

ETac: A Lightweight and Efficient Tactile Simulation Framework for Learning Dexterous Manipulation

Zhe Xu¹, Feiyu Zhao¹, Xiyang Huang¹, and Chenxi Xiao^{*1}

Abstract—Tactile sensors are increasingly integrated into dexterous robotic manipulators to enhance contact perception. However, learning manipulation policies that rely on tactile sensing remains challenging, primarily due to the trade-off between fidelity and computational cost of soft-body simulations. To address this, we present ETac, a tactile simulation framework that models elastomeric soft-body interactions with both high fidelity and efficiency. ETac employs a lightweight data-driven deformation propagation model to capture soft-body contact dynamics, achieving high simulation quality and boosting efficiency that enables large-scale policy training. When serving as the simulation backend, ETac produces surface deformation estimates comparable to FEM and demonstrates applicability for modeling real tactile sensors. Then, we showcase its capability in training a blind grasping policy that leverages large-area tactile feedback to manipulate diverse objects. Running on a single RTX 4090 GPU, ETac supports reinforcement learning across 4,096 parallel environments, achieving a total throughput of 869 FPS. The resulting policy reaches an average success rate of 84.45% across four object types, underscoring ETac’s potential to make tactile-based skill learning both efficient and scalable.

I. INTRODUCTION

Robots rely on tactile sensing to perceive and interpret physical interactions in contact-rich environments. Following this trend, tactile sensors have been widely integrated into robotic hardware, including grippers [1], [2], dexterous hands [3], [4], arms [5], [6], and feet [7]. Alongside advances in hardware, robots have demonstrated improved dexterous manipulation skills, such as grasping [1], [3] and object reorientation [8]. Collectively, these developments not only enhance the fidelity and safety of physical interactions, but also broaden the range of real-world applications where robots can be effectively deployed [9], [10].

Building on these advances, there is growing interest in developing tactile-based skills for robotic manipulation. A widely adopted approach is reinforcement learning (RL), which has shown strong potential for acquiring diverse manipulation skills. Since RL requires extensive interaction data collected through repeated task executions, simulation environments are often preferred for their low cost, high repeatability, and minimal risk to hardware [11], [12]. However, learning tactile-based skills through RL remains challenging, primarily due to the lack of tactile simulators that are both high-fidelity and computationally efficient.

This work was supported by the Natural Science Foundation of Shanghai under Grant 25ZR1402370.

The authors are with the School of Information Science and Technology at ShanghaiTech University, Shanghai 201210, China. {xuzhe2023, zhaofy12024, huangxy22023, xiaochx}@shanghaitech.edu.cn, *Corresponding author: Chenxi Xiao.

Project webpage: <https://lassford.github.io/ETac/>.

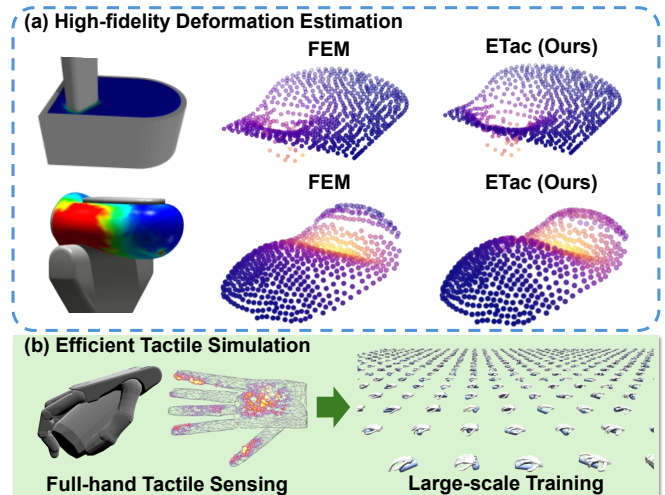


Fig. 1: **ETac: a lightweight and efficient tactile simulation framework.** (a) ETac estimates elastomer surface deformations with fidelity comparable to that of FEM. (b) It enables large-area tactile sensing for dexterous manipulators while supporting large-scale RL training.

Currently, a common choice for state-of-the-art tactile simulators is based on soft-body physics engines. For instance, simulators built on the Finite Element Method (FEM) can capture high-quality contact deformations during physical interactions [13], [14]. However, many of these simulators are developed as standalone tools, their integration with multi-fingered robotic hands remains limited and largely inaccessible to the broader research community. On the other hand, the intensive data exchange among numerous mesh elements makes FEM-based models several orders of magnitude slower than rigid-body physics engines [15]. Such computational costs not only slows RL training but also prevents large-scale parallel rollouts, which are essential for stable gradient estimation for policy updates [16]. To mitigate these issues, some studies have explored accelerating simulation with simplified models [8], [17]–[19]. Nevertheless, these approaches typically tend to emphasize local surface deformations at contact areas. They fail to sufficiently capture strain propagation, which is crucial for accurately estimating global elastomer deformation.

In this work, we introduce **ETac**, a tactile simulation framework that models the soft-body behavior of tactile sensors with both high computational efficiency and simulation fidelity. Compared to previous tactile simulation approaches [8], [18], [19], the key innovation of ETac is its learnable deformation propagation model, which integrates an analytical model with a compact residual network. This

hybrid approach enables ETac to capture the nonlinear dynamics arising from anisotropy and surface curvature, producing high-quality simulations on curved surfaces, such as those of dexterous robotic hands. Furthermore, ETac features a lightweight design that supports large-scale parallelization, achieving performance comparable to or exceeding that of existing tactile simulators.

Our experiments demonstrate that ETac not only achieves high-fidelity sensor simulation but also provides new possibilities for robot learning. First, when used as the simulation backend of tactile sensors, it faithfully reproduces surface deformation patterns upon contact. These deformation patterns can be further mapped to real sensors’ output signals, enabling the real-to-sim modeling for tactile sensor hardware. Second, ETac’s efficiency overcomes the scalability bottleneck in RL training for contact-rich tasks. In a challenging dexterous blind grasping task within Isaac Gym [20], ETac accomplished RL training across 4,096 parallel environments at up to a total of 869 FPS on a single RTX 4090 GPU. This represents an 11× increase in FPS and 128× parallelization capacity compared to an FEM solver under the same GPU hardware. Furthermore, the policy trained with ETac’s tactile feedback achieved an 84.45% success rate (a 21.48% improvement over a non-tactile baseline). These results highlight the potential of ETac as a scalable and versatile platform for tactile-driven skill learning in dexterous robots, contributing to the embodied AI community.

In summary, the contributions of this paper are as follows:

- **ETac:** A lightweight and efficient tactile simulation framework for learning dexterous manipulation.
- A learnable deformation propagation model that estimates the elastic response of deformable tactile sensor surfaces.
- Comprehensive evaluations of ETac in terms of simulation fidelity, computational efficiency, and effectiveness in training RL policies.

II. RELATED WORK

A. Tactile Sensing and Manipulation

Tactile sensing has become a fundamental component of robotic hands. Over the past decades, tactile sensors have been integrated into various manipulators such as the Svelte Hand [21], Eyesight Hand [22], and F-Tac [4], progressively achieving higher resolution and broader coverage. With these advances, researchers have been developing diverse tactile-based skills, including in-hand manipulation [23], [24], grasping [1], and object placement [25], to mention a few.

Despite this progress, a critical gap is the lack of effective approaches to fully exploit the information that modern tactile sensors conveyed. Traditional methods often rely on simplified tactile representations, encoding feedback as binary contact states or low-dimensional force/torque signals [3], [8]. These representations fail to capture the rich, high-fidelity information provided by modern tactile sensors. Reinforcement learning has emerged as a promising alternative, offering a data-driven framework that can adapt to diverse

TABLE I: **Comparison of lightweight tactile simulation methods.** Element Dimension refers to whether displacement is a scalar (1 dim) or vector (3 dims), while Propagation and Type indicate whether and how nodes influence each other.

Method	Element Dimension	Propagation	Propagation Types	Differentiability
Tacto [17]	1	-	-	-
Yin et al. [8]	1	-	-	-
TacSL [19]	3	-	-	✓
Taxim [18]	3	✓	Linear	-
ETac (Ours)	3	✓	Linear & Nonlinear	✓

sensor modalities and leverage high-dimensional tactile feedback for skill acquisition [11], [12], [15]. Since data collection from real-world interactions is prohibitively expensive, most reinforcement learning approaches still depend heavily on simulation. Nevertheless, accurately reproducing tactile feedback in simulation remains a challenge [14], [15].

B. Simulating Tactile Sensors

Simulators of tactile sensors aim to obtain simulated response signals in contact events. Developing such simulators is technically challenging, mainly due to the high degrees of freedom in soft-body models, and the complex nonlinear contact dynamics. High-fidelity methods, including finite element models (FEM) [13], [14], [26] and the Material Point Method (MPM) [27], [28], have demonstrated realistic simulation results. However, the considerable computational cost limits their scalability in large-scale RL training.

To address efficiency bottlenecks, several studies adopt approximation strategies, as summarized in Tab. I. A common approach is to approximate deformation via penetration distance [8], [17], [19], which offers efficiency but fails to simulate strain propagation to surrounding regions, thereby compromising realism. Taxim [18] introduces an affine linear deformation propagation model that partially alleviates this issue, yet it overlooks higher-dimensional effects such as anisotropy and surface curvature. To overcome these limitations, ETac incorporates a lightweight neural network-based propagation model that simulates the non-linear influences between elements on deformation, thereby achieving high fidelity while preserving the computational efficiency necessary for scalable RL training.

III. METHODOLOGY

Our research aims to develop tactile simulators that achieve both high simulation accuracy and computational efficiency. To this end, we propose ETac, a lightweight tactile simulation framework, with its response learned to be consistent with high-fidelity, computationally expensive FEM simulations (Sec. III-A). Furthermore, to substantiate ETac’s effectiveness in supporting scalable reinforcement learning for contact-rich manipulation, we establish a tactile-based blind grasping benchmark addressed through RL (Sec. III-B). The system pipeline is illustrated in Fig. 2.

A. Lightweight Tactile Simulation Framework

Overview. To efficiently simulate the physical response of tactile sensors with high fidelity, **ETac** employs a particle-based simulation technique that avoids costly global information updates on dense grids (as required in FEM). The

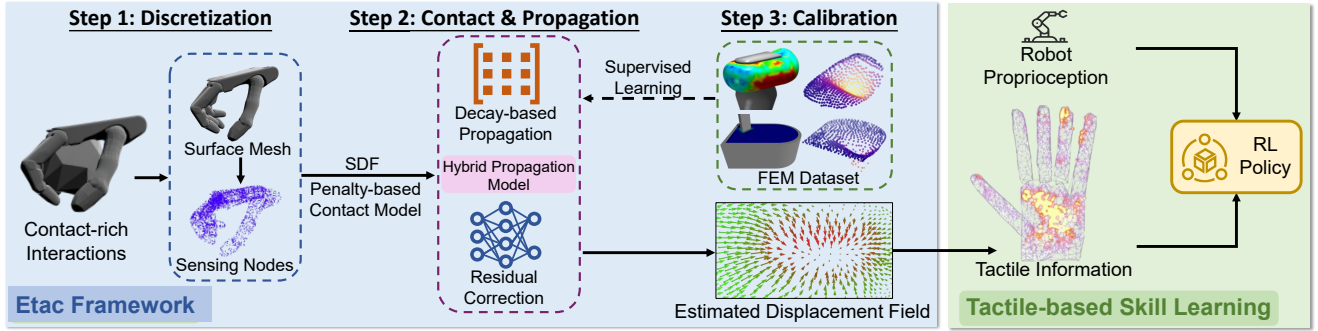


Fig. 2: **Overview of the ETac pipeline.** The manipulator’s surface mesh is discretized into nodes, whose contact dynamics are learned by a hybrid deformation propagation model. We calibrate the propagation model using data from FEM simulation to ensure fidelity. The computed displacement field of the nodes serves as the tactile input to the RL policy during skill learning.

method first discretizes the manipulator’s surface mesh into a set of nodes in 3D space. These nodes detect contact with external objects, and then determine elastomer deformation by propagating contact-induced displacements through node dynamics.

To implement this, contact events are first detected using signed distance fields (SDFs) [29] between nodes and object meshes, including both external object contact and self-contact of the hand. Here, we refer to nodes in direct contact with objects as *active nodes*. Active node’s contact dynamics have been extensively studied in previous works. Here we follow [15], [19] by adapting the penalty-based contact model. However, such model only models local displacements at the contact location. Due to the elasticity of the material, displacements at active nodes should also propagate through the medium of the elastomer and influence the neighboring region. This propagation effect is critical for simulating soft sensor behavior realistically. Yet, recent lightweight tactile simulators either neglect it entirely [8], [17], [19] or use overly simplified linear models [18], which fail to capture such propagation dynamics with high fidelity.

Modeling Contact Propagation Dynamics. To address the aforementioned limitations, we propose a deformation propagation model. This propagation model computes the displacements of nodes that are not in direct contact with external objects (denoted as *passive nodes*) based on the displacements of active nodes. Here, we denote active node’s displacements as: $U_a = [\hat{u}_1, \hat{u}_2, \dots, \hat{u}_m]$.

Then, the displacements of passive nodes are inferred jointly using two components: (1) a decay-based linear displacement propagation, and (2) a residual correction that incorporates nonlinear effects. The decay-based model provides an empirical prior for propagation, yielding smooth, translation-invariant predictions based on relative indentation locations. The residual component then captures high-dimensional effects that are difficult to model explicitly and provides additive corrections via a network. Together, these two components complement each other by balancing the generalization of physical priors and the expressiveness of the data-driven network, and thus improving simulation accuracy (as evaluated in Sec. IV-A).

1) *Decay-based Linear Propagation.* Surface displacements propagated to passive nodes decay with distances from the indented location. Following prior work [30], we use an exponential kernel to model this spatial decay. Specifically, for an active node p_j , its propagated displacement to a passive node p_i is computed as:

$$\mathbf{u}_i^j = e^{-\alpha \|\mathbf{p}_j - \mathbf{p}_i\|_2} \cdot \hat{\mathbf{u}}_j \quad (1)$$

where \mathbf{u}_i^j denotes the propagated displacement, \mathbf{p}_j and \mathbf{p}_i are the undeformed positions of p_j and p_i , respectively, and α is a learnable parameter that reflects the material-specific attenuation of deformation. To enable efficient paralleled propagation calculation, the overall effect between m active nodes and $(n-m)$ passive nodes is modeled using a propagation matrix $\Omega \in \mathbb{R}^{(n-m) \times m}$, where $\Omega[i, j] = e^{-\alpha \|\mathbf{p}_j - \mathbf{p}_i\|_2}$. The propagated displacements for all passive nodes are then computed as ΩU_a .

Note this spatial decay-based model alone provides only an approximation of the propagation process. It cannot fully capture non-linear effects caused by factors such as surface curvature and anisotropy. This is not only because linear model’s limited representability, but also because there is a subtle error: U_a are not solely caused by direct contact; they already include mutual interactions among active nodes. Hence, directly applying the decay-based propagation “double-counts” these influences. Unlike Taxim [18], which adopts a computationally expensive matrix inversion to explicitly correct the inter-node interactions, our approach learns to correct for this and other residual errors jointly using a neural network in the next stage.

2) *Residual Displacement Correction.* We address the above issues and other residual effects by introducing a residual network Φ , as illustrated in Fig. 3. It uses a lightweight PointNet [31] encoder to extract features from elastomer’s surface, followed by an MLP-based decoder that estimates node-wise residual displacements.

Specifically, the input comprises the undeformed positions $\mathbf{P} \in \mathbb{R}^{n \times 3}$ and normals $\mathbf{N} \in \mathbb{R}^{n \times 3}$ of all nodes, together with the displacements $\hat{\mathbf{U}} \in \mathbb{R}^{n \times 3}$ (the displacements of passive nodes are padded with zeros), forming an $n \times 9$ dimensional feature tensor. The encoder extracts point-wise

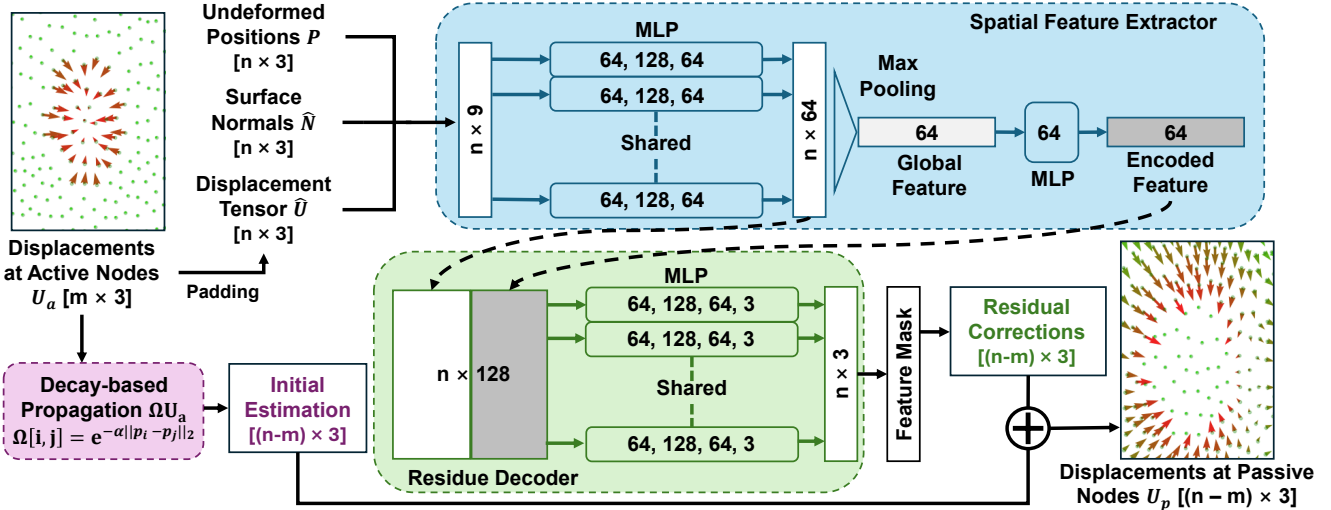


Fig. 3: **Computational process of the propagation model.** Displacements of passive nodes (nodes not in direct contact) are estimated using a decay-based model combined with a residual correction network.

features using shared MLPs and aggregates them via Max-Pooling into a 64-dimensional global feature vector. This global feature is further encoded by another MLP, then broadcast and concatenated with per-node features to form an $n \times 128$ tensor. Finally, the shared MLP decoder produces the node-wise $n \times 3$ residual displacements for all nodes, following which a binary mask $M_p \in \{0, 1\}^n$ is used to retrieve residual displacements of passive nodes, denoted as $\Phi_p(\mathbf{P}, \hat{\mathbf{U}}, \mathbf{N}) \in \mathbb{R}^{(n-m) \times 3}$. Together with the decay-based propagation, the final estimated displacements of all passive nodes \mathbf{U}_p are calculated as:

$$\mathbf{U}_p = \Omega \mathbf{U}_a + \Phi_p(\mathbf{P}, \hat{\mathbf{U}}, \mathbf{N}) \quad (2)$$

Learn Propagation Dynamics From FEM. The kernel parameter α in Eq. (1) and the network weights jointly determine the characteristics of deformation propagation in Eq. (2). As these parameters implicitly encode the material properties, we calibrate them in advance using an FEM-based simulation. Specifically, we construct an elastomer model in a finite element simulator (Isaac Gym [20] in our case, though other simulators are also applicable), with Young’s modulus and Poisson’s ratio matched to those of real-world materials. We conduct loading experiments with varying indentation locations, depths, and indenter orientations to generate diverse data. At each time step, we record the index of active nodes and the displacement tensor of all surface nodes. The collected data are then used to minimize the following loss function via gradient descent:

$$Loss = \min_{\alpha, \Phi_p} \|\mathbf{U}_p - \mathbf{U}_p^*\|_2 \quad (3)$$

where \mathbf{U}_p^* denotes oracle displacements of passive nodes from FEM simulation.

B. RL Training: Tactile-Based Dexterous Blind Grasping

To demonstrate ETac’s capability in enabling scalable reinforcement learning, we develop a tactile-based dexterous blind grasping task. Here, we used ETac as the tactile

simulation’s backend. The resultant displacement tensor from ETac is provided as input to the manipulation policy. This state variable design follows protocols in [32], [33].

The objective of the task is to manipulate and grasp a variety of objects using a ShadowHand manipulator [34] in a vision-deprived setting, thereby emphasizing the usage of simulated tactile feedback. The manipulated objects include both geometric primitives and common household items such as an egg, a soda can, and a cube. Each grasp trial begins with initial contact between the hand and the object, with both the object’s pose and the contact location randomized. As the object may not be centered in the palm, the hand must gradually conform its fingers to the object’s surface and ultimately lift it.

We train the policy using Proximal Policy Optimization (PPO) [35]. The input state to the policy is defined as $S_t = (s_t^r, I_t, a_{t-1})$, where s_t^r represents robot joint’s proprioceptive states, I_t denotes tactile observations (i.e., the displacement tensor described above), and a_{t-1} is the previous action. Given this input, the policy outputs a six DoF wrist translation and rotation in Euler angles (floating base controlled using PD controllers), along with 18 joint torque values applied to the actuators. The reward function is defined below:

$$r = r_{\text{proximity}} + r_{\text{lift}} + r_{\text{goal}} + r_{\text{contact}} \quad (4)$$

where the proximity reward $r_{\text{proximity}}$ encourages minimizing the distance between the hand and the object. The contact reward r_{contact} promotes broader contact areas to enhance grasp stability. Once a grasp is established, the lift reward r_{lift} is activated to encourage object lifting, following the design in [36]. Finally, the goal reward r_{goal} provides a bonus for successful grasping and penalizes failures. Details of the reward design are provided in Appendix section.

IV. EXPERIMENTS

In this section, we evaluate ETac through experiments to address two questions: (Q1) Does it provide high-quality

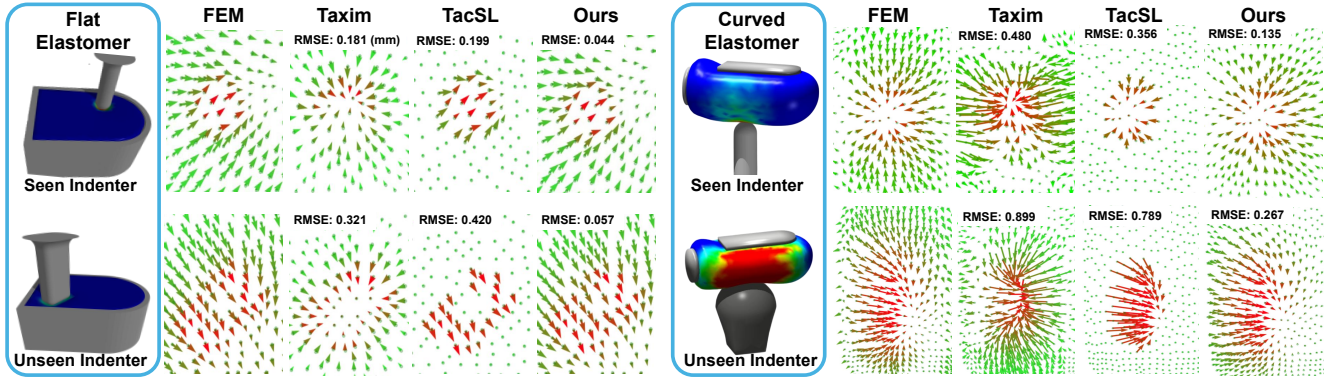


Fig. 4: **Comparison of elastomer surface displacement fields estimated by different methods.** Arrows show node-wise displacement direction and magnitude, with color shifting to red as z -axis displacement increases.

TABLE II: The average RMSE (mm) between estimated displacements and the FEM simulation results.

	Flat Elastomer	Curved Elastomer
TacSL [19]	0.194 ± 0.076	0.445 ± 0.079
Taxim [18]	0.163 ± 0.048	0.447 ± 0.358
Ours (Linear Decay Only)	0.151 ± 0.058	0.256 ± 0.062
Ours (Residual Network Only)	0.074 ± 0.032	0.128 ± 0.055
Ours (Full Pipeline)	0.058 ± 0.034	0.116 ± 0.049

tactile information? and (Q2) Is it efficient for parallel RL training of dexterous manipulation policies?

A. Q1: Simulation Fidelity

Deformation Estimation. We first evaluate ETac’s simulation quality by estimating sensor elastomer deformation. Indentation experiments are conducted on two elastomer types: a flat elastomer [37] and a curved BioTac-style elastomer [32]. The experiments were with eight indenters, each with over 12,000 frames of data. Loading positions cover the full elastomer surface with depths from 0 to 4 mm.

Using this dataset, we compare ETac with two particle-based simulators: Taxim [18] and TacSL [19]. The quality metric is based on displacements’ node-wise RMSE against FEM ground truth (Tab. II). On the flat elastomer, ETac achieves 0.058 mm RMSE, outperforming TacSL (0.194 mm) and Taxim (0.163 mm). The advantage is larger on the curved elastomer surface, with ETac at 0.116 mm versus 0.445 mm (TacSL) and 0.447 mm (Taxim). Fig. 4 shows qualitative results from four randomly selected cases. In all examples, ETac most closely matches the FEM results and achieves the lowest RMSE. By contrast, Taxim [18] performs well on flat surfaces but struggles with curved deformations, while TacSL [19] produces noticeably sparser outputs due to the absence of a deformation propagation mechanism.

Furthermore, we conduct an ablation study to evaluate the necessity of decomposing ETac’s propagation model into decay-based propagation and residual correction components. For this evaluation, we retrained two ablated baselines. Using only decay-based propagation significantly degrades performance, with RMSE increasing to 0.151 mm on the flat elastomer and 0.256 mm on the curved elastomer. Relying solely on the residual network structure yields better results (0.074 mm and 0.128 mm, respectively) but still falls short

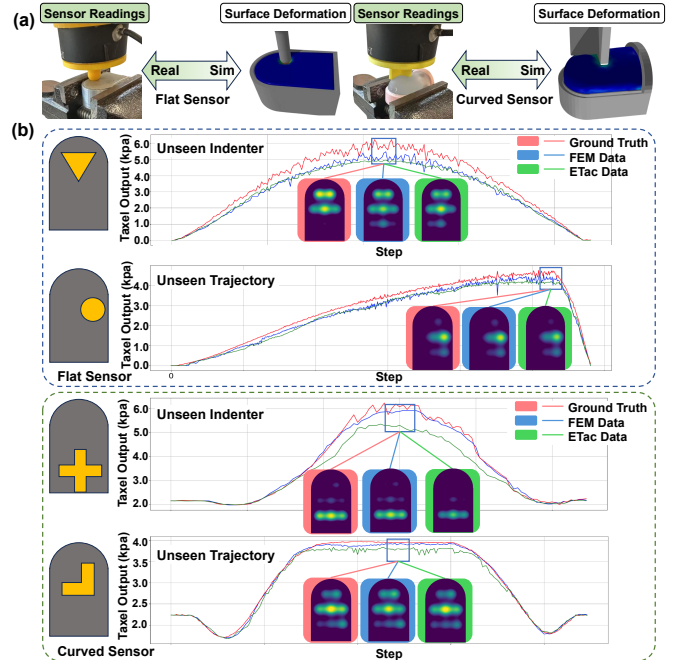


Fig. 5: **Predicting signals of real sensors.** (a) Paired data collection in real and simulated settings. (b) Comparison of sensor output predictions using ETac and FEM data. “Unseen Trajectory” denotes novel loading patterns with seen indenters.

of the full model with better physical prior knowledge for modeling attenuation. Their combination achieves the best performance in deformation estimation.

Real-to-Sim Tactile Sensor Modeling. Moreover, we validated ETac’s ability to simulate real sensors by replicating the experimental setup from [37]. The goal is to reproduce the outputs of the tactile sensors consisting of eight MEMS-based pressure-sensing taxels. The evaluation is conducted on two types of elastomers: one flat and one curved. While the original study [37] used Isaac Gym’s FEM backend for Real2Sim, we employed ETac for benchmarking. We collected paired data by subjecting both real sensors and their simulated counterparts to identical indentation conditions with four different indenters (Fig. 5(a)). During this process, we recorded signals from the real sensor’s eight taxels and

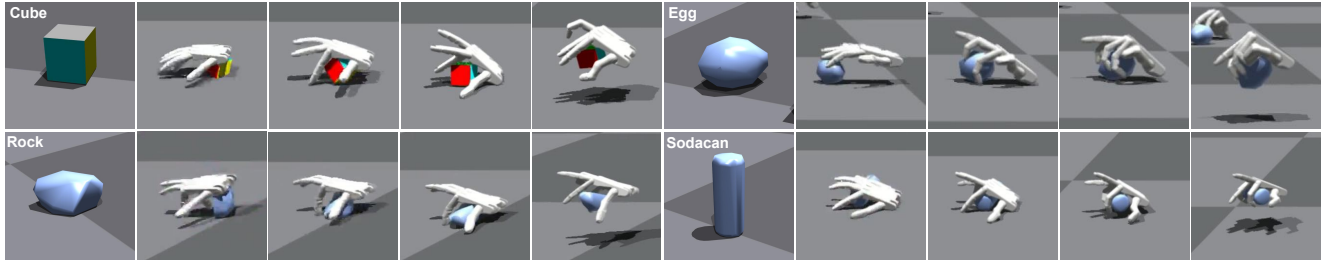


Fig. 6: **Demonstrations of Blind Grasping Skill.** The process of successfully grasping 4 different objects over time.

TABLE III: Success rates (%) of blind grasping policies under different settings.

	Cube	Egg	Rock	Sodacan	Average
Baseline (Object Pose)	63.79 ± 36.25	72.05 ± 30.22	47.66 ± 43.63	68.70 ± 12.12	62.97 ± 37.82
Fingertip Sensors	74.35 ± 5.24	86.46 ± 10.57	70.36 ± 15.30	60.43 ± 3.01	72.90 ± 21.06
Full-hand Sensors	86.95 ± 4.65	92.98 ± 3.91	78.42 ± 9.03	82.45 ± 11.03	84.45 ± 13.09

corresponding simulated surface deformations from ETac and FEM. This dataset was then used to train a PointNet-based model for predicting real sensor responses (refer to Appendix). Qualitative results are shown in Fig. 5(b), including the time series response of the most heavily loaded taxel and the outputs of all eight taxels at peak load. Our analysis shows that the model trained using ETac data achieved L1 losses of 3.94% and 3.61% for the flat and curved sensors, respectively, comparable to the FEM backend (2.46% and 2.75%). These findings demonstrate that ETac is a viable alternative to FEM for data-driven sim-to-real learning.

Based on all the experimental results, we conclude that ETac provides high-quality information for tactile simulation (Q1).

B. Q2: Evaluation on RL-based Blind Grasping

We next evaluate ETac’s effectiveness for training reinforcement learning policies in a vision-deprived grasping task (refer to Sec. III-B). We tested three sensor settings:

- 1) **Full-Hand Sensors:** The tactile sensor covers the front side of the hand, demonstrating the simulator’s capability to handle a large number of nodes (1,000 nodes).
- 2) **Fingertip Sensors:** Tactile sensors cover robot’s fingertips (150 nodes). This follows conventional Shadow Hand which has 5 BioTac fingertip sensors [34].
- 3) **Baseline:** As a comparison baseline, the hand is not equipped with any tactile sensors. Instead, the oracle 6D pose of the object is provided as input to the policy. This setting follows the configuration used in [36].

Computational Efficiency. A key factor for ETac’s usability in parallel RL training is its computational efficiency. We evaluated this in the blind grasping task, comparing ETac with Taxim [18], TacSL [19], and an FEM solver (Fig. 7). When simulating a full-hand sensory setup on an RTX 4090 desktop GPU (24 GB RAM), the FEM solver can handle only up to 32 environments in parallel, whereas ETac efficiently scales to support up to 4096 environments. At 32 environments, ETac maintains 14.85 FPS per environment, which is faster than FEM (2.38 FPS per env). For total FPS

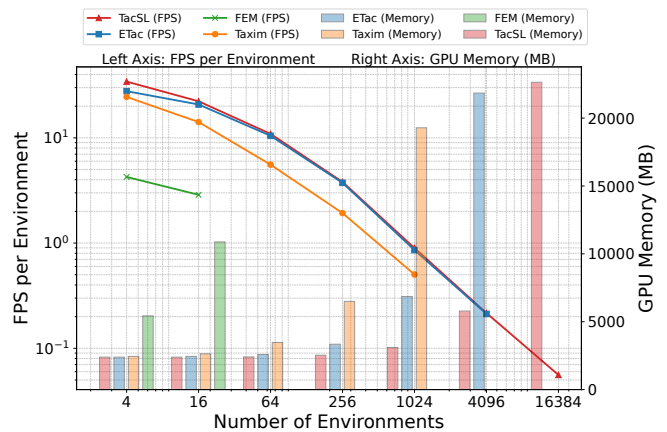


Fig. 7: Comparison of RL training performance.

achieved, ETac scales efficiently with more environments: reaching 669, 956, 878, and 869 total FPS at 64, 256, 1024, and 4096 environments, respectively. This outperforms Taxim (508, 620, 650 FPS at 64, 256, 1024 environments, but out-of-memory when scale up to 4096 envs) and is comparable to TacSL (698, 975, 918, 886 FPS at 64, 256, 1024, 4096 environments). ETac’s memory usage is lower than Taxim’s due to less model parameters, but higher than TacSL’s because of its propagation calculation. This trade-off enables higher-fidelity deformation estimation, making ETac suitable for tactile-based dexterous manipulation.

Grasping Performance. Our tactile simulation enables the successful acquisition of blind grasping skills. The learned skill performances are shown in Tab. III. Specifically, the *Full-Hand Sensor* setup achieved the highest performance, with an average grasp success rate of 84.45% across four objects. Fig. 6 illustrates successful grasping processes under this configuration, where the hand continuously adjusted its pose based on tactile feedback, gradually enclosed and ultimately lifted the object. This demonstrates that ETac effectively provides rich contact information for perception, which is crucial for successful training. When limited to *Fingertip Sensors*, the average success rate decreased to 72.90%. Nonetheless, both tactile configurations outperformed the

baseline without tactile sensing, which relied solely on the object’s pose and could not reflect physical characteristics such as size and shape. These results confirm that ETac provides effective tactile information that support parallel RL training for acquiring tactile dexterous manipulation policies (Q2).

V. CONCLUSIONS

In this paper, we introduced ETac, a novel framework for efficient and high-fidelity tactile sensor simulation. By distilling high-fidelity FEM contact dynamics to a lightweight data-driven deformation propagation model, ETac accurately estimates elastomer surface deformations, achieving performance comparable to FEM and superior to baselines. ETac supports manipulation skill learning in two ways. First, it enables the real-to-sim modeling for tactile sensors by functioning as a simulation backend. Second, its low computational cost enables large-scale parallel RL training. We demonstrated its utility in a blind grasping task, where ETac achieved 864 FPS when simulating full-hand tactile sensing across 4,096 environments. Using ETac, we successfully trained policies that leverage simulated full-hand tactile feedback to lift objects, highlighting its potential for contact-rich manipulation.

While ETac demonstrates significant advantages in efficiency and simulation fidelity, we acknowledge several limitations. First, regarding scalability in calibration, ETac relies on high-fidelity FEM simulations to calibrate the propagation model. Extending it to new sensor geometries or materials currently involves creating specific oracle models and re-calibration, imposing a pre-computation overhead. Second, concerning application scope, this work focuses on validating the framework’s capacity for large-scale parallel training. Although sensor-level real-to-sim alignment is validated, full sim-to-real transfer of the learned blind grasping policy remains our ongoing work. Finally, in terms of interaction dynamics, ETac is currently integrated with a rigid-body physics backend. Although it accurately estimates the deformation of sensor elastomers, the two-way coupling dynamics (specifically the influences onto object states) is simplified, limiting support for deformable objects and soft–soft contact. These limitations outline the improvement directions for future work.

APPENDIX

A. Reward Design for Training Blind Grasping Policy

Our reward design used for blind grasping policy (Sec. III-B) comprises four sub-goals. Here we give a detailed description of each term (note that all the ω_{**} here denote weight hyper-parameters):

Proximity Reward. First, we encourage hand to be positioned within close proximity to the object:

$$r_{proximity} = -\omega_{ho} \cdot D_{ho} - \omega_{fo} \cdot D_{fo} \quad (5)$$

where D_{ho} is the distance between the object and the origin of the hand, D_{fo} denotes the summed distances from the object to all fingertips.

Lifting Reward. When the following proximity conditions are satisfied: $D_{ho} \leq \lambda_{l1}$ and $D_{fo} \leq \lambda_{l2}$ (λ_{l1} and λ_{l2} are user-defined thresholds), the lifting behavior is activated. In this case, the agent continuously receives a lifting reward:

$$r_{lift} = \omega_{lift} \cdot (f_z - 2 \cdot \Delta h) \quad (6)$$

where f_z is the force exerted on the hand root along the lifting direction (z-axis), and Δh denotes the difference between the target height and the current object height. This follows [36].

Contact Reward. To prevent slippage, it is beneficial to enlarge the contact area between the hand and the object. To encourage this, we define the contact reward $r_{contact}$ as:

$$r_{contact} = \text{nonzero}(S_{cr})/\text{len}(S_{cr}) + \omega_a \cdot \begin{cases} R_a & \text{if } n_a > 0 \\ -0.1 & \text{otherwise} \end{cases} \quad (7)$$

where n_a denotes the number of active nodes (i.e., nodes in direct contact), and $R_a = n_a/n$ represents their proportion relative to the total number of nodes n . The one-hot vector S_{cr} encodes the contact between the object and different linkages of the hand (e.g., the palm, the proximal, middle, and distal of each finger). The term $\text{nonzero}(S_{cr})/\text{len}(S_{cr})$ aims to distribute the contact points across multiple fingers.

Goal Reward. Finally, a goal reward is assigned based on whether hand has successfully grasped the object:

$$r_{goal} = \begin{cases} \omega_{goal} & \text{if success} \\ -2 \cdot \omega_{goal} & \text{if fail} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

the success condition is: $\Delta h \leq \lambda_{goal}, n_a > 0$, with λ_{goal} as a user-defined threshold. A failure occurs if the object is not lifted within the allotted time or if it falls after being lifted.

In our implementation, we set $\omega_{ho} = 1.0$, $\omega_{fo} = 0.8$, $\omega_{lift} = 1.5$, $\omega_a = 5.0$, $\omega_{goal} = 1.0$, and $\lambda_{l1} = 0.12$, $\lambda_{l2} = 0.25$ and $\lambda_{goal} = 0.03$, respectively. We trained the policy with PPO [35] (Sec. IV-B) using 1024 parallel environments. The PPO hyperparameters are shown in Tab. IV.

TABLE IV: PPO hyperparameters for blind grasping policy.

Hyperparameters	Value
Hidden Layers in Actor Network	[512, 512, 256, 128]
Hidden Layers in Critic Network	[512, 512, 256, 128]
Activation	elu
Initial Learning Rate	5e-4
Learning Rate Schedule	adaptive
Mini Batch Size	32
PPO clip parameter	0.2
GAE λ	0.95
Discount Factor γ	0.99

B. Modeling Real Sensors

We trained a PointNet-based regressor to map simulated displacement fields to real sensor outputs (Sec. IV-A). The architecture of the regressor is illustrated in Fig. 8. Residual connections are added to improve gradient back propagation, while attention weighting replaces max pooling in global feature extraction (following [38]). The self-attention module

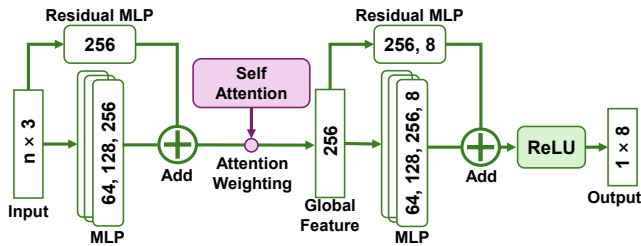


Fig. 8: Network used for sensor output prediction.

consists of an MLP with shape [256, 128, 1], with tanh as activation. The resulting attention scores are normalized using a softmax function.

REFERENCES

- [1] A. Murali, Y. Li, D. Gandhi, and A. Gupta, "Learning to grasp without seeing," in *International Symposium on Experimental Robotics*, pp. 375–386, Springer, 2018.
- [2] N. Guo, X. Han, S. Zhong, Z. Zhou, J. Lin, J. S. Dai, F. Wan, and C. Song, "Proprioceptive state estimation for amphibious tactile sensing," *IEEE Transactions on Robotics*, 2024.
- [3] K. Ganguly, B. Sadfaridpour, K. B. Kidambi, C. Fermler, and Y. Aloimonos, "Grasping in the dark: Compliant grasping using shadow dexterous hand and biotac tactile sensor," *arXiv e-prints*, pages arXiv–2011, 2020.
- [4] Z. Zhao, W. Li, Y. Li, T. Liu, B. Li, M. Wang, K. Du, H. Liu, Y. Zhu, Q. Wang, *et al.*, "Embedding high-resolution touch across robotic hands enables adaptive human-like grasping," *arXiv preprint arXiv:2412.14482*, 2024.
- [5] Z. Si, T. C. Yu, K. Morozov, J. McCann, and W. Yuan, "Robotsweater: Scalable, generalizable, and customizable machine-knitted tactile skins for robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10352–10358, IEEE, 2023.
- [6] D. Crowder, K. Vandeyck, X. Sun, J. McCann, and W. Yuan, "Social gesture recognition in sphri: Leveraging fabric-based tactile sensing on humanoid robots," *arXiv preprint arXiv:2503.03234*, 2025.
- [7] G. Zhang, Y. Du, Y. Zhang, and M. Y. Wang, "A tactile sensing foot for single robot leg stabilization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14076–14082, IEEE, 2021.
- [8] J. Yin, H. Qi, J. Malik, J. Pikul, M. Yim, and T. Hellebrekers, "Learning in-hand translation using tactile skin with shear and normal force sensing," *arXiv preprint arXiv:2407.07885*, 2024.
- [9] J. Tegin and J. Wikander, "Tactile sensing in intelligent robotic manipulation—a review," *Industrial Robot: An International Journal*, vol. 32, no. 1, pp. 64–70, 2005.
- [10] A. Yamaguchi and C. G. Atkeson, "Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision?," *Advanced Robotics*, vol. 33, no. 14, pp. 661–673, 2019.
- [11] J. Wang, Y. Yuan, H. Che, H. Qi, Y. Ma, J. Malik, and X. Wang, "Lessons from learning to spin" pens," *arXiv preprint arXiv:2407.18902*, 2024.
- [12] K.-W. Lee, Y. Qin, X. Wang, and S.-C. Lim, "Dextouch: Learning to seek and manipulate objects with tactile dexterity," *arXiv preprint arXiv:2401.12496*, 2024.
- [13] L. Lan, D. M. Kaufman, M. Li, C. Jiang, and Y. Yang, "Affine body dynamics: Fast, stable & intersection-free simulation of stiff materials," *arXiv preprint arXiv:2201.10022*, 2022.
- [14] Z. Si, G. Zhang, Q. Ben, B. Romero, Z. Xian, C. Liu, and C. Gan, "Diff tactile: A physics-based differentiable tactile simulator for contact-rich robotic manipulation," *arXiv preprint arXiv:2403.08716*, 2024.
- [15] J. Xu, S. Kim, T. Chen, A. R. Garcia, P. Agrawal, W. Matusik, and S. Sueda, "Efficient tactile simulation with differentiability for robotic manipulation," in *Conference on Robot Learning*, pp. 1488–1498, PMLR, 2023.
- [16] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*, pp. 91–100, PMLR, 2022.
- [17] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra, "Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3930–3937, 2022.
- [18] Z. Si and W. Yuan, "Taxim: An example-based simulation model for gelsight tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2361–2368, 2022.
- [19] I. Akinola, J. Xu, J. Carius, D. Fox, and Y. Narang, "Tacsl: A library for visuotactile sensor simulation and learning," *arXiv preprint arXiv:2408.06506*, 2024.
- [20] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.
- [21] J. Zhao and E. H. Adelson, "Gelsight svelte hand: A three-finger, two-dof, tactile-rich, low-cost robot hand for dexterous manipulation," *ArXiv*, vol. abs/2309.10886, 2023.
- [22] B. Romero, H.-S. Fang, P. Agrawal, and E. H. Adelson, "Eyesight hand: Design of a fully-actuated dexterous robot hand with integrated vision-based tactile sensors and compliant actuation," *ArXiv*, vol. abs/2408.06265, 2024.
- [23] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang, "Rotating without seeing: Towards in-hand dexterity through touch," *arXiv preprint arXiv:2303.10880*, 2023.
- [24] Y. Yuan, H. Che, Y. Qin, B. Huang, Z.-H. Yin, K.-W. Lee, Y. Wu, S.-C. Lim, and X. Wang, "Robot synesthesia: In-hand manipulation with visuotactile sensing," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6558–6565, IEEE, 2024.
- [25] Y. Chen, C. Wang, L. Fei-Fei, and C. K. Liu, "Sequential dexterity: Chaining dexterous policies for long-horizon manipulation," 2023.
- [26] Y. Li, W. Du, C. Yu, P. Li, Z. Zhao, T. Liu, C. Jiang, Y. Zhu, and S. Huang, "Taccel: Scaling up vision-based tactile robotics via high-performance gpu simulation," *arXiv preprint arXiv:2504.12908*, 2025.
- [27] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang, "A moving least squares material point method with displacement discontinuity and two-way rigid body coupling," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [28] Y. Hu, J. Liu, A. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik, "Chainqueen: A real-time differentiable physical simulator for soft robotics," in *2019 International conference on robotics and automation (ICRA)*, pp. 6265–6271, IEEE, 2019.
- [29] C. Fuji Tsang, M. Shugrina, J. F. Lafecche, T. Takikawa, J. Wang, and C. e. a. Loop, "Kaolin: A pytorch library for accelerating 3d deep learning research." <https://github.com/NVidiaGameWorks/kaolin>, 2022.
- [30] O. I. Vinogradova and F. Feuillebois, "Interaction of elastic bodies via surface forces: 2. exponential decay," *Journal of colloid and interface science*, vol. 268, no. 2, pp. 464–475, 2003.
- [31] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.
- [32] Y. Narang, B. Sundaralingam, M. Macklin, A. Mousavian, and D. Fox, "Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6444–6451, IEEE, 2021.
- [33] W. Z. E. Amri, M. Kuhlmann, and N. Navarro-Guerrero, "Transferring tactile data across sensors," *arXiv preprint arXiv:2410.14310*, 2024.
- [34] "Shadowrobot." <https://www.shadowrobot.com/dexterous-hand-series/>.
- [35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [36] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen, *et al.*, "Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4737–4746, 2023.
- [37] X. H. *et al.*, "Twintac: A wide-range, highly sensitive tactile sensor with real-to-sim digital twin sensor model," *arXiv preprint arXiv:2509.10063*, 2025.
- [38] Y. Yang, Y. Ma, J. Zhang, X. Gao, and M. Xu, "Attnet: Attention-based deep neural network for 3d point set analysis," *Sensors*, vol. 20, no. 19, p. 5455, 2020.