TouchGo: Self-Supervised Visuo-Tactile Pretraining to contact deformation representation learning via multi-sensor

Abstract— **Optical tactile sensors provide robots with rich force information for robot grasping in unstructured environments. The fast and accurate calibration of three-dimensional contact forces holds significance for new sensors and existing tactile sensors which may have incurred damage or aging. However, the conventional neural-network-based force calibration method necessitates a large volume of force-labeled tactile images to minimize force prediction errors, with the need for accurate Force/Torque measurement tools as well as a time-consuming data collection process. To address this challenge, we propose a novel deep domain-adaptation force calibration method, designed to transfer the force prediction ability from a calibrated optical tactile sensor to uncalibrated ones with various combinations of domain gaps, including marker presence, illumination condition, and elastomer modulus. Experimental results show the effectiveness of the proposed unsupervised force calibration method, with lowest force prediction errors of 0.102N (3.4% in full force range) for normal force, and 0.095N (6.3%) and 0.062N (4.1%) for shear forces along the x-axis and y-axis, respectively. This study presents a promising, general force calibration methodology for optical tactile sensors.**

Contact-rich manipulation remains a major challenge in robotics. Optical tactile sensors like GelSight Mini offer a low-cost solution for contact sensing by capturing soft-body deformations of the silicone gel. However, accurately inferring shear and normal force distributions from these gel deformations has yet to be fully addressed. In this work, we propose a machine learning approach using a U-net architecture to predict force distributions directly from the sensor's raw images. Our model, trained on force distributions inferred from fea, demonstrates promising accuracy in predicting normal and shear force distributions. It also shows potential for generalization across sensors of the same type and for enabling real-time application.

I. Introduction

Tactile sensing plays an important role in advancing the state-of-the-art in robotic manipulation [?], [?], [?], [?], [?], [?], [?], [?]. Successful applications include grip adaptation through slip detection [?], [?], [?], medical procedures [?], [?] and tele-operation [?].

In particular, optical tactile sensors have emerged as a promising technology for capturing contact information due to their high spatial resolution, multimodal sensing capabilities—including shape [?], hardness [?], texture [?], and temperature [?]—and cost-effectiveness [?], [?]. However, many prior works have focused on extracting only low-dimensional tactile information, such as total force [?], [?], [?], limiting operational flexibility. Access to contact force distributions, on the other hand, would enable better handling of multiple contacts and diverse manipulation scenarios [?].

Conventional methods for extracting force distributions require calculating the three-dimensional deformation of the contact medium and utilizing elasticity theory [?], [?], [?], [?]. Yet, accounting for non-linear material behavior, such as with fea, is computationally intensive and unsuitable for real-time applications.

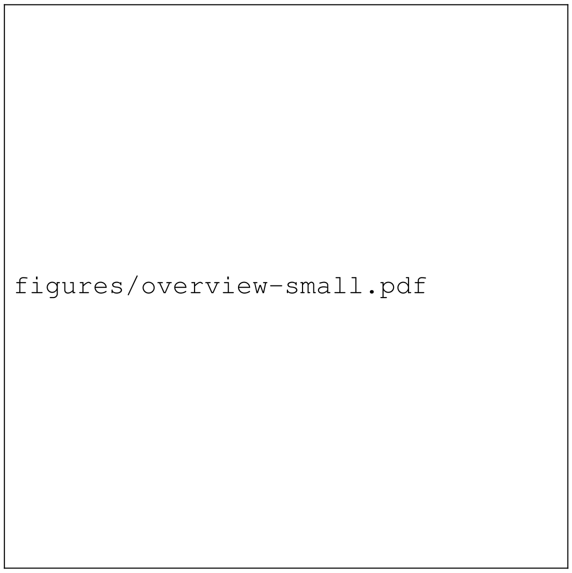
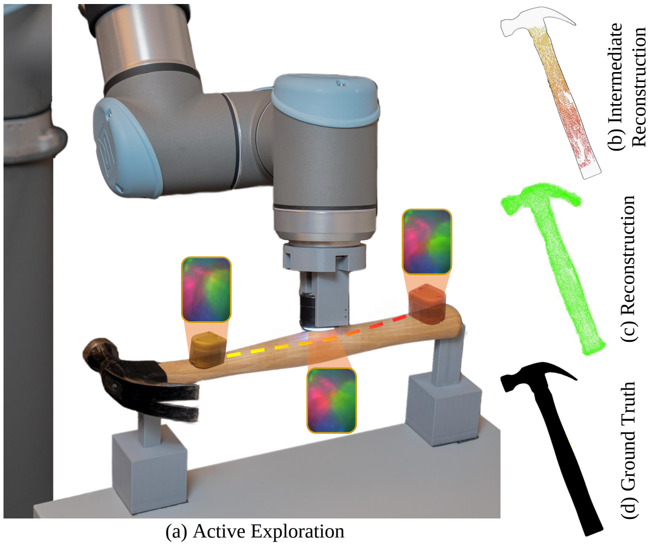


Fig. 1: Complete Method Overview: from data collection to force distribution prediction. After data collection in a precisely calibrated setup with a CNC milling machine, Finite Element Analysis is employed to generate labels ("ground truth" force distributions). Using the labels and raw images captured by the GelSight Mini tactile sensor, we train a U-net for efficiently mapping raw tactile images to the corresponding force distributions.

Recent works leverage Deep Learning to address the challenge of real-time force estimation. In [?], Convolutional Neural Networks (CNNs) were used to predict contact forces from sensor images, while [?] introduced CANFnet for estimating normal force distributions at the pixel level. In [?], [?], [?], fea-derived data was used to train a model for predicting force distributions, demonstrating the effectiveness of combining simulations with data-driven methods.

In this paper we introduce FEATS (see Fig. 1)—a machine learning approach that directly maps raw tactile images to force distributions, building upon the method by Sferrazza et al. [?]. We utilize FEA to generate labeled data for training, ensuring accurate ground truth across various indenters and force levels. A U-net neural network architecture [?] is employed to estimate force distributions from images captured by the GelSight Mini optical sensor [?], [?]. In contrast to [?], our method is tailored to a widely available commercial sensor GelSight Mini, dropping the requirement of a custom-made gel with immersed particles, thereby drastically



(a) Active Exploration

Fig. 2: **Reconstruction of a hammer**. (a) showcases the trajectory of the tactile sensor in 3D space. (b) depicts the intermediate tactile readings on the hammer's surface, with the color gradient representing the passage of time. Following thorough tactile exploration, we achieve a complete object reconstruction (c), highlighting the effectiveness of our active strategy in exploring the entire reachable surface.

extending the applicability of the approach. Furthermore, this sensor allows for a significantly expanded range of measurable forces $[0 - 40]N$, an 8-fold increase in the maximum measurable force compared to [**?**]. Finally, we open source our code, dataset and model, aiding reuse and reproducibility.

Experimental results demonstrate that the proposed method accurately predicts high-dimensional contact force distributions from raw tactile images. This capability advances robotic manipulation by accommodating a wider range of contact scenarios and offers a versatile representation applicable to downstream tasks.

II. Related Works

Extracting meaningful contact-related information from the raw RGB images of optical tactile sensors is a major challenge in visual-tactile perception [**?**], [**?**], [**?**], [**?**], [**?**], [**?**]. A number of methods have been proposed for constructing or learning such "tactile representations".

A. Marker Displacement Methods

Li et al. [**?**] posit that it is the contact layer deformations that capture the crucial information within tactile images. By analyzing these deformations, various contact features can be extracted, with mdm being the most common approach [**?**]. In mdm, markers are placed on or within the elastomer and appear as features in the sensor's imagery (Fig. 1). For the GelSight sensor [**?**], [**?**], markers were first introduced in [**?**] to study normal and shear forces, along with slip dynamics. They identified a linear relationship between loads and marker motion, but this applied only in non-slip conditions. Beyond marker motion, optical sensors can capture detailed height maps and contact geometry through careful illumination and photometric stereo [**?**]. These height maps can be used to estimate contact forces with a third-degree polynomial [**?**].

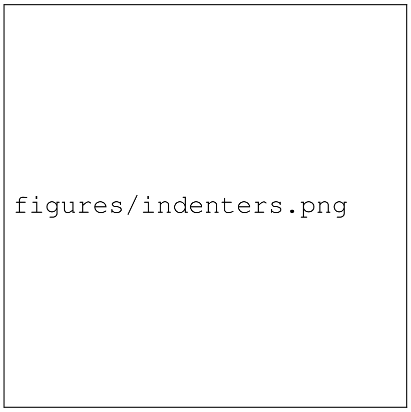


Fig. 3: Models of 3D-printed indenters used for data collection. Different colors represent groups of indenters with similar shapes.

In this paper, we use a gel with markers, but their movement is not explicitly tracked. Instead, they serve as implicit features within the sensor image, which is analyzed by a neural network to predict force distributions.

B. Deep Learning-Based Tactile Representations

Advancements in computer vision directly translate to vision-based tactile sensing. Models such as CNNs and LSTMs were adapted to assess object hardness [**?**] and grip stability [**?**], whereas SVMs were used for lump detection [**?**]. More tactile-specific deep learning methods have been developed for overall force prediction [**?**] and for pixel-wise contact area and normal force estimation [**?**].

Building on the demonstrated effectiveness of deep neural networks for feature extraction and prediction, we employ a U-net architecture similar to that of [**?**]. However, in contrast to [**?**], FEATS estimates both normal and shear forces, thus providing a physically grounded representation in the form of a 3D force distribution acting upon the sensor.

C. Force Distribution Prediction Through Elasticity Theory

Elasticity theory has been effectively applied to create more refined and accurate load distributions acting upon the soft silicone gel of optical tactile sensors. In [**?**], elasticity theory with mdm was used to derive force vectors from marker movements assuming a linear elastic, uniform and half-spaced material. This method was later adopted in [**?**] for the GelSlim sensor [**?**]. More recently, sensors enabling 3D surface deformation reconstruction have been proposed, such as TacLINK [**?**] and Tac3D [**?**]. They compute force distributions from measured 3D marker displacements.

However, direct prediction of force distributions from displacements, usually through a linear stiffness matrix, does not account for the nonlinearities of soft elastomers. Sun et al. [**?**] addressed this limitation by employing ResNet [**?**], which was trained on sensor images with approximated force distributions. Similarly, Sferrazza et al. [**?**] utilized a dnn trained on image features with force distributions obtained

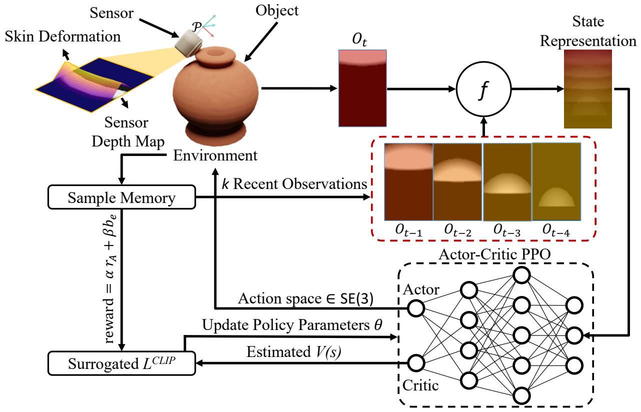


Fig. 4: **Overview.** This figure illustrates the key steps and components of TouchGo in a scenario where the sensor moves upward along the jar's edge. We employed Temporal Tactile Averaging for state representation f (Sec. **??**) to encode consecutive observations, enabling the perception of movement on the sensor vital for learning dexterous actions. We also incorporate an Upper Confidence Bound (UCB) exploration as a bonus to encourage effective exploration.

from fea, and later improved this method by incorporating simulated training data [**?**], [**?**].

Building on these approaches, we also aim to estimate contact forces using supervised deep learning. However, instead of working on optical flow features or grayscale images [**?**] applicable to the custom-made sensor with a dense 3D-marker field [**?**], [**?**], [**?**], we use raw RGB images from a widely available GelSight Mini sensor. Crucially, we develop specific procedures for data collection and model training that enable the efficient use of this widely accessible sensor, thereby significantly lowering the entry barrier into the field.

Our key contributions are: i) method for creating force labels from FEA outputs tailored to GelSight Mini + implementation in CalculiX, ii) data collection procedure + dataset, iii) trained model applicable to varying objects and gels.

III. Method

In this section, we introduce AnyTouch, a unified multi-sensor tactile representation learning framework from the perspectives of both static and dynamic perception, as shown in Figure **??**. We integrate the input format of tactile images and videos (Sec. III-A) and focus on learning both fine-grained pixel-level details for refined tasks (Sec. III-B) and semantic-level sensor-agnostic features for understanding properties (Sec. III-C) and building unified space (Sec. III-D) by a multi-level structure. We also propose universal sensor tokens for better knowledge transfer.

A. *Unified input format for Static and dynamic tactile Perception*

In daily life, human tactile perception includes both static and dynamic processes. A brief touch allows quick recognition of properties like material and texture, while tasks such as unlocking a lock require continuous dynamic perception. These two types of perception complement each other, enabling us to comprehensively understand the physical surroundings and engage in a variety of interactions. This inspires us to learn unified multi-sensor representation from the perspective of combining static and dynamic perception, using tactile images and videos respectively.

Given a static tactile image $I \in \mathbb{R}^{1\times H\times W\times 3}$ and a dynamic tactile video clip $V \in \mathbb{R}^{F\times H\times W\times 3}$, we consider tactile images as single-frame static videos to unify tactile images and videos. Concretely, we replicate I along the time axis for F times, and use a unified 4-D tensor $X_T \in \mathbb{R}^{F\times H\times W\times 3}$ to represent both I and V as [**?**], [**?**], where F is the number of frames and H, W denote the shape of images. We then process $X_T \in \mathbb{R}^{F\times H\times W\times 3}$ into spatio-temporal tokens $z \in \mathbb{R}^{N\times d}$ through a shared patch projection layer, where N is the length of tokens and d represents the feature dimension. By unifying the processing of images and videos in this manner, our approach integrates tactile images and video input, enhancing the model's ability to comprehend both static and dynamic information, and endows the model with the potential to accomplish various tasks.

B. *Masked Modeling: learning Pixel-level Details*

Visuo-tactile images are fine-grained data with pixel-level details of subtle tactile deformations and continuous changes during dynamic processes, especially for refined perception tasks. To enhance the fine-grained perception capabilities of the tactile representation model, we employ the masked autoencoder technique he2022masked,tong2022videomae, compelling the model to capture pixel-level details across multiple sensors. Concretely, we randomly mask the tokens of both tactile images and videos with a masking ratio ρ, and build a decoder to obtain the reconstructed static images $\hat{I}$ and dynamic videos $\hat{V}$. The corresponding loss function $\mathcal{L}^S_{rec}$ and $\mathcal{L}^D_{rec}$ are *mean squared error* (MSE) loss between the original masked tokens and reconstructed ones in the pixel space:

$$\mathcal{L}^S_{rec} = \frac{1}{|\Omega_M|}\sum_{p\in\Omega_M}|\hat{I}(p)-I(p)|^2,\ \ \mathcal{L}^D_{rec} = \frac{1}{F|\Omega_M|}\sum_{f}^{F}\sum_{p\in\Omega_M}|\hat{V}_f(p)-V_f(\tag{1}$$

where p is the token index, Ω_M is the set of masked tokens and V_f is the f-th frame in the video V. We use masked modeling to learn fine-grained tactile deformation features at the pixel level, as well as the temporal dynamics of tactile changes.

To further enhance the model's understanding of continuous deformation changes, we introduce an additional task of predicting the next frame V_{F+1} while reconstructing the dynamic video V. The loss function $\mathcal{L}^D_{pred}$ is MSE loss between the original frame V_{F+1} and the predicted frame $\hat{V}_{F+1}$:

$$\mathcal{L}^D_{pred} = \frac{1}{N}\sum_{p}^{N}|\hat{V}_{F+1}(p) - V_{F+1}(p)|^2. \tag{2}$$

C. *Multi-modal Aligning: understanding semantic-level properties*

After obtaining tactile representations with fine-grained perceptual details via masked modeling, we aim to further understand semantic-level tactile properties and use paired multi-modal data as a bridge to narrow the gap between sensors. Therefore, we propose using multi-modal aligning, which binds data from various sensors with paired modalities for a more comprehensive semantic-level perception and reduce perceptual differences between sensors. However, differences in data collection scenarios across various datasets (*e.g.*, simulation vs. reality) make simple vision-tactile alignment less effective in bridging sensor gaps. Therefore, we select the text modality, which consistently describes tactile attributes across datasets, as an anchor to align touch, vision, and text modalities. Since tri-modal tactile datasets are rare, with most containing only vision-touch pairs, we explore two strategies: automatically expanding the amount of text modality pairings and designing aligning methods that are compatible with missing modalities. We first select representative datasets for each sensor and then use GPT-4o to generate or expand the text modality within these datasets. Through this method, we create new text pairs for 1.4 million samples across the four datasets.

Based on these extensive tactile datasets, we develop a modality-missing-aware touch-vision-language contrastive learning method to leverage the paired data between touch and other modalities for alignment. We maximize the use of paired data by selecting the largest subset for each modality combination within the batch for multi-modal aligning. Considering a pair of uni-modal representations (x_T, x_V, x_L) derived from uni-modal encoders, where $x_T \in \mathbb{R}^d$ is the touch representation, $x_V \in \mathbb{R}^d \cup \varnothing$ is the vision representation and $x_L \in \mathbb{R}^d \cup \varnothing$ is the text representation. We then perform multi-modal alignment radford2021clip within the batch as:

$$\mathcal{L}_{T\to V} = -\frac{1}{|\Omega_V|}\sum_{i\in\Omega_V}\log\frac{\exp(x_{T,i}^\top \cdot x_{V,i}/\tau)}{\sum_{j\in\Omega_V}\exp(x_{T,i}^\top \cdot x_{V,j}/\tau)},$$
$$\mathcal{L}_{T\to L} = -\frac{1}{|\Omega_L|}\sum_{i\in\Omega_L}\log\frac{\exp(x_{T,i}^\top \cdot x_{L,i}/\tau)}{\sum_{j\in\Omega_L}\exp(x_{T,i}^\top \cdot x_{L,j}/\tau)},$$
$$\mathcal{L}_{V\to L} = -\frac{1}{|\Omega_V \cap \Omega_L|}\sum_{i\in\Omega_V\cap\Omega_L}\log\frac{\exp(x_{V,i}^\top \cdot x_{L,i}/\tau)}{\sum_{j\in\Omega_v\cap\Omega_L}\exp(x_{V,i}^\top \cdot x_{L,j}/\tau)} \tag{3}$$

where B is the batchsize, Ω_V, Ω_L are sets of indices for the samples containing vision and text, and τ is the scalar temperature. This approach maximizes the use of paired data with missing modalities by aligning the sample intersections between modalities. The computation of $\mathcal{L}_{V\to T}$, $\mathcal{L}_{L\to T}$ and $\mathcal{L}_{L\to V}$ is similar but in the opposite direction. We then obtain the joint aligning loss as:

$$\mathcal{L}_{align} = \frac{\alpha_{TV}}{2}(\mathcal{L}_{T\to V}+\mathcal{L}_{V\to T})+\frac{\alpha_{TL}}{2}(\mathcal{L}_{T\to L}+\mathcal{L}_{L\to T})+\frac{\alpha_{VL}}{2}(\mathcal{L}_{V\to L}+\mathcal{L}_{L\to V}), \tag{4}$$

where α_{TV}, α_{TL} and α_{VL} are hyper-parameters to control the alignment strength.

D. Cross-Sensor Matching: extracting sensor-agnostic features

To fully utilize multi-sensor aligned data and build unified space by clustering multi-sensor tactile representations of the same object, we introduce a novel cross-sensor matching task. In this task, the model needs to determine whether two tactile images or videos are collected from the same position on the same object. We aim to cluster representations of the same tactile information from different sensors while performing multi-modal aligning, thereby enhancing the learning of sensor-agnostic features and forming a unified multi-sensor representation space, as shown in Figure 5.

We treat data collected from the same object and position by two different sensors as a positive pair, and data from different objects or positions as a negative pair. The model is trained to distinguish between positive and negative pairs. For each image and video sample X_T in our TacQuad, we randomly select one sample from the same object at the same location captured by another sensor as the positive sample X_T^+, and choose another sample from any dataset of any other object or location as a negative sample X_T^-. We element-wisely multiply the touch representation x_T with x_T^+ and x_T^-, and then input each result into an MLP to compute the matching scores m^+ and m^-:

$$m^+ = MLP(x_T \cdot x_T^+),\ m^- = MLP(x_T \cdot x_T^-), \tag{5}$$

where x_T, x_T^+ and x_T^- are the representations of X_T, X_T^+ and X_T^-. The loss function $\mathcal{L}_{match}$ is a Binary Cross Entropy Loss similar to [**?**]:

$$\mathcal{L}_{match} = -(y^+ \log(m^+) + (1-y^+)\log(1-m^-)) - (y^- \log(m^-) + (1-y^- \tag{6}$$

This task requires the model to distinguish features with the same semantics from different sensors, thus explicitly clustering representations with the same object information form a unified multi-sensor representation space. As shown in Figure 5, AnyTouch, incorporating this task, differs from existing multi-modal aligning efforts. The construction of this unified multi-sensor representation space can explicitly reduce the gap between sensors and aid in generalizing to unseen sensors.

As both this task and multi-modal aligning focus on semantic-level features, we combine them as the second stage, with masked modeling as the first stage. This multi-level training approach allows us to develop unified multi-sensor representations adaptable to tasks of varying granularities.

E. Universal Sensor Token

In addition to building a multi-sensor representation space, we aim to extract and store information related to each sensor to aid the understanding of input data. More importantly, we want to integrate and effectively transfer this information when generalizing to new sensors. Using sensor-specific tokens is a method for extracting sensor-specific information, but this approach cannot fully transfer information from all seen sensors when generalizing to new sensors yang2024binding.

Therefore, we propose using universal sensor tokens to integrate and store information related to various sensors, thereby maximizing the utilization of multi-sensor data when generalizing to new sensors. During training, we randomly replace the sensor-specific tokens with the universal sensor tokens, expecting them to aid in understanding input data

TABLE I: U-net Mean Absolute Error (MAE) on the Test Set

	MAE$_{\text{GUF}}$ [N]	MAE$_{\text{TF}}$ [N]
f_x	0.0006 ± 0.0006	0.2242 ± 0.4007
f_y	0.0005 ± 0.0003	0.0934 ± 0.1356
f_z	0.0015 ± 0.0010	0.3720 ± 0.4727

ICLR 2025 Template/figures/compare.pdf

Fig. 5: **Comparison with existing multi-modal aligning methods.** Combining the cross-sensor matching task, our method not only uses multi-modal data to bridge the gap between sensors, but also **explicitly** clusters representations of the same position on the same object from different sensors together, constructing a unified multi-sensor representation space.

TABLE II: Model Ablation on the Total Force Estimation Task

Method	MAE$_{\text{TF}}$ [N]		
	f_x	f_y	f_z
ResNet^{13}	$\mathbf{0.085 \pm 0.115}$	$\mathbf{0.069 \pm 0.085}$	1.593 ± 1.131
U-net^{12163}	$\mathbf{0.102 \pm 0.216}$	$\mathbf{0.089 \pm 0.123}$	$\mathbf{0.447 \pm 0.539}$
3U-net^{24321}	0.438 ± 0.585	0.189 ± 0.225	$\mathbf{0.448 \pm 0.523}$
U-net^{24323} (ours)	0.224 ± 0.401	$\mathbf{0.093 \pm 0.136}$	$\mathbf{0.372 \pm 0.473}$
U-net^{48643}	0.318 ± 0.436	0.119 ± 0.184	$\mathbf{0.459 \pm 0.516}$

from various sensors. Specifically, we introduce a set of learnable sensor tokens $\{s_k\}_{k=1}^{K} \cup s_u$, where K is the number of sensor types, $s_k \in \mathbb{R}^{L \times d}$ are the sensor-specific tokens for the k-th sensor, $s_u \in \mathbb{R}^{L \times d}$ are universal sensor tokens and L is the number of sensor tokens for each sensor. When inputting the tactile token sequence z from the k-th sensor into the encoder Φ_{enc} to obtain its representation x_T, we randomly select one from s_k and s_u to concatenate with z, as follows:

$$\begin{aligned} s = i \cdot s_u + (1 - i) \cdot s_k,\ i \sim B(p_u), \\ x_T = \Phi_{enc}(z, s), \end{aligned} \tag{7}$$

where p_u is the probability of using universal sensor tokens s_u. During inference, we consistently use universal sensor tokens for data from new sensors. We transfer all available sensor information through these universal sensor tokens to aid in understanding new sensors.

F. Training Paradigm

Our framework has a multi-level structure, with the training of two stages conducted sequentially. In the first stage, we simultaneously perform the reconstruction of static tactile images and dynamic tactile videos, as well as the unique next frame prediction task for tactile videos. The loss for the first stage $\mathcal{L}_{stage1}$ is as follows:

$$\mathcal{L}_{stage1} = \mathcal{L}_{rec}^{S} + \mathcal{L}_{rec}^{D} + \mathcal{L}_{pred}^{D}. \tag{8}$$

In the second stage, we continue to use both tactile images and videos, and simultaneously perform multi-modal aligning and cross-sensor matching tasks. Hence, the loss function for the second stage is the sum of the losses from these two tasks:

$$\mathcal{L}_{stage2} = \mathcal{L}_{align} + \lambda \mathcal{L}_{match}, \tag{9}$$

where λ is a hyper-parameter controlling the weight of cross-sensor matching task. From both static and dynamic perspectives, we employ this multi-level framework to comprehensively learn unified multi-sensor representations for tasks requiring fine-grained perception and semantic understanding.

IV. Experiments

This section evaluates and analyzes our method TouchGo, with various rewards and states on zero-shot (*unseen*) objects. In addition, we validate our method on over 400 quantitative and qualitative experiments in reconstructing unknown objects with varying complexities. In our experiments, *we use reconstruction accuracy as the metric for tactile exploration with a limited number of steps as it represents the TouchGo exploration potential.*

A. Experimental Setup

Simulation. We employ TACTO [10], [3] to simulate tactile sensor skin deformation during object interactions and modified PPO from StableBaselines3 [7] in TouchGo.

The TACTO simulator is calibrated with real-sensor data to ensure Sim-to-Real generalization. It generates depth map images from real-world signals, serving as our observation O. We train the agent only with primitive objects – sphere and cube – for 300K steps. These primitive objects are selected as they represent a broad range of object shapes, with the sphere having curvature and the cube having sharp edges, corners, and flat surfaces. To assess the model's performance, we evaluate it on YCB objects that were not encountered during training time. This evaluation demonstrates the efficacy of training with primitives, which exhibit strong generalization capabilities for objects with realistic textures (Fig. S2$^{\textbf{??}}$). For the termination condition, each episode either spans 5000 steps (Sec. SI-A$^{\textbf{??}}$) or concludes once the Intersection over Union (IoU) metric exceeds 90%, or when the sensor leaves the workspace boundaries. This strategy is adopted to reduce the training time. In Tab. IV, we show that these termination conditions do not limit the IoU performance during testing as our methods achieved over 90%.

Real-World System. We employ a UR10 arm to manipulate the 6D pose of the DIGIT (Fig. 7). This control is achieved by transforming changes in the DIGIT's frame into a set of joint trajectories via inverse kinematics which are facilitated with ur_rtde. The resulting trajectories are executed only if free from self-collisions and within the defined workspace. When an invalid trajectory is generated, we select an alternative action based on the PPO's advantage values.

Unlike simulations, where consecutive action executions while in contact with the object have minimal impact, our real-world implementation introduces significant shearing on the sensing surface. To ensure the safe execution of actions generated by our policy, we have adopted a strategy of lifting the DIGIT in the normal direction of the contact after each contact event. This strategy remains well-founded due to our policy's consistent alignment of our sensor surface with the object's surface and does not compromise its general applicability. Our method successfully transferred to real-world experiments without requiring further fine-tuning. Fig. 7 illustrates the effectiveness of our exploration policy on a drill in the real-world.

Baselines Configuration. To evaluate the efficacy of each component, we have established a collection of baselines for three different state rep. and reward functions in Tab. III.

TABLE III: Baseline Formulations. **TTA**: Temporal Tactile Averaging, **TTS**: Temporal Tactile Stacking (concatenation is denoted as $\|$), **TM**: binary Touch indicator ($\mathbb{I}(\cdot)$) + short Memory, **AM**: contact Area + short Memory, **AMB**: contact Area + short Memory + UCB Bonus

State	Depth	TTA	TTS
	O_t	$\sum_{i=0}^{k-1} \alpha_i O_{t-i}$	$O_t \parallel \ldots \parallel O_{t-(k-1)}$
Reward	TM	AM	AMB
	$\mathbb{I}(O_t)$	$r_A(O_t)$	$\alpha r_A(O_t) + \frac{\beta}{\sqrt{\hat{N}(\mathcal{P}_t, a_t)}}$

B. Analysis & Discussion

State Comparison. We compare different state representations using the same reward function (AMB), considering both representations with and without temporal information. This analysis highlights the influential impact of temporal information on learning dexterous and high-level actions. As shown in Fig. **??**, all state representations achieve a specified IoU during training. However, the state representations incorporating temporal information demonstrate higher stability, consistently reaching the 90% IoU objective after 200K steps. In contrast, the depth-only representation struggles to maintain the IoU objective and is outperformed by temporal representations in Tab. IV. Furthermore, when considering the number of steps required to achieve the IoU objective, TTS training takes longer than TTA as $s_t^{TTS} \in R^{k\times H\times W}$ is k times bigger than $s_t^{TTA} \in R^{H\times W}$ which is averaging observations rather than stacking them. However, in our experiments in Fig. 6, we witnessed that both TTA and TTS are competitive, with TTS excelling on longer objects and TTA performing better on complex shapes.

Reward Comparison. In our pursuit of efficient explo-

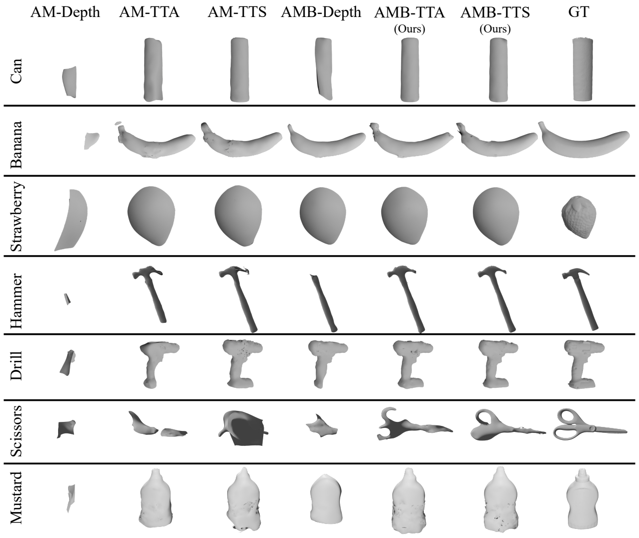


Fig. 6: **Qualitative results on unseen YCB objects** with different state and reward settings. We obtain point cloud data from active tactile exploration on the object's surface. To generate a mesh from the collected point cloud, we apply Poisson surface reconstruction algorithm [4]. Further experiments are provided in supplementary materials.

TABLE IV: **Quantitative results on unseen YCB objects:** The table presents IoU and Chamfer-L1 distance (cm) [6] between ground-truth and predicted meshes from methods in Tab. III. The surface area is listed below each object's name as a severity metric. The details of metrics, confidence intervals, and step counts are given in the supplementary material??.

Methods		**Can** (616 cm^2)	**Banana** (216 cm^2)	**Strawberry** (68 cm^2)	**Hammer** (410 cm^2)	**Drill** (591 cm^2)	**Scissors** (165.48 cm^2)	**Mustard** (454.54 cm^2)
				IoU ↑	(Chamfer-L_1 ↓)			
TM	depth	31.93 (2.66)	11.11 (7.52)	83.60 (0.44)	32.78 (1.86)	19.19 (4.1)	24.29 (8.15)	10.07 (4.07)
	TTA	17.60 (3.57)	6.03 (9.03)	41.0 (1.23)	14.85 (6.94)	28.15 (3.99)	14.17 (4.98)	19.94 (3.22)
	TTS	15.93 (5.22)	18.23 (5.48)	57.89 (0.88)	28.66 (2.47)	15.5 (3.97)	11.26(4.97)	14.55(2.95)
AM	depth	11.59 (5.49)	10.22 (6.84)	47.33 (1.16)	5.07 (7.69)	9.49 (4.03)	5.11(6.78)	11.04(5.16)
	TTA	72.70 (0.56)	97.70 (0.35)	**100** (0.28)	79.80 (0.82)	57.58 (1.43)	41.77 (2.87)	71.72 (0.80)
	TTS	**98.25** (0.22)	**100** (0.34)	**100** (0.31)	88.22 (0.44)	99.02 (0.37)	28.37 (2.38)	87.13 (0.59)
AMB	depth	41.45 (1.42)	98.64 (**0.25**)	**100 (0.23)**	61.42 (1.17)	79.68 (0.95)	31.99 (3.2)	65.74 (0.9)
	depth+LSTM	88.54 (0.3)	99.96 (0.28)	**100** (0.24)	87.54 (0.49)	92.81 (0.36)	29.83 (0.58)	88.33 (0.36)
	TTA (ours)	89.6 (0.29)	**100** (0.33)	**100** (0.25)	**98.22** (0.29)	98.85 (0.32)	67.02 (0.87)	**95.91** (0.51)
	TTS (ours)	97.45 (**0.20**)	**100** (0.3)	**100** (0.25)	96.96 (**0.28**)	**99.74 (0.31)**	**74.62 (0.61)**	95.02 (**0.49**)

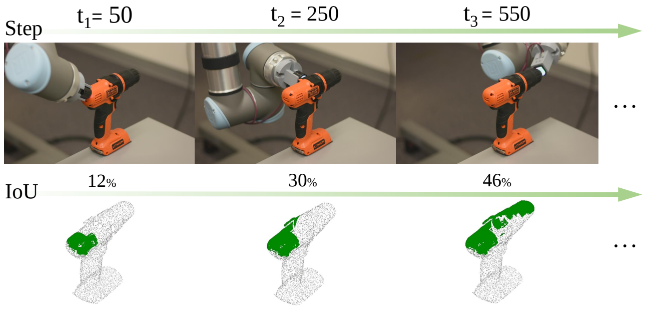


Fig. 7: **Real-World Exploration Execution.** Still frames from TouchGo's exploration of a drill, starting from the rear and progressing towards the chuck. The second row shows the covered area per step, with IoU computed over the exploration workspace above the drill's grip. t_i is the i-th step of the trajectory.

ration, we tried various reward functions mentioned in Tab. III. During training, we plotted the IoU and episode length until termination in Fig. **??**. Notably, the AMB reward function outperformed the others, satisfying the IoU objective through encouraging exploration of less visited poses. In contrast, TM and AM cannot use environmental feedback as much as AMB can. This limitation arises from TM and AM's deprivation of long-horizon history, which hampers their capacity to gather sufficient information through intrinsic rewards. As a result, AMB is better equipped to leverage environment feedback $\left(\frac{1}{\hat{N}(\mathcal{P},a)}\right)$ effectively for improved exploration and sample efficiency. However, AM outperforms TM as it utilizes contact area information and can still align the sensor's sensing area with the object surface to collect more information and maintain a reliable touch for future actions. Indeed, the disparity between TM and AM can also be understood as the distinction between using a touch sensor versus a tactile sensor for exploring an object.

Limitations and Future Work. The current formulation of our method has certain limitations. First, it assumes a moving sensor relative to a **fixed-pose** rigid object, necessitating a physically accurate simulator to narrow the sim2real gap for moving objects. Second, Although TouchGo is not restricted by object shape, it is designed to keep the sensor close to recent touching poses. This could pose challenges in environments with disconnected components. Workspace splitting can be a potential solution to address this problem. Third, the sensor exhibits a small depth bias in the simulation resulting in larger reconstructions. While generally negligible, this bias becomes dominant when handling objects roughly the same size as the sensor, such as the strawberry shown in Fig. 6.

As a step towards benchmarking in tactile exploration, we have released our extensive explorations for YCB objects in Tab. S1?? with a maximum of 5000 steps. While employing tactile sensors on multi-finger robotic hands may streamline the exploration process [9], there remains a promising direction for future research in modifying the POMDP that effectively handles collisions between sensors while maintaining object generalization.

V. Conclusion

In this work, we introduced a novel reinforcement learning method using tactile sensing to explore unknown 3D objects actively. It addresses the need for an active exploration method to enable numerous works [2], [5], [8], [1] to become fully automated. TouchGo is not limited to specific shape distributions as it has only been trained on primitive shapes to learn fundamental movements by leveraging temporal tactile information and intrinsic exploration bonuses. We demonstrated this through our experiments with various shape complexities like a drill or a clay pot in both the real world and simulation.

References

[1] Mauro Comi, Yijiong Lin, Alex Church, Alessio Tonioni, Laurence Aitchison, and Nathan F. Lepora. Touchsdf: A deepsdf approach for 3d shape reconstruction using vision-based tactile sensing. *ArXiv*, abs/2311.12602, 2023.

[2] Cristiana de Farias, Naresh Marturi, Rustam Stolkin, and Yasemin Bekiroglu. Simultaneous tactile exploration and grasp refinement for unknown objects. *CoRR*, abs/2103.00655, 2021.

[3] Benjamin Ellenberger. Pybullet gymperium. https://github.com/benelot/pybullet-gym, 2018–2019.

[4] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0, 2006.

[5] Wenyu Liang, Qinyuan Ren, Xiaoqiao Chen, Junli Gao, and Yan Wu. Dexterous manoeuvre through touch in a cluttered scene. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6308–6314, 2021.

[6] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[7] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[8] S. Suresh, Z. Si, J. Mangelson, W. Yuan, and M. Kaess. ShapeMap 3-D: Efficient shape mapping through dense touch and vision. In *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA*, Philadelphia, PA, USA, May 2022.

[9] Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, Joseph Ortiz, and Mustafa Mukadam. Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation. In *arXiv preprint arXiv:2312.1346*, December 2023.

[10] 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. *CoRR*, abs/2012.08456, 2020.