Taccel: Scaling Up Vision-based Tactile Robotics via High-performance GPU Simulation

Yuyang Li[1,2⋆], Wenxin Du[3⋆], Chang Yu[3⋆], Puhao Li[2], Zihang Zhao[1], Tengyu Liu[2], Chenfanfu Jiang[3†], Yixin Zhu[1†], Siyuan Huang[2†]
[⋆]Equal contributor [†]Corresponding author
[1] Institute for AI, Peking University [2] State Key Lab of General AI, Beijing Institute for General AI
[3] AIVC Laboratory, University of California, Los Angeles
taccel-simulator.github.io

Abstract

Tactile sensing is crucial for achieving human-level robotic capabilities in manipulation tasks [52]. Vision-based tactile sensors (VBTSs) have emerged as a promising solution, offering high spatial resolution and cost-effectiveness by sensing contact through camera-captured deformation patterns of elastic gel pads [59, 35]. However, these sensors' complex physical characteristics and visual signal processing requirements present unique challenges for robotic applications. The lack of efficient and accurate simulation tools for VBTSs has significantly limited the scale and scope of tactile robotics research [49, 8, 48]. Here we present **Taccel**, a high-performance simulation platform that integrates Incremental Potential Contact (IPC) and Affine Body Dynamics (ABD) to model robots, tactile sensors, and objects with both accuracy and unprecedented speed, achieving an 18-fold acceleration over real-time across thousands of parallel environments. Unlike previous simulators that operate at sub-real-time speeds with limited parallelization, **Taccel** provides precise physics simulation and realistic tactile signals while supporting flexible robot-sensor configurations through user-friendly APIs. Through extensive validation in object recognition, robotic grasping, and articulated object manipulation, we demonstrate precise simulation and successful sim-to-real transfer. These capabilities position **Taccel** as a powerful tool for scaling up tactile robotics research and development. By enabling large-scale simulation and experimentation with tactile sensing, **Taccel** accelerates the development of more capable robotic systems, potentially transforming how robots interact with and understand their physical environment.

1 Introduction

The ability to physically interact with the environment through touch is fundamental to robotic manipulation [3, 13]. While vision provides global scene understanding, tactile sensing captures crucial local contact information [55] essential for precise manipulation. Among various tactile sensing technologies [61, 24, 36, 25], vision-based tactile sensors (VBTSs) such as GelSight [59] and 9DTact [35] have emerged as a central focus in tactile research. Their ability to provide high-resolution tactile feedback through camera-captured deformation patterns of elastic gel pads, combined with cost-effectiveness, has driven significant advances in robotics [63, 33, 44, 8, 62].

The primary challenge in scaling up VBTS-equipped robot simulation lies in accurately modeling the hyperelastic soft gel pad and its contact [63, 12]. Current approaches follow two main directions: rigid-body approximations [49, 54] and soft-body simulations [44, 8, 12, 20, 28, 63]. While rigid-body methods efficiently support basic tasks like pick-and-place [1, 49], they cannot capture the fine-grained contactsd and elastomer deformations essential for complex manipulation tasks [63, 12]

Preprint. Under review.

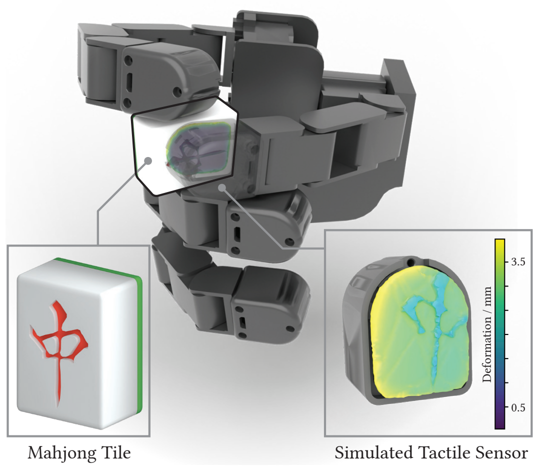


Figure 1: **`Taccel` demonstration of dexterous manipulation with tactile feedback.** The simulation shows an Allegro robotic hand equipped with four VBTSs performing a precision grasp on a mahjong tile. The sensor on the thumb actively presses against the tile's face, while other fingers maintain a stable grip. The left inset shows the mahjong tile with its characteristic Chinese character marking. The right inset displays the tactile sensor's output as a depth map (green-yellow colormap, scale in millimeters), where brighter regions indicate deeper deformation of the gel pad. This depth information precisely captures the geometric features of the tile's surface, demonstrating the simulator's capability to generate realistic tactile feedback during complex manipulation tasks.

Table 1: **Comprehensive comparison of FEM-based VBTS simulators. Soft Mat.**: modeling of deformable materials (FEM: Finite Element Methods). **Stiff Mat.**: modeling of stiff materials (Rigid: traditional rigid body, ABD: Affine Body Dynamics, MPM: Material Point Methods, PBD: Position-based Dynamics). Contact: collision handling method (Virtual: approximated contact, Penalty: penalty-based, IPC: Incremental Potential Contact). **RGB Signal**: RGB tactile pattern generation method (Look-up: look-up tables, DNN: Deep Neural Network, ✗: not supported). **Robot**: range of supported robotic systems. The last two columns report parallel simulation capabilities (**# Env**: maximum number of parallel environments) and simulation speed relative to real-time in a peg insertion simulation, with dual sensors in low/high resolutions, measured on an NVIDIA H100 80G GPU. Details are provided in Fig. 4.

Simulator	Soft Mat.	Stiff Mat.	Contact	RGB Signal	Robot	# Env ↑	Sim Speed ↑
Taxim [43]	-	Rigid	Virtual	Look-up	Sensor	1	-
DiffTactile [44]	FEM	MPM/PBD	Penalty	DNN	Gripper	1	-
SAPIEN-IPC [48]	FEM	**ABD**	**IPC**	✗	Gripper	256 / *4*	0.81× / 0.03×
`Taccel` (Ours)	FEM	**ABD**	**IPC**	**DNN**	**Any**	***4096 / 64***	***18.30*× / 0.25×**

and detailed force distribution analysis [38, 44]. Soft-body simulations offer higher fidelity but face significant computational challenges that limit their practical application in large-scale experiments.

An ideal VBTS simulator must simultaneously achieve:

- **Precision**: precise modeling of robots, sensors, and objects with physically valid solutions, particularly maintaining inversion-free and intersection-free states during complex contact interactions; generation of realistic tactile signals across multiple resolutions, from high-resolution RGB patterns and depth maps to low-resolution marker movements.
- **Scalability**: Capability for large-scale parallel simulation for extensive data generation and algorithm development.
- **Flexibility**: Support for diverse robotic platforms and sensor configurations, from parallel grippers to multi-finger hands with varying sensor arrangements.

As detailed in Tab. 1, existing solutions often compromise on precision, scalability, or flexibility. They typically produce suboptimal physics, operate slower than real-time with limited parallel environments, or focus on specific sensor setups or simple grippers. These limitations significantly impede the broader application of tactile robotics.

To address these challenges, we present **`Taccel`**, a high-performance simulation platform for scaling up robots with VBTS-integration. Built on state-of-the-art simulation techniques (Sec. 3), **`Taccel`** provides dedicated components for simulating robots (Sec. 4), tactile sensors (Sec. 5), and tactile signal generation (Sec. 6). Comprehensive evaluations (Sec. 7) demonstrate its effectiveness across all objectives:

- **Precision**: `Taccel` leverages advanced solid material simulation techniques, combining IPC [30] and ABD [29] to ensure physical accuracy. The platform models gel pads using neo-Hookean solids with IPC guaranteeing inversion- and intersection-free contact solutions, while integrating ABD for efficient and precise simulation of robot links and stiff objects. This combination enables precise physics simulation and support realistic tactile signal generation.
- **Scalability**: With an efficient implementation of the ABD-IPC system using NVIDIA warp [39], `Taccel` achieves unprecedented parallelization. On a single H100 GPU, it reaches over 900 FPS (4096 environments, $18\times$ wallclock time) for a peg-insertion task with dual sensors and 12.67 FPS (256 environments, $0.25\times$ wallclock time) for a dexterous manipulation scenario with full-hand tactile sensing.
- **Flexibility**: `Taccel` provides user-friendly APIs for seamless integration of diverse robotic platforms and sensor configurations. Users can easily load and configure robots through Unified Robot Description Format (URDF) with auxiliary configurations, supporting applications from simple grippers to complex manipulation tasks, like the mahjong tile sensing task in Fig. 1.

We validate `Taccel` through three fundamental tactile-informed robotic tasks (Sec. 8). In object classification, models trained solely on `Taccel`'s synthetic tactile signals demonstrate strong generalization to real-world data without adaptation. In grasping experiments across four robotic hand designs, we showcase the platform's versatility in handling diverse robot configurations and tactile signal types. In articulated object manipulation tasks, we demonstrate `Taccel`'s physical fidelity through close correspondence between simulated and real-world robot behavior.

Our key contributions include: (i) development of a high-performance simulation platform combining precise physics modeling, realistic tactile signal generation, and massive parallelization; (ii) implementation of user-friendly APIs enabling flexible robot-sensor integration and high-fidelity tactile signal synthesis; (iii) comprehensive evaluation of the platform's precision and scalability; and (iv) extensive experimental validation across diverse tactile robotics applications. By enabling large-scale, high-fidelity simulation of VBTS-equipped robots, `Taccel` aims to accelerate future research in tactile robotics.

2 Related Work

2.1 Robot Tactile Sensors

Tactile sensing plays a fundamental role in precise manipulation, as established by neuroscientific studies [52, 27, 26, 3]. This understanding has driven the development of artificial tactile sensing systems for robots [41]. Among these, VBTSs have gained prominence by offering high-resolution sensing with cost-effectiveness and operational simplicity [59, 51, 35, 33]. While these sensors have advanced robotic manipulation [42, 38, 63], their development remains constrained by the reliance on physical hardware experimentation. `Taccel` addresses this limitation by providing a comprehensive simulation platform to accelerate research and development in tactile robotics.

2.2 Simulating VBTSs

Early VBTS simulators focused on normal deformation scenarios, approximating hyperelastic behavior through geometric computations and surface modifications [17, 49, 54, 1, 43]. While efficient in generating high-resolution tactile signals through physics-based rendering, these approaches inadequately capture elastomer dynamics during complex manipulation tasks, especially those involving tangential forces and continuous interactions [63].

Recent approaches have achieved higher physical fidelity by incorporating advanced solid material simulation techniques. Methods using Material Point Methods (MPM) [46, 21, 22] and Finite Element Methods (FEM) [30, 29] better model elastomer properties through time-integrated deformation computations [10, 8, 44, 12]. Notable improvements include the adoption of IPC [30] by several simulators [12, 8], providing robust contact handling with guaranteed inversion- and intersection-free solutions. Tab. 1 compares key features of representative approaches.

Taccel builds on these advances by combining IPC and ABD in a unified platform, achieving both physical accuracy and computational efficiency while supporting diverse robot configurations and enabling large-scale parallel simulation for robot learning applications.

2.3 Tactile-Informed Robotic Tasks

Tactile sensing enhances robotic capabilities across three fundamental domains through precise contact interaction measurements:

Perception Tactile feedback enables sophisticated object understanding through contact-based sensing. Applications include shear and slip detection [60, 11], object classification and pose estimation [32, 56, 47, 2], material property inference [18, 23], and interaction reconstruction [47, 58, 55]. These perceptual capabilities form the foundation for advanced manipulation algorithms.

Grasping Stable grasping requires precise control of contact forces to balance external loads [15, 45]. Tactile sensing provides direct force-torque feedback essential for diverse grasping strategies [37, 34, 57]. This tactile information complements vision-based approaches by enabling fine-grained contact monitoring and in-hand adjustments [6, 5].

Manipulation Tactile feedback enables complex manipulation beyond basic pick-and-place operations. Applications include precision tasks like peg insertion [8], object pivoting [20], and articulated object manipulation [64, 3]. Systems such as Tac-Man [63] and DoorBot [50] demonstrate how tactile sensing guides contact geometry understanding and articulation control. This sensing modality is particularly crucial for high-frequency object tracking during dexterous manipulation.

We validate **Taccel**'s capabilities through three representative applications: (i) multi-platform robotic grasping with both rigid and soft objects, (ii) object classification using purely synthetic training data with strong real-world transfer, and (iii) articulated object manipulation including drawers, cabinets, and bolt-nut assembly tasks, extending the Tac-Man framework [63].

3 Unified IPC Simulation in Taccel

This section presents the unified IPC simulation framework in **Taccel**, detailing its mathematical foundations and implementation principles. For complete derivations, we refer readers to the original IPC [30] and ABD [29] works.

3.1 Problem Formulation and Soft Body Dynamics

We consider n_s tetrahedralized soft bodies discretized into N_s vertices with positions $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{N_s}$ in Cartesian space. The system state is represented by the stacked position vector $\mathbf{x} = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, ..., \boldsymbol{x}_{N_s}^T]^T \in \mathbb{R}^{3N_s}$. Following Lagrangian mechanics, we express the system's Lagrangian as $L(\mathbf{x}, \dot{\mathbf{x}}) = T(\mathbf{x}, \dot{\mathbf{x}}) - V(\mathbf{x})$, where $T(\mathbf{x}, \dot{\mathbf{x}}) = \frac{1}{2}\dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}$ represents kinetic energy with mass matrix $\mathbf{M} \in \mathbb{R}^{3N_s \times 3N_s}$. The potential energy $V(\mathbf{x})$ comprises two terms: an elastic energy $\Phi(\mathbf{x})$ utilizing the Neo-Hookean constitutive model for hyperelastic materials (characterized by Young's modulus $\mathcal{E}$ and Poisson's ratio ν), and external forces $E_{\text{ext}}(\mathbf{x})$. The elastic energy is defined as $\Phi(\mathbf{x}) = \int_\Omega \Psi(\mathbf{x}) d\mathbf{x}$, where $\Psi(\mathbf{x})$ denotes elastic energy density over the volume region Ω of all objects in rest configuration.

3.2 Time Stepping

Substituting $L(\mathbf{x}, \dot{\mathbf{x}})$ into the Euler-Lagrange equation $\frac{\partial L}{\partial \mathbf{x}}(\mathbf{x}, \dot{\mathbf{x}}) - \frac{d}{dt}\frac{\partial L}{\partial \dot{\mathbf{x}}}(\mathbf{x}, \dot{\mathbf{x}}) = 0$ yields the governing dynamics:

$$\mathbf{M}\ddot{\mathbf{x}} = -\frac{dV}{d\mathbf{x}}(\mathbf{x}). \tag{1}$$

We temporally discretize Eq. (1) using backward Euler:

$$\frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{\Delta t} = \dot{\mathbf{x}}^{n+1}, \quad \frac{\mathbf{M}(\dot{\mathbf{x}}^{n+1} - \dot{\mathbf{x}}^n)}{\Delta t} = -\frac{dV}{d\mathbf{x}}(\mathbf{x}^{n+1}), \tag{2}$$

where time is discretized into steps $\{t_n = n\Delta t : n \in \mathbb{N}\}$ with step size $\Delta t > 0$, and $\mathbf{x}^n = \mathbf{x}(t_n)$. Under this discretization, Eq. (1) can be formulated as:

$$\frac{d}{d\mathbf{x}}\left(E_{\text{IP}}(\mathbf{x}^n)\right) = 0. \tag{3}$$

If we define the incremental potential energy of the constrained system as:

$$E_{\text{IP}}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^n - \Delta t\dot{\mathbf{x}}^n)^T \mathbf{M}(\mathbf{x} - \mathbf{x}^n - \Delta t\dot{\mathbf{x}}^n) + \Delta t^2 V(\mathbf{x}), \tag{4}$$

then the general simulation problem in a conservative system can be reformulated as the minimization problem:

$$\mathbf{x}^{n+1} = \arg\min_{\mathbf{x}} E_{\text{IP}}(\mathbf{x}). \tag{5}$$

3.3 Frictional Contact

We employ IPC [30] to handle contact interactions. The method operates on surface contact pairs $\mathcal{B}$, comprising point-triangle and edge-edge pairs from the surface meshes of soft and affine objects. For each contact pair $k \in \mathcal{B}$ with distance $d_k > 0$, IPC defines two key energy terms. First, a barrier energy that prevents interpenetration:

$$b(d_k(\mathbf{x})) = -\left(d_k - \hat{d}\right)^2 \log\left(\frac{d_k}{\hat{d}}\right) I_{\{d_k \in (0,\hat{d})\}}(d_k), \tag{6}$$

where $\hat{d} > 0$ is the distance threshold for contact force activation and $I(\cdot)$ is the indicator function.

Second, an approximated friction potential energy:

$$D_k(\mathbf{x}, \mathbf{x}^n) = \mu \lambda_k^n f_0(\|\boldsymbol{u}_k\|), \tag{7}$$

where $\mathbf{x}^n$ represents the configuration at the previous timestep t_n, λ_k^n is the magnitude of the lagged normal contact force, and $\boldsymbol{u}_k \in \mathbb{R}^2$ denotes the tangential relative displacement in the local contact frame. The friction transition function $f_0(x) = \int_{\epsilon_v \Delta t}^{x} f_1(y)dy + \epsilon_v \Delta t$ uses:

$$f_1(y) = \begin{cases} -\frac{y^2}{\epsilon_v^2 \Delta t^2} + \frac{2y}{\epsilon_v \Delta t}, & y \in (0, \Delta t\epsilon_v), \\ 1, & y \geqslant \Delta t \epsilon_v, \end{cases} \tag{8}$$

where $\epsilon_v > 0$ serves as a velocity threshold distinguishing between static and dynamic friction regimes. These contact and friction terms augment our incremental potential energy:

$$E_{\text{IPC}}(\mathbf{x}) = E_{\text{IP}}(\mathbf{x}) + \Delta t^2 B(\mathbf{x}) + \Delta t^2 D(\mathbf{x}, \mathbf{x}^n), \tag{9}$$

with $B(\mathbf{x}) = \kappa \sum_{k \in \mathcal{B}} A_k b(d_k(\mathbf{x}))$, $D(\mathbf{x}, \mathbf{x}^n) = \sum_{k \in \mathcal{B}} D_k(\mathbf{x}, \mathbf{x}^n)$, where $\kappa > 0$ controls contact stiffness.

3.4 ABD and Unified Simulation

For n_a affine bodies, we introduce a reduced coordinate space $\mathbf{y} \in \mathbb{R}^{12n_a}$ with an embedding map $\phi : \mathbb{R}^{12n_a} \to \mathbb{R}^{3N_a}$ that projects reduced coordinates to full-space vertices $\phi(\mathbf{y})$ [29], where N_a denotes the total vertex count of affine bodies' surface meshes. Each affine body uses 12 Degree of Freedom (DoF): three for translation ($\mathbb{R}^3$) and nine for affine deformation ($\mathbb{R}^{3\times 3}$).

$$\begin{aligned} T(\mathbf{y}, \dot{\mathbf{y}}) &= \frac{1}{2}\dot{\mathbf{x}}^T \mathbf{M}\dot{\mathbf{x}} = \frac{1}{2}\dot{\phi}(\mathbf{y})^T \mathbf{M}\dot{\phi}(\mathbf{y}) \\ &= \frac{1}{2}(\mathbf{J}\dot{\mathbf{y}})^T \mathbf{M}(\mathbf{J}\dot{\mathbf{y}}) = \frac{1}{2}\dot{\mathbf{y}}^T(\mathbf{J}^T\mathbf{M}\mathbf{J})\dot{\mathbf{y}} = \frac{1}{2}\dot{\mathbf{y}}^T \mathbf{M}^{\mathbf{y}}\dot{\mathbf{y}}, \end{aligned} \tag{10}$$

where $\mathbf{J} = \frac{\partial \phi}{\partial \mathbf{y}} \in \mathbb{R}^{3N_a \times 12n_a}$ is the Jacobian, $\mathbf{M}$ is the full-space mass matrix, and $\mathbf{M}^{\mathbf{y}} = \mathbf{J}^T\mathbf{M}\mathbf{J}$ is the reduced-space mass matrix. The potential energy $V(\mathbf{y})$ includes an As-Rigid-As-Possible (ARAP) term $\Phi^{\mathbf{y}}(\mathbf{y}) = \Phi^{\mathbf{x}}(\phi(\mathbf{x}))$ with high stiffness κ_s to limit deformation, plus external forces $E_{\text{ext}}(\mathbf{y})$.

Combining with Eq. (9), we obtain the unified affine-deformable coupled IPC energy [9] for the full system state $\{\mathbf{y}; \mathbf{x}\} \in \mathbb{R}^{12n_a + 3N_s}$:

$$E_{\text{IPC}}(\mathbf{y}; \mathbf{x}) = E_{\text{IP}}(\mathbf{x}) + E_{\text{IP}}(\mathbf{y}) + \Delta t^2 B(\phi(\mathbf{y}); \mathbf{x}) + \Delta t^2 D(\phi(\mathbf{y}); \mathbf{x}, \phi(\mathbf{y}^n); \mathbf{x}^n), \tag{11}$$

where $E_{\text{IP}}(\mathbf{y})$ is defined as:

$$E_{\text{IP}}(\mathbf{y}) = \frac{1}{2}(\mathbf{y} - \mathbf{y}^n - \Delta t \dot{\mathbf{y}}^n)^T \mathbf{M}^{\mathbf{y}} (\mathbf{y} - \mathbf{y}^n - \Delta t \dot{\mathbf{y}}^n) + \Delta t^2 V(\mathbf{y}). \tag{12}$$

The next timestep's configuration follows from minimizing this barrier-augmented incremental potential:

$$\mathbf{y}^{n+1}; \mathbf{x}^{n+1} = \arg\min_{\mathbf{y};\mathbf{x}} E_{\text{IPC}}(\mathbf{y}; \mathbf{x}). \tag{13}$$

3.5 Kinematic Constraints

We express kinematic constraints as $\mathbf{S}^{\mathbf{x}}\mathbf{x} = \mathbf{s}^{\mathbf{x}}$ and $\mathbf{S}^{\mathbf{y}}\mathbf{y} = \mathbf{s}^{\mathbf{y}}$, where $\mathbf{S}^{\mathbf{x}} \in \mathbb{R}^{c^{\mathbf{x}} \times 3N_s}$, $\mathbf{s}^{\mathbf{x}} \in \mathbb{R}^{3N_s}$ for soft bodies, and $\mathbf{S}^{\mathbf{y}} \in \mathbb{R}^{c^{\mathbf{y}} \times 12n_a}$, $\mathbf{s}^{\mathbf{y}} \in \mathbb{R}^{12n_a}$ for affine bodies. To enforce these constraints, we employ the Augmented Lagrangian method by augmenting E_{IPC} to:

$$E_{\text{IPC}}^{\text{AL}}(\mathbf{y}; \mathbf{x}) = E_{\text{IPC}}(\mathbf{y}; \mathbf{x}) + \|(\mathbf{S}^{\mathbf{x}}\mathbf{x} - \mathbf{s}^{\mathbf{x}})^T \lambda^{\mathbf{x}}\|_2^2 + \|(\mathbf{S}^{\mathbf{y}}\mathbf{y} - \mathbf{s}^{\mathbf{y}})^T \lambda^{\mathbf{y}}\|_2^2, \tag{14}$$

where $\lambda^{\mathbf{x}} \in \mathbb{R}^{c^{\mathbf{x}}}$ and $\lambda^{\mathbf{y}} \in \mathbb{R}^{c^{\mathbf{y}}}$ are Lagrangian multipliers.

Optimizing $E_{\text{IPC}}^{\text{AL}}(\mathbf{y}; \mathbf{x})$ yields the solution to the constrained system:

$$\mathbf{y}^{n+1}; \mathbf{x}^{n+1} = \arg\min_{\mathbf{y};\mathbf{x}} E_{\text{IPC}}(\mathbf{y}; \mathbf{x}), \tag{15}$$

$$\text{s.t.} \quad \mathbf{S}^{\mathbf{x}}\mathbf{x} = \mathbf{s}^{\mathbf{x}} \quad \text{and} \quad \mathbf{S}^{\mathbf{y}}\mathbf{y} = \mathbf{s}^{\mathbf{y}}. \tag{16}$$

4 Robot and VBTS Simulation in `Taccel`

Building upon our unified simulation framework, **`Taccel`** implements robot and VBTS simulation through a modular design that leverages the complementary strengths of affine and soft-body dynamics. Robot links are efficiently modeled as affine bodies to capture their primarily rigid motion, while VBTSs are simulated as soft bodies to accurately represent their deformation mechanics. This natural division allows **`Taccel`** to balance computational efficiency with physical accuracy while maintaining consistent contact handling through IPC.

Robot Modeling For a robot with D-DoF, L links, and N integrated vision-based tactile sensors (VBTSs), **`Taccel`** constructs its kinematic model from a URDF specification, loading visual and collision meshes as affine bodies (Sec. 3.4). Given the robot's global transformation T_r and joint configuration $q \in \mathcal{Q}$, the transformation of each link j in the world frame, ${}^{r}_{l_j}T(q)$, is computed via forward kinematics.

Tactile Sensors Modeling Each VBTS is represented by a tetrahedral mesh as a soft volumetric body (Sec. 3), with attachment specifications to its corresponding robot link. For the i-th sensor's gel pad $\mathcal{G}_i$ attached to link l_j with local transformation ${}^{l_j}_{\mathcal{G}_i}T$, we denote its outer surface as $\mathcal{S}_i = \partial\mathcal{G}$. The surface comprises a reflective-coated region $\partial^+\mathcal{G}_i$ and a sensor-attached region $\partial^-\mathcal{G}$. Within $\partial^+\mathcal{G}_i$, m_i markers are positioned at locations ${}^{\mathcal{G}_i}\mathbf{P}_i = \{p_k^{(i)} \in \mathbb{R}^3, k = 1, \ldots, m_i\}_{i=1}^N$, each defined by barycentric coordinates in its triangle:

$$p_k^{(i)} = \sum_{u=1}^{3} \alpha_u \mathbf{x}_u^{(i,k)}, \text{ where } \sum_{u=1}^{3} \alpha_u = 1, \alpha_u \in [0, 1]. \tag{17}$$

Robot-Sensor Simulation Scene initialization computes affine states through forward kinematics for robot links and explicit specification for stiff objects. Gel pad node positions are transformed to the world frame, with all states written to $\mathbf{x}$ and $\mathbf{y}$ and velocities $\dot{\mathbf{x}}, \dot{\mathbf{y}}$ initialized to zero. Robot actions are implemented through kinematic constraints (Eq. (15)). From joint space targets, we compute affine state targets for links and node position targets for gel pad attached surfaces $\partial^-\mathcal{G}$, assembled into vectors $\mathbf{s}^{\mathbf{x}}, \mathbf{s}^{\mathbf{y}}$. Selection matrices $\mathbf{S}^{\mathbf{x}}, \mathbf{S}^{\mathbf{y}}$ apply these constraints (Eq. (16)), with remaining states solved via time stepping (Sec. 3.2).

5 Tactile Signal Simulation in **Taccel**

Contact interactions during simulation cause gel pad deformation, transforming the coated surface $\partial^+\mathcal{G}$ to $\partial^+\tilde{\mathcal{G}}$ and marker positions to $\tilde{\mathbf{P}}_i$. These deformed quantities serve as the foundation for generating multiple types of tactile signals, each suited for different robotic applications.

5.1 High-resolution Signals

High-resolution tactile signals, including RGB images and depth maps, are essential when fine-grained details like object texture and local geometries are required. While many works directly use the object geometry in the contact region as a depth map for efficiency [49], this approach can lead to inauthentic signals in dynamic scenarios. Instead, **Taccel** supports accurate simulation of high-resolution soft volumetric bodies to fully capture the fine-grained contact and deformation patterns.

The signal generation process proceeds in two stages. First, we extract the depth and normal maps $d_{(u,v)}, n_{(u,v)}$ for pixel coordinates (u, v) from the deformed coated surface $\partial^+\tilde{\mathcal{G}}$. Next, following the method of Si *et al.* [44], we apply a Deep Neural Network (DNN) to generate RGB tactile signals from the depth information. Specifically, for each pixel coordinate (u, v), a pixel-to-pixel DNN parameterized by θ maps the inputs to the pixel color relative to a reference image: $f_\theta\left(\gamma(u, v), n_{(u,v)}\right) \mapsto \Delta\sigma_{(u,v)}$. The result is added to a reference image (the RGB signal when the gel pad has no deformation) to obtain the final RGB image. Here, $\gamma(\cdot, \cdot)$ provides the 2D positional encoding of the pixel coordinate. The model is trained on patches from 200 real robot tactile images and their corresponding depth map annotations via the pin-pressing procedure [59, 44].

We demonstrate **Taccel**'s capability to scale up synthetic data generation of high-resolution tactile signals through robotic grasping simulations and explore downstream object recognition model learning in Secs. 8.1 and 8.2.

5.2 Low-resolution signals

Low-resolution tactile signals primarily track marker positions and their movements. The deformed markers can be tracked by computing their new positions throughout the simulation (at time step t):

$$\tilde{p}_k^{(i)}(t) = \sum_{u=1}^{3} \alpha_u \tilde{\mathbf{x}}_u^{(i,k)}(t), \tag{18}$$

and projecting them on the tactile image. The marker flows, representing local deformation patterns, are then computed as:

$$\Delta\mathbf{P}_i = \{\tilde{p}_k^{(i)} - p_k^{(i)}\} = \{\Delta p_k^{(i)}\}. \tag{19}$$

5.3 Tactile Signals in 3D

The depth map and marker positions can be transformed into a dense or sparse 3D point cloud in the world frame using robot kinematics and sensor configurations. These 3D tactile signals provide crucial spatial information for robotic manipulation tasks. We demonstrate their effectiveness through the simulation of Tac-Man framework [63] in Sec. 8.3.

6 API Designs in **Taccel**

Taccel provides intuitive Python APIs designed to make tactile robotics simulation accessible to researchers while maintaining high performance through NVIDIA Warp [39] GPU acceleration. The APIs allows for seamless loading of robots from URDF files with automatic parsing, sensor configurations from auxiliary files, and objects from mesh files. Users can efficiently reset simulation states or setting kinematic targets to control the robots in familiar formats (NumPy arrays, PyTorch tensors). Further, **Taccel**'s architecture supports parallel simulation of multiple environments, isolated via unique environment IDs.

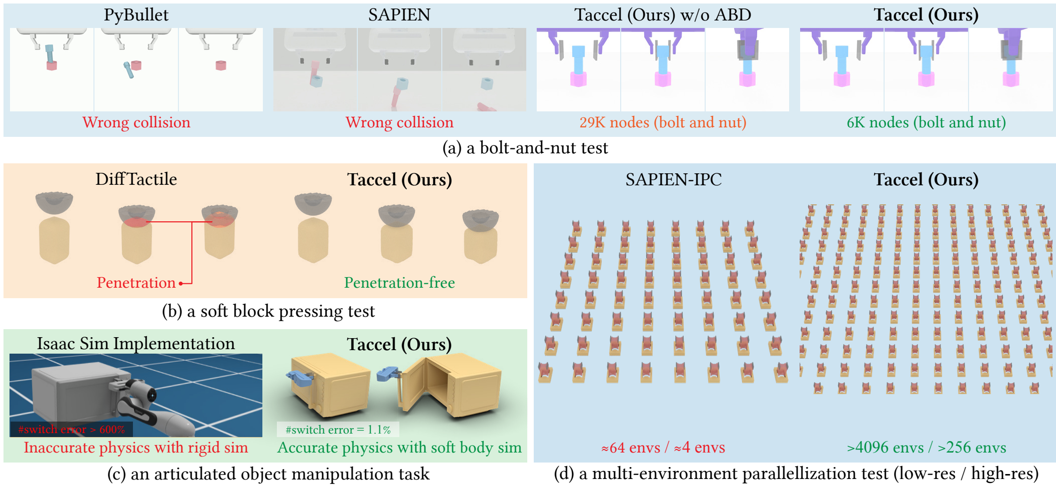


Figure 2: **Comprehensive evaluation of physics simulation capabilities across VBTS simulators.** The comparison spans four challenging scenarios: (a) a bolt-and-nut assembly requiring precise contact handling between non-convex stiff objects, where PyBullet and SAPIEN exhibit collision resolution failures while our full method achieves stable simulation with just 6K nodes compared to 25K nodes without ABD; (b) a soft block pressing test demonstrating contact handling between deformable bodies, where DiffTacile shows penetration artifacts while our method maintains penetration-free interactions; (c) an articulated object manipulation task integrating robot control, tactile sensing, and object interaction, where Isaac Sim's rigid body approach produces physically inaccurate results while our method achieves precise soft-body simulation with physical error around 1%; and (d) a parallel environment test using peg insertion tasks, demonstrating our method's superior scaling capability with over 4096 concurrent environments compared to SAPIEN-IPC's 64 environments, representing a 64-fold improvement in simulation throughput.

To foster community development, we will release **`Taccel`** and maintain active collaboration with researchers to incorporate feedback, add features, and expand capabilities, ensuring its evolution as a comprehensive tool for tactile robotics research.

7 Performance Evaluation of `Taccel`

We evaluate **`Taccel`** through comprehensive benchmarks focusing on simulation precision, tactile signal fidelity, and computational efficiency. Our analysis demonstrates that the combination of ABD and IPC provides significant advantages over existing approaches (Sec. 7.1), generates high-quality tactile signals (Sec. 7.2), and enables efficient scaling (Sec. 7.3).

7.1 Physics Simulation Performances

As illustrated in Fig. 2, **`Taccel`** achieves superior precision and efficiency through its unified ABD and IPC framework, demonstrated across three challenging scenarios.

First, the bolt screwing task (Fig. 2a) demonstrates **`Taccel`**'s ability to handle complex physical interactions between highly non-convex objects, where conventional simulators including PyBullet and SAPIEN [53] fail to handle. This capability stems from our ABD formulation, which provides an efficient approach to simulating dense rigid-soft body interactions. Alternative approaches either model stiff objects as soft bodies, introducing excessive DoFs and computational overhead (Fig. 2a, ours w/o ABD), or rely on RBD [14], which requires expensive nonlinear Continuous Collision Detection (CCD) calculations to maintain intersection-free configurations, significantly degrading performance.

Next, physical realism in **`Taccel`** is demonstrated through the soft block pressing scenario (Fig. 2b), where our collision-free and intersection-free guarantees produce notably more realistic deformations compared to penalty-based approaches. This precision extends to practical robotics applications, as shown in the Tac-Man microwave manipulation task (Fig. 2c). Here, **`Taccel`**'s accurate contact force solving enables faithful reproduction of gel pad-object handle interactions, closely matching real-world execution patterns (detailed analysis in Sec. 8.3).

Finally, the computational efficiency of **`Taccel`** emerges from our optimized implementation of the ABD and IPC algorithms, enabling unprecedented scaling capabilities. In a peg insertion task, **`Taccel`** achieves parallel simulation of over 4096 environments on a single GPU with 80GB VRAM—representing a 64-fold improvement over SAPIEN-IPC [48, 8]. This performance gain opens new possibilities for large-scale robotics simulation and learning; see also comprehensive scaling analysis in Sec. 7.3.

7.2 Tactile Signal Simulation

We conducted systematic experiments to evaluate the fidelity of tactile signals generated by **`Taccel`**. Our experimental setup consisted of a calibrated GelSight-type sensor mounted on a vertical rack for precise movement control (Fig. 3).

The evaluation followed a controlled procedure in both real-world and simulated environments. For real-world data collection, 3D-printed objects were fixed beneath the sensor, which applied consistent pressure while recording RGB camera images. Depth maps were extracted using Yuan *et al.*'s method [59]. We then replicated this pressing sequence in **`Taccel`** using a high-resolution soft body gel pad (maximal cell volume $V_{\max} \approx 10^{-12}\text{m}^3$) to ensure signal fidelity.

We evaluated the simulation accuracy using 18 objects from a standard tactile shape testing dataset [16]. The qualitative and quantitative comparisons shown in Fig. 3 demonstrate **`Taccel`**'s ability to produce highly realistic tactile patterns, achieving an average SSIM of 0.93 across all test objects. Minor variations between simulated and real signals primarily stem from manufacturing tolerances in the 3D-printed objects and challenges in precise camera calibration. Despite these practical limitations, the results establish **`Taccel`**'s capability to generate high-fidelity tactile signals suitable for VBTSs, with simulated patterns closely matching experimental measurements.

7.3 Multi-environment Simulation

Efficient and stable parallel simulation of multiple environments is crucial for scaling up synthetic data collection across diverse downstream tasks. To evaluate **`Taccel`**'s capabilities in this regard, we designed three test cases of increasing complexity. First, we implemented a peg-insertion task adapted from SAPIEN-IPC [8, 48], where two gel pads (139 nodes and 317 cells each) follow a scripted trajectory to squeeze the peg around the hold. Next, we scaled this task to a higher resolution, increasing each gel pad to 1,533 nodes and 5,360 cells. Finally, to demonstrate **`Taccel`**'s potential for advanced tactile sensing research, we created a scripted grasping task using a customized five-fingered dexterous hand equipped with 17 gel pads covering the entire hand, totaling 5,157 nodes and 14,311 cells.

Using a single NVIDIA H100 80G GPU, we benchmarked **`Taccel`** against SAPIEN-IPC. The results in Fig. 4 showcase **`Taccel`**'s superior performance: in the low-resolution peg-insertion task, **`Taccel`** achieves 915 FPS (18.30× faster than wallclock time) while managing over 4096 parallel environments—a 16-fold improvement over the baseline using the same GPU memory, enabled by our efficient parallelization implementation. The high-resolution test further demonstrates **`Taccel`**'s advantages, maintaining both stability and precision while consistently outperforming the baseline. Even in the complex dexterous hand scenario, **`Taccel`** efficiently handles full-hand tactile sensing, achieving 12.67 FPS across 256 environments. We observed that SAPIEN-IPC's use of FP32 precision

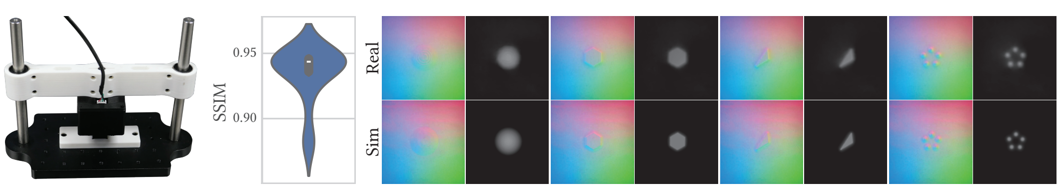


Figure 3: **Quantitative and qualitative evaluation of tactile signal fidelity between real and simulated sensors.** The experimental setup (upper left) shows a GelSight-type sensor mounted on a vertical testing rack for controlled contact measurements. Statistical analysis of simulation accuracy is presented as a violin plot (upper right) showing the distribution of SSIM scores between real and simulated tactile patterns. Representative examples (lower portion) display paired comparisons of real-world (top row) and simulated (bottom row) tactile signals across four different contact scenarios, demonstrating the high-fidelity reproduction of surface features in both RGB patterns and depth maps.

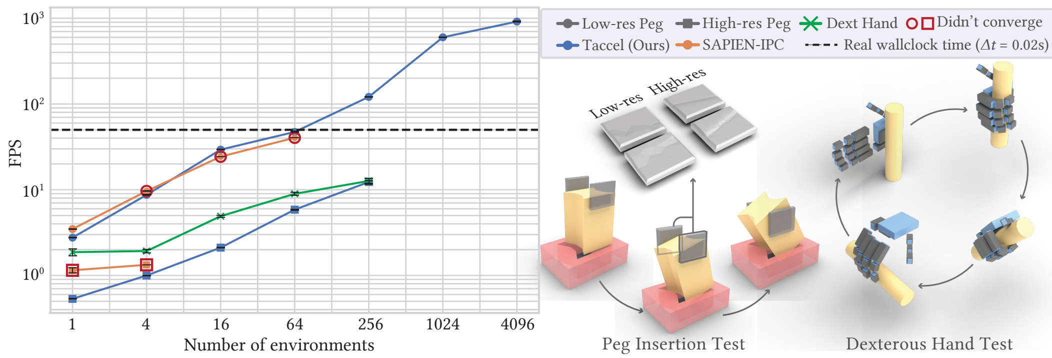


Figure 4: **Parallel simulation performance analysis across environment scaling.** The plot compares frames per second (FPS) achieved by **`Taccel`** (FP64) and SAPIEN-IPC (FP32) on an NVIDIA H100 80G GPU for both peg insertion and dexterous hand manipulation tasks. The dashed line indicates real-world clock time ($\Delta t = 0.02$s), while red circles mark cases where projected Newton steps failed to converge within the optimization time limit. Our method demonstrates superior scaling efficiency, maintaining performance even at 4096 parallel environments.

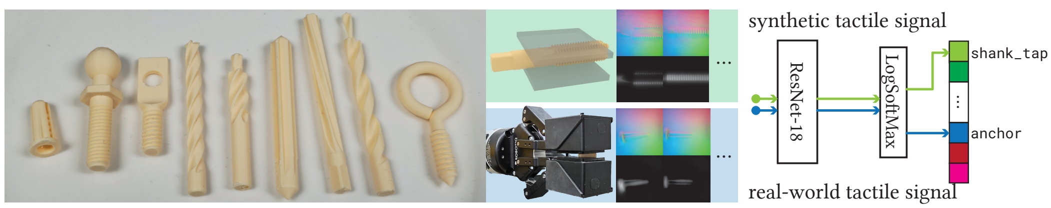


Figure 5: **Object classification pipeline using synthetic tactile data generated from `Taccel`.** Top: Set of 10 mechanical parts (bolts, nuts, and other hardware components) used in our experiments, featuring diverse geometric characteristics. Bottom left: Comparison between simulated and real-world GelSight tactile sensor readings during grasping, demonstrating the fidelity of our synthetic data generation. Each pair shows the RGB image and corresponding depth map captured by the tactile sensor. Bottom right: Classification network architecture.

leads to convergence issues when solving contact forces in Eq. (6), particularly in the logarithmic barrier energy term calculations, as indicated by red outlines in Fig. 4.

8 Applications in Tactile Robotics

We demonstrate **`Taccel`**'s capabilities across three tactile-informed robotic tasks: training object classification models with synthetic data (Sec. 8.1), generating a large-scale dataset through parallel grasp simulations (Sec. 8.2), and manipulating articulated objects (Sec. 8.3). These applications showcase how our framework enables precise robotic simulation and scalable synthetic data generation.

8.1 Learning object classification Models

To demonstrate **`Taccel`**'s ability to generate synthetic training data that generalizes to real-world scenarios, we developed a tactile-based object classification system. As shown in Fig. 5, our approach trains a DNN to classify objects using high-resolution tactile signals obtained from single grasp trials.

Following Yang *et al*. [56], we selected 10 mechanical parts with distinct fine-grained geometries (illustrated in Fig. 5, top). For each object, we simulated 200 grasp trials using a parallel gripper with randomized grasping poses. The depth maps $d_{(u,v)}$ extracted from these simulated tactile signals yielded approximately 4,000 training samples. To enhance dataset robustness, we augmented the depth maps through random affine transformations, morphological operations (erosion and dilation), and Gaussian filtering. We then trained a ResNet-18 model [19] for 10-category object classification using these depth images.

For real-world validation, we collected tactile signals using a RobotiQ-2F85 parallel gripper equipped with GelSight-type sensors. The test objects were 3D printed at 0.2mm layer height to maintain high geometric fidelity. We gathered 16 images per object and applied random affine transformations for augmentation before evaluating the trained network's classification performance.

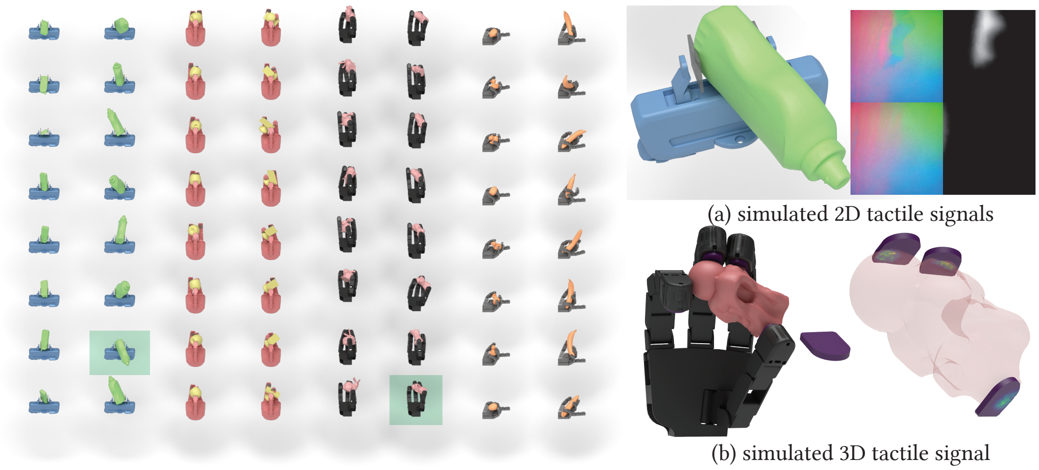


Figure 6: **Examples of the synthesized grasps from `MultiDex-Tac` and the simulated tactile signals in 2D and 3D.** Diverse grasping configurations are synthesized and simulated for different objects. The corresponding tactile feedback (bottom row) represented as both 2D pressure maps (colored visualizations) and 3D deformation patterns of the tactile sensor surface are synthesized.

As shown in Tab. 2, our model achieved 86.50% accuracy on the synthetic test set and 70.94% on real-world samples without any domain adaptation. This modest sim-to-real gap demonstrates **`Taccel`**'s capability to generate precise tactile signals that enable data-efficient training of transferable tactile perception models.

Table 2: **Object classification and sim-to-real performance on the mechanical parts object group.** Performance metrics include number of tactile sensors (N), Degree of Freedom (DoF), total sensing area (S.A.), average sensor-object contact area (C.A.) with percentage of total sensing area in parentheses, and classification accuracy (Acc.). Results compare synthetic and real-world data for mechanical parts classification.

Data	**DoF**	N	**S.A.** / cm^2	**C.A.** / cm^2	**Acc.** ↑
Mech.	2	2	32.0	4.77 (14.92%)	86.50%
Mech. (Real)	2	2	32.0	5.69 (17.78%)	70.94%

Table 3: **Object classification performance across different robotic hand configurations.** The abbreviations follows Tab. 2. Results compare performance across different robotic hand designs, including parallel gripper, Robotiq-3F, Allegro Hand, and the F-TAC hand.

Data	**DoF**	N	**S.A.** / cm^2	**C.A.** / cm^2	**Acc.** ↑
Gripper	2	2	32.0	5.05 (15.78%)	44.56%
Robotiq-3F	8	3	27.0	4.98 (18.43%)	44.61%
Allegro	16	4	23.0	**7.67 (33.34%)**	**54.30%**
F-TAC [62]	15	17	59.7	4.00 (6.700%)	42.54%

8.2 Robotic Grasping with Tactile Sensors

We investigate robotic grasping across different hand configurations with varying tactile sensor arrangements. By extending the DFC algorithm [37], we first generate contact-oriented grasping poses for four robotic hands, then simulate their tactile responses within **`Taccel`**, as illustrated in Fig. 6. Our key modification to the DFC algorithm promotes perpendicular contact between gel pads and object surfaces, optimizing for downstream tasks that rely on tactile perception.

Following Liu *et al.* [37], we synthesize grasping poses in the robot's joint space $q \in \mathcal{Q}$ relative to the object frame by minimizing a modified Gibbs energy:

$$E(O, q, T) = E_{\mathrm{DFC}} + \lambda_{\mathrm{contact}} E_{\mathrm{contact}}(O, q, T). \tag{20}$$

The force-closure term E_{DFC} maintains its original formulation [37], with added constraints to ensure gel pad penetration depth ϵ (typically $0.5\,\mathrm{mm}$). We introduce a new contact term E_{contact}

that aligns the normals of gel pad contacts $c_i \in \partial^+ \mathcal{G}_i$ with their corresponding object surface normals $o_i = \arg\min_{o \in \partial O} \|o - c_i\|$:

$$E_{\text{contact}}(O, q, T) = 1 - \left\langle c_i^{\perp}, o_i^{\perp} \right\rangle, \tag{21}$$

where $(\cdot)^{\perp}$ represents the surface normals of the gel pad and object surfaces.

We evaluated this approach on 10 diverse objects from ContactDB [4], YCB [7], and adversarial object [40] datasets. For each object, we generated grasps using 4 different robotic hands, producing approximately 14k total grasps. These were simulated in **`Taccel`** to generate tactile signals, with key metrics summarized in Tab. 2, including sensor count, robot DoF, sensing areas, and average contact ratios.

We release this synthetic data as **`MultiDex-Tac`**, extending the MultiDex dataset [31] with tactile perception capabilities. This dataset can be further expanded with additional robotic hands and objects, serving as a foundation for various robotic tasks. To demonstrate its utility, we implemented the object classification pipeline described in Sec. 8.1, adapting it to use the deformed coat's point cloud as input and PointNet as the feature extractor. The classification performance across different hand configurations is reported in Tab. 2.

Our analysis reveals an interesting trade-off: while robots differ in sensor count and sensing area, higher dexterity (as in the Allegro hand with more DoFs along different axes) enables better object contact despite fewer sensors. This enhanced contact leads to superior classification accuracy, highlighting the balance between sensor count and dexterity in tactile hand design. These findings demonstrate **`Taccel`**'s value in validating robotic hand designs before physical fabrication.

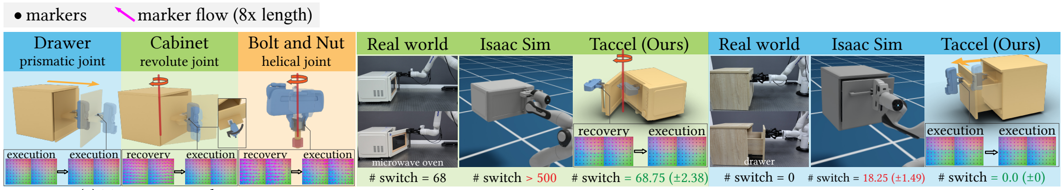
8.3 Articulated Object Manipulation

We consider articulated object manipulation, a challenging task where tactile perception provides critical feedback on hand-object contact, informing object articulation and guiding robot actions [25, 63]. In the Tac-Man framework [63], the system alternates between *execution* and *recovery* phases. During execution, the system performs coarse manipulation actions (*e.g.*, pulling backward) to gradually move the articulated object part. When the actual motion deviates from intended trajectories due to articulation constraints, the gel pad deforms, creating contact deviation reflected in marker flow magnitudes. Once these flows exceed threshold δ_0, the system enters recovery mode to restore stable contact by reducing deviation, before resuming execution. This execution-recovery cycle typically requires tens of iterations for finishing manipulation.

Tac-Man's effectiveness relies heavily on gel pad deformation for motion adaptation and tactile feedback. While the original implementation by Zhao *et al.* used rigid body simulation with gripper compliance approximations, it couldn't authentically replicate gel pad deformation, contact dynamics, and tactile feedback, resulting in significant sim-to-real gaps during large-scale verification.

`Taccel` overcomes these limitations through accurate simulation of sensor-object contact and gel pad deformation. We validated our approach on three types of articulated objects: drawers with prismatic joints, cabinets with revolute joints, and bolt-nut pairs with helical joints. For drawers and cabinets, we randomly sampled joint positions. The manipulation sequences and corresponding tactile signals are presented in Fig. 7 (top row).

To evaluate sim-to-real correspondence, we compared our simulation against real-world Tac-Man execution using a microwave (revolute joint) and drawer (prismatic joint), shown in Fig. 7's middle and bottom rows. We manually created URDF models to match real-world geometries and kinematics, maintaining identical initial grasping poses across simulation and physical setups. Following Tac-Man's implementation, we set $\delta_0 = 0.4\,\text{mm}$ and $\alpha = 0.6$. For comprehensive comparison, we also tested Tac-Man's official implementation on these objects. As demonstrated in Fig. 7, **`Taccel`** faithfully reproduces real-world execution patterns, with execution-recovery switch counts averaging 68.75 and 0.0 for revolute and prismatic settings respectively—remarkably matching real-world observations. While Isaac Sim successfully simulated the manipulation process, its dynamics gap resulted in substantially higher switch counts. These results highlight **`Taccel`**'s capability to authentically replicate physical interactions in precise manipulation scenarios.



(a) Tac-Man simulations (b) comparisons among real-world execution, Isaac Sim simulation, and Taccel simulation

Figure 7: **Tac-Man manipulation simulation.** (a) Demonstration of Tac-Man's execution-recovery cycles on three articulated objects: a drawer (prismatic joint), cabinet (revolute joint), and bolt-nut assembly (helical joint). Black dots indicate marker positions, with purple lines showing marker flow (8x magnified). (b) Three-way comparison between real-world execution (left), Isaac Sim implementation [63] (middle), and our **`Taccel`** simulation (right). The execution-recovery switch counts (#switch) demonstrate **`Taccel`**'s accuracy in replicating real-world behavior, with values closely matching physical experiments compared to Isaac Sim's higher counts. Tactile feedback visualizations (bottom right) show marker positions and flows during execution and recovery phases.

9 Summary

We present **`Taccel`**, a flexible and high-performance physics simulator designed for diverse robots with VBTS integration. Through user-friendly APIs, it enables precise and efficient simulation of complex tactile robotic tasks while generating realistic tactile signals. The simulator excels in capturing intricate deformation and contact dynamics of soft gel pads with unprecedented stability, while supporting parallel environments at over 900 FPS with thousands of concurrent simulations. These capabilities position **`Taccel`** as a powerful tool for validating hand designs and sensor configurations before fabrication, potentially reducing development time and costs.

Limitations The computational demands of large-scale simulation remain challenging for **`Taccel`**. One main bottleneck is the PCG-based linear system solving in Newton iterations, which we will further optimize. Additional optimization strategies include carefully relaxing convergence tolerance of solvers or simplifing simulation protocols to enhance efficiency. Faster speed can also be achieved with larger timesteps (enabled by IPC's unconditionally stable solver) and relaxed solver tolerances. Additionally, while **`Taccel`** excels at physics simulation, it currently lacks real-time rendering capabilities and a graphical user interface. Developing **`Taccel`** into an IsaacGym-like comprehensive simulation platform would require substantial additional engineering effort.

Acknowledgement

This work is supported in part by the National Science and Technology Major Project (2022ZD0114900), the National Natural Science Foundation of China (62376031), the Beijing Nova Program, the State Key Lab of General AI at Peking University, the PKU-BingJi Joint Laboratory for Artificial Intelligence, and the National Comprehensive Experimental Base for Governance of Intelligent Society, Wuhan East Lake High-Tech Development Zone. We thank Lei Yan (LeapZenith AI Research) for his support in mechanical engineering, Zhenghao Qi (PKU, THU), Sirui Xie (PKU) for their assistance in setting up experiments, and Ms. Hailu Yang (PKU) for her help in procuring raw materials.

References

[1] Iretiayo Akinola, Jie Xu, Jan Carius, Dieter Fox, and Yashraj Narang. Tacsl: A library for visuotactile sensor simulation and learning. *arXiv preprint arXiv:2408.06506*, 2024. 1, 3

[2] Maria Bauza, Antonia Bronars, and Alberto Rodriguez. Tac2pose: Tactile object pose estimation from the first touch. *International Journal of Robotics Research (IJRR)*, 42(13):1185–1209, 2023. 4

[3] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446):eaat8414, 2019. 1, 3, 4

[4] Samarth Brahmbhatt, Cusuh Ham, Charles C Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 12

[5] Antonia Bronars, Sangwoon Kim, Parag Patre, and Alberto Rodriguez. Texterity: Tactile extrinsic dexterity. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 4

[6] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3300–3307, 2018. 4

[7] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 12

[8] Weihang Chen, Jing Xu, Fanbo Xiang, Xiaodi Yuan, Hao Su, and Rui Chen. General-purpose sim2real protocol for learning contact-rich manipulation with marker-based visuotactile sensors. *IEEE Transactions on Robotics (T-RO)*, 40:1509–1526, 2024. 1, 3, 4, 9

[9] Yunuo Chen, Minchen Li, Lei Lan, Hao Su, Yin Yang, and Chenfanfu Jiang. A unified newton barrier method for multibody dynamics. *ACM Transactions on Graphics (TOG)*, 41(4):1–14, 2022. 5

[10] Zixi Chen, Shixin Zhang, Shan Luo, Fuchun Sun, and Bin Fang. Tacchi: A pluggable and low computational cost elastomer deformation simulator for optical tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 8(3):1239–1246, 2023. 3

[11] Siyuan Dong, Wenzhen Yuan, and Edward H Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 4

[12] Wenxin Du, Wenqiang Xu, Jieji Ren, Zhenjun Yu, and Cewu Lu. Tacipc: Intersection-and inversion-free fem-based elastomer simulation for optical tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 2024. 1, 3

[13] Mark Edmonds, Feng Gao, Hangxin Liu, Xu Xie, Siyuan Qi, Brandon Rothrock, Yixin Zhu, Ying Nian Wu, Hongjing Lu, and Song-Chun Zhu. A tale of two explanations: Enhancing human trust by explaining robot behavior. *Science Robotics*, 4(37):eaay4663, 2019. 1

[14] Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil-Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M Kaufman, and Daniele Panozzo. Intersection-free rigid body dynamics. *ACM Transactions on Graphics (TOG)*, 40(4), 2021. 8

[15] Carlo Ferrari, John F Canny, et al. Planning optimal grasps. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1992. 4

[16] Daniel Fernandes Gomes, Paolo Paoletti, and Shan Luo. Generation of gelsight tactile images for sim2real learning. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):4177–4184, 2021. 9

[17] Daniel Fernandes Gomes, Achu Wilson, and Shan Luo. Gelsight simulation for sim2real learning. In *ICRA ViTac Workshop*, 2019. 3

[18] Xiaofeng Guo, Hung-Jui Huang, and Wenzhen Yuan. Estimating properties of solid particles inside container using touch sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023. 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 10

[20] Francois R Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 1, 4

[21] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 3

[22] Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B Tenenbaum, William T Freeman, Jiajun Wu, Daniela Rus, and Wojciech Matusik. Chainqueen: A real-time differentiable physical simulator for soft robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 3

[23] Hung-Jui Huang, Xiaofeng Guo, and Wenzhen Yuan. Understanding dynamic tactile sensing for liquid property estimation. *Robotics: Science and Systems (RSS)*, 2022. 4

[24] Bassam Jalgha, Daniel Asmar, Elie Shammas, and Imad Elhajj. Hierarchical fall avoidance strategy for small-scale humanoid robots. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012. 1

[25] Carlos A Jara, Jorge Pomares, Francisco A Candelas, and Fernando Torres. Control framework for dexterous manipulation using dynamic visual servoing and tactile sensors' feedback. *Sensors*, 14(1):1787–1804, 2014. 1, 12

[26] Chunpeng Jiang, Wenqiang Xu, Yutong Li, Zhenjun Yu, Longchun Wang, Xiaotong Hu, Zhengyi Xie, Qingkun Liu, Bin Yang, Xiaolin Wang, et al. Capturing forceful interaction with deformable objects using a deep learning-powered stretchable tactile array. *Nature Communications*, 15(1):9513, 2024. 3

[27] Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345–359, 2009. 3

[28] Sangwoon Kim, Antonia Bronars, Parag Patre, and Alberto Rodriguez. Texterity–tactile extrinsic dexterity: Simultaneous tactile estimation and control for extrinsic dexterity. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 1

[29] Lei Lan, Danny M Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. Affine body dynamics: fast, stable and intersection-free simulation of stiff materials. *ACM Transactions on Graphics (TOG)*, 41(4):1–14, 2022. 3, 4, 5

[30] Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy R Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M Kaufman. Incremental potential contact: intersection- and inversion-free, large-deformation dynamics. *ACM Transactions on Graphics (TOG)*, 39(4):49, 2020. 3, 4, 5

[31] Puhao Li, Tengyu Liu, Yuyang Li, Yixin Zhu, Yaodong Yang, and Siyuan Huang. Gendexgrasp: Generalizable dexterous grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 12

[32] Rui Li, Robert Platt, Wenzhen Yuan, Andreas Ten Pas, Nathan Roscup, Mandayam A Srinivasan, and Edward Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014. 4

[33] Wanlin Li, Zihang Zhao, Leiyao Cui, Weiyi Zhang, Hangxin Liu, Li-An Li, and Yixin Zhu. Minitac: An ultra-compact 8 mm vision-based tactile sensor for enhanced palpation in robot-assisted minimally invasive surgery. *IEEE Robotics and Automation Letters (RA-L)*, 9(12):11170–11177, 2024. 1, 3

[34] Yuyang Li, Bo Liu, Yiran Geng, Puhao Li, Yaodong Yang, Yixin Zhu, Tengyu Liu, and Siyuan Huang. Grasp multiple objects with one hand. *IEEE Robotics and Automation Letters (RA-L)*, 9(5):4027–4034, 2024. 4

[35] Changyi Lin, Han Zhang, Jikai Xu, Lei Wu, and Huazhe Xu. 9dtact: A compact vision-based tactile sensor for accurate 3d shape reconstruction and generalizable 6d force estimation. *IEEE Robotics and Automation Letters (RA-L)*, 2023. 1, 3

[36] Hongbin Liu, Juan Greco, Xiaojing Song, Joao Bimbo, Lakmal Seneviratne, and Kaspar Althoefer. Tactile image based contact shape recognition using neural network. In *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2012. 1

[37] Tengyu Liu, Zeyu Liu, Ziyuan Jiao, Yixin Zhu, and Song-Chun Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *IEEE Robotics and Automation Letters (RA-L)*, 7(1):470–477, 2021. 4, 11

[38] John Lloyd and Nathan F Lepora. Pose-and-shear-based tactile servoing. *International Journal of Robotics Research (IJRR)*, 43(7):1024–1055, 2024. 2, 3

[39] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/nvidia/warp, March 2022. NVIDIA GPU Technology Conference (GTC). 3, 7

[40] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *Robotics: Science and Systems (RSS)*, 2017. 12

[41] Juntian Qu, Baijin Mao, Zhenkun Li, Yining Xu, Kunyu Zhou, Xiangyu Cao, Qigao Fan, Minyi Xu, Bin Liang, Houde Liu, et al. Recent progress in advanced tactile sensing technologies for soft grippers. *Advanced Functional Materials*, 33(41):2306249, 2023. 3

[42] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *International Journal of Robotics Research (IJRR)*, 40(12-14):1385–1401, 2021. 3

[43] Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):2361–2368, 2022. 2, 3

[44] Zilin Si, Gu Zhang, Qingwei Ben, Branden Romero, Zhou Xian, Chao Liu, and Chuang Gan. Difftactile: A physics-based differentiable tactile simulator for contact-rich robotic manipulation. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 1, 2, 3, 7

[45] Bruno Siciliano and Oussama Khatib. Robotics and the handbook. In *Springer Handbook of Robotics*, pages 1–6. Springer, 2016. 4

[46] Deborah Sulsky, Zhen Chen, and Howard L Schreyer. A particle method for history-dependent materials. *Computer methods in applied mechanics and engineering*, 118(1-2):179–196, 1994. 3

[47] Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, et al. Neuralfeels with neural fields: Visuotactile perception for in-hand manipulation. *Science Robotics*, 9(96):eadl0628, 2024. 4

[48] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024. 1, 2, 9

[49] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):3930–3937, 2022. 1, 3, 7

[50] Zhi Wang, Yuchen Mo, Shengmiao Jin, and Wenzhen Yuan. Doorbot: Closed-loop task planning and manipulation for door opening in the wild with haptic feedback. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. 4

[51] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft Robotics*, 5(2):216–227, 2018. 3

[52] Goran Westling and Roland S Johansson. Factors influencing the force control during precision grip. *Experimental Brain Research*, 53:277–284, 1984. 1, 3

[53] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

[54] Jie Xu, Sangwoon Kim, Tao Chen, Alberto Rodriguez Garcia, Pulkit Agrawal, Wojciech Matusik, and Shinjiro Sueda. Efficient tactile simulation with differentiability for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2023. 1, 3

[55] Wenqiang Xu, Zhenjun Yu, Han Xue, Ruolin Ye, Siqiong Yao, and Cewu Lu. Visual-tactile sensing for in-hand object reconstruction. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 4

[56] Sanghoon Yang, Won Dong Kim, Hyunkyu Park, Seojung Min, Hyonyoung Han, and Jung Kim. In-hand object classification and pose estimation with sim-to-real tactile transfer for robotic manipulation. *IEEE Robotics and Automation Letters (RA-L)*, 9(1):659–666, 2023. 4, 10

[57] Kunpeng Yao and Aude Billard. Exploiting kinematic redundancy for robotic grasping of multiple objects. *IEEE Transactions on Robotics (T-RO)*, 39(3):1982–2002, 2023. 4

[58] Zhenjun Yu, Wenqiang Xu, Pengfei Xie, Yutong Li, and Cewu Lu. Dynamic reconstruction of hand-object interaction with distributed force-aware contact representation. *arXiv preprint arXiv:2411.09572*, 2024. 4

[59] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. 1, 3, 7, 9

[60] Wenzhen Yuan, Rui Li, Mandayam A Srinivasan, and Edward H Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 304–311. IEEE, 2015. 4

[61] Kappassov Zhanat, Juan-Antonio Corrales, and Perdereau Veronique. Tactile sensing in dexterous robot hands-review. *Robotics & Autonomous Systems*, 74:195–220, 2015. 1

[62] Zihang Zhao, Wanlin Li, Yuyang Li, Tengyu Liu, Boren Li, Meng Wang, Kai Du, Hangxin Liu, Yixin Zhu, Qining Wang, et al. Embedding high-resolution touch across robotic hands enables adaptive human-like grasping. *arXiv preprint arXiv:2412.14482*, 2024. 1, 11

[63] Zihang Zhao, Yuyang Li, Wanlin Li, Zhenghao Qi, Lecheng Ruan, Yixin Zhu, and Kaspar Althoefer. Tac-Man: Tactile-informed prior-free manipulation of articulated objects. *IEEE Transactions on Robotics (T-RO)*, 41:538–557, 2024. 1, 3, 4, 7, 12, 13

[64] Yixin Zhu, Tao Gao, Lifeng Fan, Siyuan Huang, Mark Edmonds, Hangxin Liu, Feng Gao, Chi Zhang, Siyuan Qi, Ying Nian Wu, et al. Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense. *Engineering*, 6(3):310–345, 2020. 4