%. In order to evaluate the importance of TacAlign in sim-to-real transfer, we conducted ablation studies across different levels of TacAlign, where the control alignment, baseline IPC calibration, task-based calibration, and randomization are labeled as levels 1, 2, 3, and 4, respectively. The ablation results are presented in Tab. 3. We found that the task-based calibration plays an important role in narrowing the sim-to-real gap, without which the success rate will decrease significantly.

In order to show the generalizability of our method, we also implement Tac2Real on the peg insertion task of different peg size (diameters of 12 mm and 16 mm). As shown in Tab. 3, both these two sizes achieve a higher success rate in simulation and real world than 8mm peg. Besides, we have also preliminarily deployed the random orientation nut threading task using Tac2Real in real world, and we achieved a success rate of 58.3%. More details are shown in supplementary material. These results further validate the efficiency of our Tac2Real framework.

6 Conclusion

In this paper, we present a framework named Tac2Real, which contains effective and robust visuotactile simulation using the PNCG-IPC based method with a highly parallel multi-node, multi-GPU computing architecture. Tac2Real supports seamless integration with current physics engines to perform online RL training. We also upgrade Tac2Real with a practical and reproducible framework named TacAlign to systematically reduce the sim-to-real gap from structured and stochastic perspectives. Compared with existing methods, Tac2Real achieves robust and high-fidelity tactile simulation, as well as excellent performance in RL policy zero-shot real-world deployment in the peg insertion contact-rich manipulation task. As for future work, TacAlign can be generalized to a systemic sim-real pair sequence dataset for narrowing sim-to-real gap, covering robot control, IPC simulation and task-based calibration, rather than the task-specific discussion in our work. And the concept of general sim-real pair dataset of TacAlign will be extended and verified on more tasks. Besides, Tac2Real can be extended to RL training for dexterous hand manipulations or deformable objects, and it can also provide large-scale high-quality data for tactile-based world model (WM) or vision-language-action (VLA) training. Furthermore, we can consider using AI models to accelerate tactile simulation to further improve efficiency. Finally, the architecture of Tac2Real can be generalized to any physical sensor simulation, including tactile, acoustic, or even olfactory sensor simulation.

Acknowledgements

This work was supported by Shanghai Artificial Intelligence Laboratory.

References

1. Abad, A.C., Ranasinghe, A.: Visuotactile sensors with emphasis on gelsight sensor: A review. IEEE Sensors Journal **20**(14), 7628–7638 (2020)
2. Akinola, I., Xu, J., Carius, J., Fox, D., Narang, Y.: Tacsl: A library for visuotactile sensor simulation and learning. IEEE Transactions on Robotics (2025)
3. Bardenhagen, S.G., Kober, E.M.: The generalized interpolation material point method. Computer Modeling in Engineering & Sciences **5**(6), 477 (2004)
4. Blanco-Mulero, D., Barbany, O., Alcan, G., Colomé, A., Torras, C., Kyrki, V.: Benchmarking the sim-to-real gap in cloth manipulation. IEEE Robotics and Automation Letters **9**(3), 2981–2988 (2024)
5. Chen, W., Xu, J., Xiang, F., Yuan, X., Su, H., Chen, R.: General-purpose sim2real protocol for learning contact-rich manipulation with marker-based visuotactile sensors. IEEE Transactions on Robotics **40**, 1509–1526 (2024)
6. Chen, Z., Zhang, S., Luo, S., Sun, F., Fang, B.: Tacchi: A pluggable and low computational cost elastomer deformation simulator for optical tactile sensors. IEEE Robotics and Automation Letters **8**(3), 1239–1246 (2023)
7. De Vaucorbeil, A., Nguyen, V.P., Sinaie, S., Wu, J.Y.: Material point method after 25 years: Theory, implementation, and applications. Advances in applied mechanics **53**, 185–398 (2020)
8. Dhatt, G., Lefrançois, E., Touzot, G.: Finite element method. John Wiley & Sons (2012)
9. Du, W., Xu, W., Ren, J., Yu, Z., Lu, C.: Tacipc: Intersection-and inversion-free fem-based elastomer simulation for optical tactile sensors. IEEE Robotics and Automation Letters **9**(3), 2559–2566 (2024)
10. Ferguson, Z., Jain, P., Zorin, D., Schneider, T., Panozzo, D.: High-order incremental potential contact for elastodynamic simulation on curved meshes. In: ACM SIGGRAPH 2023 conference proceedings. pp. 1–11 (2023)
11. Gu, X., Wang, Y.J., Chen, J.: Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer. arXiv preprint arXiv:2404.05695 (2024)
12. Hansen, N.: The cma evolution strategy: a comparing review. Towards a new evolutionary computation: Advances in the estimation of distribution algorithms pp. 75–102 (2006)
13. Hietala, J., Blanco-Mulero, D., Alcan, G., Kyrki, V.: Closing the sim2real gap in dynamic cloth manipulation. arXiv preprint arXiv:2109.04771 **2** (2021)
14. Hu, Y., Li, T.M., Anderson, L., Ragan-Kelley, J., Durand, F.: Taichi: a language for high-performance computation on spatially sparse data structures. ACM Transactions on Graphics (TOG) **38**(6), 1–16 (2019)
15. Huang, K., Chitalu, F.M., Lin, H., Komura, T.: Gipc: Fast and stable gauss-newton optimization of ipc barrier energy. ACM Transactions on Graphics **43**(2), 1–18 (2024)
16. Huang, P., Zhang, X., Ma, S., Huang, X.: Contact algorithms for the material point method in impact and penetration simulation. International journal for numerical methods in engineering **85**(4), 498–517 (2011)
17. Jiang, C., Schroeder, C., Teran, J., Stomakhin, A., Selle, A.: The material point method for simulating continuum materials. In: Acm siggraph 2016 courses, pp. 1–52 (2016)
18. Josifovski, J., Malmir, M., Klarmann, N., Žagar, B.L., Navarro-Guerrero, N., Knoll, A.: Analysis of randomization effects on sim2real transfer in reinforcement learning

for robotic manipulation tasks. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10193–10200. IEEE (2022)
19. Lan, L., Yang, Y., Kaufman, D., Yao, J., Li, M., Jiang, C.: Medial ipc: accelerated incremental potential contact with medial elastics. ACM Transactions on Graphics **40**(4) (2021)
20. Li, M., Ferguson, Z., Schneider, T., Langlois, T., Zorin, D., Panozzo, D., Jiang, C., Kaufman, D.M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. ACM transactions on graphics (2020)
21. Li, Y., Du, W., Yu, C., Li, P., Zhao, Z., Liu, T., Jiang, C., Zhu, Y., Huang, S.: Taccel: Scaling up vision-based tactile robotics via high-performance gpu simulation. arXiv preprint arXiv:2504.12908 (2025)
22. Ma, J., Wang, D., Randolph, M.: A new contact algorithm in the material point method for geotechnical simulations. International Journal for Numerical and Analytical Methods in Geomechanics **38**(11), 1197–1210 (2014)
23. Makoviichuk, D., Makoviychuk, V.: rl-games: A high-performance framework for reinforcement learning. https://github.com/Denys88/rl_games (May 2021)
24. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al.: Isaac gym: High performance gpu-based physics simulation for robot learning. arXiv preprint arXiv:2108.10470 (2021)
25. Mittal, M., Roth, P., Tigue, J., Richard, A., Zhang, O., Du, P., Serrano-Muñoz, A., Yao, X., Zurbrügg, R., Rudin, N., Wawrzyniak, L., Rakhsha, M., Denzler, A., Heiden, E., Borovicka, A., Ahmed, O., Akinola, I., Anwar, A., Carlson, M.T., Feng, J.Y., Garg, A., Gasoto, R., Gulich, L., Guo, Y., Gussert, M., Hansen, A., Kulkarni, M., Li, C., Liu, W., Makoviychuk, V., Malczyk, G., Mazhar, H., Moghani, M., Murali, A., Noseworthy, M., Poddubny, A., Ratliff, N., Rehberg, W., Schwarke, C., Singh, R., Smith, J.L., Tang, B., Thaker, R., Trepte, M., Wyk, K.V., Yu, F., Millane, A., Ramasamy, V., Steiner, R., Subramanian, S., Volk, C., Chen, C., Jawale, N., Kuruttukulam, A.V., Lin, M.A., Mandlekar, A., Patzwaldt, K., Welsh, J., Zhao, H., Anes, F., Lafleche, J.F., Moënne-Loccoz, N., Park, S., Stepinski, R., Gelder, D.V., Amevor, C., Carius, J., Chang, J., Chen, A.H., de Heras Ciechomski, P., Daviet, G., Mohajerani, M., von Muralt, J., Reutskyy, V., Sauter, M., Schirm, S., Shi, E.L., Terdiman, P., Vilella, K., Widmer, T., Yeoman, G., Chen, T., Grizan, S., Li, C., Li, L., Smith, C., Wiltz, R., Alexis, K., Chang, Y., Chu, D., Fan, L.J., Farshidian, F., Handa, A., Huang, S., Hutter, M., Narang, Y., Pouya, S., Sheng, S., Zhu, Y., Macklin, M., Moravanszky, A., Reist, P., Guo, Y., Hoeller, D., State, G.: Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. arXiv preprint arXiv:2511.04831 (2025), https://arxiv.org/abs/2511.04831
26. Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M.I., et al.: Ray: A distributed framework for emerging {AI} applications. In: 13th USENIX symposium on operating systems design and implementation (OSDI 18). pp. 561–577 (2018)
27. Qureshi, M.N., Garg, S., Yandun, F., Held, D., Kantor, G., Silwal, A.: Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 6502–6509. IEEE (2025)
28. Shen, X., Cai, R., Bi, M., Lv, T.: Preconditioned nonlinear conjugate gradient method for real-time interior-point hyperelasticity. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024)

29. Si, Z., Zhang, G., Ben, Q., Romero, B., Xian, Z., Liu, C., Gan, C.: Difftactile: A physics-based differentiable tactile simulator for contact-rich robotic manipulation. arXiv preprint arXiv:2403.08716 (2024)
30. Sulsky, D., Kaul, A.: Implicit dynamics in the material-point method. Computer Methods in Applied Mechanics and Engineering **193**(12-14), 1137–1170 (2004)
31. Sun, Y., Zhang, S., Chen, Z., Shen, Z., Sun, F., Stefanini, C., Guo, D., Luo, S., Zhang, J., Shan, J., et al.: Soft contact simulation and manipulation learning of deformable objects with vision-based tactile sensor. IEEE Transactions on Automation Science and Engineering (2025)
32. Sun, Y., Zhang, S., Li, W., Zhao, J., Shan, J., Shen, Z., Chen, Z., Sun, F., Guo, D., Fang, B.: Tacchi 2.0: A low computational cost and comprehensive dynamic contact simulator for vision-based tactile sensors. arXiv preprint arXiv:2503.09100 (2025)
33. Tegin, J., Wikander, J.: Tactile sensing in intelligent robotic manipulation–a review. Industrial Robot: An International Journal **32**(1), 64–70 (2005)
34. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5026–5033. IEEE (2012). https://doi.org/10.1109/IROS.2012.6386109
35. Wang, S., Lambeta, M., Chou, P.W., Calandra, R.: Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. IEEE Robotics and Automation Letters **7**(2), 3930–3937 (2022)
36. Wang, T., Zheng, P., Li, S., Wang, L.: Multimodal human–robot interaction for human-centric smart manufacturing: a survey. Advanced Intelligent Systems **6**(3), 2300359 (2024)
37. Xu, J., Chen, T., Zlokapa, L., Foshey, M., Matusik, W., Sueda, S., Agrawal, P.: An end-to-end differentiable framework for contact-aware robot design. arXiv preprint arXiv:2107.07501 (2021)
38. Xu, J., Kim, S., Chen, T., Garcia, A.R., Agrawal, P., Matusik, W., Sueda, S.: Efficient tactile simulation with differentiability for robotic manipulation. In: Conference on Robot Learning. pp. 1488–1498. PMLR (2023)
39. Xue, H., Ren, J., Chen, W., Zhang, G., Fang, Y., Gu, G., Xu, H., Lu, C.: Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. arXiv preprint arXiv:2503.02881 (2025)
40. Yousef, H., Boukallel, M., Althoefer, K.: Tactile sensing for dexterous in-hand manipulation in robotics—a review. Sensors and Actuators A: physical **167**(2), 171–187 (2011)
41. Yu, K., Han, Y., Wang, Q., Saxena, V., Xu, D., Zhao, Y.: Mimictouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. arXiv preprint arXiv:2310.16917 (2023)
42. Yuan, W., Li, R., Srinivasan, M.A., Adelson, E.H.: Measurement of shear and slip with a gelsight tactile sensor. In: 2015 IEEE international conference on robotics and automation (ICRA). pp. 304–311. IEEE (2015)
43. Yuan, W., Srinivasan, M.A., Adelson, E.H.: Estimating object hardness with a gelsight touch sensor. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 208–215. IEEE (2016)
44. Zhang, C., Cui, S., Hu, J., Jiang, T., Zhang, T., Wang, R., Wang, S.: Tacflex: Multi-mode tactile imprints simulation for visuotactile sensors with coating patterns. IEEE Transactions on Robotics (2025)
45. Zhang, K., Shen, S.L., Wu, H., Zhou, A.: Nitsche-based material point method for large deformation frictional contact problems. Computational Particle Mechanics **12**(2), 947–970 (2025)

46. Zhao, H., Zeng, C., Zhuang, L., Zhao, Y., Xue, S., Wang, H., Zhao, X., Li, Z., Li, K., Huang, S., et al.: High-fidelity simulated data generation for real-world zero-shot robotic manipulation learning with gaussian splatting. IEEE Robotics and Automation Letters (2026)
47. Zhao, J., Kuppuswamy, N., Feng, S., Burchfiel, B., Adelson, E.: Polytouch: A robust multi-modal tactile sensor for contact-rich manipulation using tactile-diffusion policies. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 104–110. IEEE (2025)
48. Zhao, Z., Dong, H., He, Z., Li, Y., Yi, X., Li, Z.: Closing the reality gap: Zero-shot sim-to-real deployment for dexterous force-based grasping and manipulation. arXiv preprint arXiv:2601.02778 (2026)
49. Zhong, S., Albini, A., Maiolino, P., Posner, I.: Tactgen: tactile sensory data generation via zero-shot sim-to-real transfer. IEEE Transactions on Robotics **41**, 1316–1328 (2024)
50. Zienkiewicz, O.C., Taylor, R.L., Nithiarasu, P., Zhu, J.: The finite element method, vol. 3. Elsevier (1977)