Tactile MNIST: Benchmarking Active Tactile Perception

Tim Schneider[1,2], **Guillaume Duret**[2], **Cristiana de Farias**[1], **Roberto Calandra**[3], **Liming Chen**[2], **and Jan Peters**[4]

[1]Department of Computer Science, TU Darmstadt, Germany.
[2]LIRIS, CNRS UMR5205, Ecole Centrale de Lyon, France.
[3]LASR Lab & CeTI, TU Dresden, Germany.
[4]DFKI, Hessian.AI, and Centre for Cognitive Science, TU Darmstadt, Germany.

Abstract

Tactile perception has the potential to significantly enhance dexterous robotic manipulation by providing rich local information that can complement or substitute for other sensory modalities such as vision. However, because tactile sensing is inherently local, it is not well-suited for tasks that require broad spatial awareness or global scene understanding on its own. A human-inspired strategy to address this issue is to consider active perception techniques instead. That is, to actively guide sensors toward regions with more informative or significant features and integrate such information over time in order to understand a scene or complete a task. Both active perception and different methods for tactile sensing have received significant attention recently. Yet, despite advancements, both fields lack standardized benchmarks. To bridge this gap, we introduce the *Tactile MNIST Benchmark Suite*, an open-source, Gymnasium-compatible benchmark specifically designed for active tactile perception tasks, including localization, classification, and volume estimation. Our benchmark suite offers diverse simulation scenarios, from simple toy environments all the way to complex tactile perception tasks using vision-based tactile sensors. Furthermore, we also offer a comprehensive dataset comprising 13,500 synthetic 3D MNIST digit models and 153,600 real-world tactile samples collected from 600 3D printed digits. Using this dataset, we train a CycleGAN for realistic tactile simulation rendering. By providing standardized protocols and reproducible evaluation frameworks, our benchmark suite facilitates systematic progress in the fields of tactile sensing and active perception.
Project page: https://sites.google.com/robot-learning.de/tactile-mnist

1 Introduction

Tactile perception is fundamental for enabling agents to interact effectively with their environments. Studies of humans with impaired touch reveal that they face significant challenges in grasping and performing routine manipulation tasks due to insufficient feedback about contact states between fingers and objects [4]. Moreover, touch often complements—or even substitutes—other sensory modalities such as vision: we feel the shape of a hard-to-see object on a high shelf, count cookies in a jar without looking, or locate a key inside a bag purely by touch. Unlike vision, which typically offers a broad field of view, touch provides highly localized yet information-rich feedback confined to the point of contact [5, 6]. It is this inherently interactive nature that enables agents (such as robots) to explore visually occluded areas, classify textures, infer material properties like stiffness

Corresponding Author: tim@robot-learning.de

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

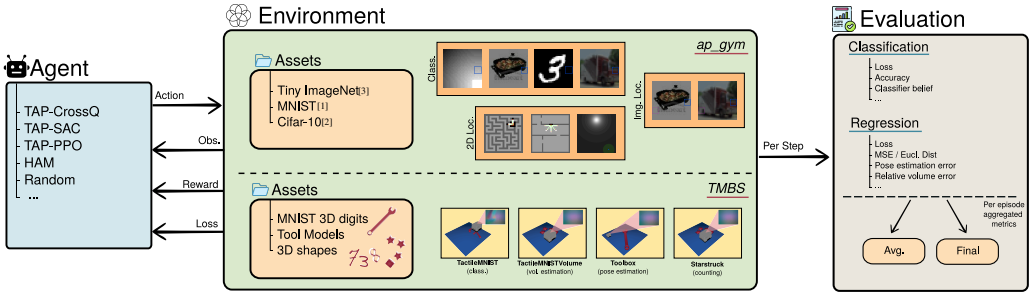

Figure 1: Overview of the *Active Perception Gym* (`ap_gym`), the *Tactile MNIST Benchmark Suite* (TMBS), and their associated assets. In the center we depict each environment from `ap_gym` and TMBS, along with the included asset sets (both custom and external). On the left, an agent (e.g., an active perception algorithm) interacts with the environment by receiving observations, rewards, and losses, and returning actions. On the right, we show task-specific evaluation metrics, available at each step, with support for both per-step outputs and aggregated performance scores.

and friction, and detect fine local features of objects [7, 8, 9, 10, 11, 12, 13, 14]. However, without standardized benchmarks and widely adopted datasets, it becomes difficult to rigorously evaluate new algorithms, reproduce results, or compare different approaches in a fair way. Yet, despite its growing importance, the field of tactile sensing still lacks such structured benchmarks and community-wide datasets tailored to touch-related tasks

In contrast to tactile sensing, computer vision has significantly benefited from such standardized datasets and clearly defined evaluation protocols. MNIST [1] established an early baseline for digit recognition that is still relevant today. Datasets such as ImageNet [15] and COCO [16] expanded both scale and complexity, driving major advances in object classification, detection, and scene understanding. Additionally, domain-specific benchmarks such as KITTI [17] and Omni3D [18] have enabled focused progress on challenges like fine-grained categorization and robustness in real-world applications. By comparison, the few tactile perception benchmarks in the literature still rely on custom hardware or narrowly scoped datasets, which limits their generalizability and slows broader adoption [14, 19].

In this work, we focus explicitly on benchmarking *active tactile perception* tasks where agents deliberately interact with their environment to gather task-relevant information. Active perception involves strategic decisions about where and when to sense, efficiently choosing actions that maximize information gain [10, 9, 20, 8]. Here, as each contact takes time, resources, and may cause wear on sensors or environments, efficiency becomes a central concern: agents must extract as much information as possible using as few interactions as necessary. A well-designed benchmark for active tactile perception can help answer key questions such as: How should an agent select contact points to maximize information gain? What policies enable accurate inference with minimal touch? How does uncertainty (from sensor noise, ambiguous contact, or environmental variability) affect the trade-off between exploration and confidence?

We introduce *Active Perception Gym* (`ap_gym`)[1], a framework compatible with Gymnasium [21], designed to benchmark active perception algorithms. `ap_gym` includes nine toy scenarios where an agent must learn to efficiently extract information to solve perception tasks. Building on `ap_gym`, we present the *Tactile MNIST Benchmark Suite* (TMBS)[2], which extends the framework to simulated active tactile perception problems. In TMBS, agents control a simulated GelSight Mini sensor [22] without access to visual inputs. Tactile perception tasks include MNIST-style classification of 3D digit models, pose estimation of tools on platforms, and object counting. Across all tasks, the agent must actively determine *what* to sense and strategically select *where and when* to explore through touch. Thus, solving these tasks requires solving a dual problem: making accurate predictions from past observations while optimizing an exploration policy to maximize information gain. An overall visualization of our framework and tasks in `ap_gym` and TMBS is shown in Fig. 1.

[1] `https://github.com/TimSchneider42/active-perception-gym`

[2] `https://github.com/TimSchneider42/tactile-mnist`

Table 1: Comparison of Active Tactile Perception Benchmarks. Tactile modalities are **bolded**.

Method	Dataset Available	Active Benchmark	Sensor Modality
Active Vision Grasp [24]	✗	✓	Vis
R3ED [25]	✓	✓	Vis
Robotic Vision Challenge [26]	✗	✓	Vis
NBV_Bench [27]	✓	✓	Vis
AVD [28, 29]	✓	✓	Vis
Active Object Search [30]	✓	✓	Vis
ActiView [31]	✓	✓	Vis+Text
TIP Bench. [32]	✗	✗	**Tac**
FoTa [33]	✓	✗	**Tac**
YCB-Slide [34]	✓	✗	**Tac**
TacBench [14]	✓	✗	**Tac**
ActiveCloth [35]	✓	✗	**Tac**
Touch and Go [36]	✓	✗	**Tac**+Vis
FeelSight [19]	✓	✗	**Tac**+Vis
ViTac [37]	✗	✗	**Tac**+Vis
SSVTP [13]	✓	✗	**Tac**+Vis
GelFabric [38]	✓	✗	**Tac**+Vis
VisGel [39]	✓	✗	**Tac**+Vis
PHYSICLEAR [40]	✓	✗	**Tac**+Text
TVL [41]	✓	✗	**Tac**+Vis+Text
Touch100k [42]	✓	✗	**Tac**+Vis+Text
ObjectFolder [43, 44, 45]	✓	✗	**Tac**+Vis+Audio
Ours	✓	✓	**Tac**

With this benchmark, our goal is to provide a reproducible and accessible evaluation framework for the tactile perception community. To that end, TMBS is a fully *simulated* environment that is easy to set up and enables rapid, consistent comparisons, without the need to replicate a complex real-world setup. However, we acknowledge the inherent challenges and noise present in real-world tactile data that are often absent in simulation. To help bridge this sim-to-real gap, we complement our simulated environment with a large-scale dataset of 13,580 high-fidelity 3D object models. From this collection, we 3D-printed 600 objects and constructed a curated real-world dataset comprising 153,600 tactile contacts, each annotated with detailed temporal and spatial metadata. We further leverage this dataset to train a CycleGAN [23], enabling the rendering of realistic tactile signals within the simulation.

In summary, our main contributions are:

- To the best of our knowledge, we introduce the first benchmark suite specifically for active *tactile* perception. It offers a range of tasks from simple toy problems to challenging, high-dimensional scenarios.
- We further introduce `ap_gym`, an extensible framework for generic active perception algorithms. `ap_gym` is Gymnasium compatible and is, thus, easy to integrate with existing reinforcement-learning pipelines, enabling fair evaluation.
- We provide an open dataset of 13,580 high-resolution 3D models of handwritten digits, designed for both simulated tactile-image generation and physical 3D printing.
- From our library of 3D models, we 3D-printed 600 objects and captured 153,600 tactile contacts using a GelSight Mini sensor, each annotated with spatial location and class labels.

2 Related Work

(Active) Tactile Perception: Tactile sensing enables robots to infer object geometry, texture and material properties through physical contact, complementing or, in some cases, substituting vision. Vision-based tactile sensors such as GelSight [22] and DIGIT [46] have become widely available, producing high-resolution "tactile images" that capture local surface features and force distributions. These rich signals have been exploited for shape reconstruction and material recognition [7, 32, 41, 40], as well as advanced dexterous manipulation [47, 48, 19].

Much of the work in tactile sensing, however, focuses on passive touch: the robot either registers a single contact or follows a predefined exploration policy. Inspired by the successes of active vision (as well as by early research on active touch in robotics [49, 50]), recent efforts have revisited the idea of tactile exploration as an active, decision-driven process. In this framing, the robot dynamically selects where and how to touch next, rather than relying on a fixed sequence of actions. Gaussian

process and Bayesian optimization have been employed to drive this active exploration, yielding significant improvements in tasks such as shape reconstruction, texture classification and grasp planning [10, 7, 51]. Reinforcement-learning based approaches [9] tackle active exploration in high-dimensional tactile state spaces. Furthermore, [8] introduced HAM a selective-attention mechanism to optimize scene exploration, and in [52] task-agnostic strategies generalize active touch across different objectives. Additionally, [35] demonstrated how Kinect-based vision can guide active touch for material classification, and [13] proposed a self-supervised visuo-tactile pretraining scheme that benefits both passive and active perception tasks.

Benchmarking Methods for Tactile Sensing & Active Perception: Over the past decade, the active vision community has produced a number of datasets and challenges to evaluate a range of tasks. Early work such as the Active Vision Dataset (AVD) provided large-scale Kinect captures for navigation and class-incremental learning tasks [28, 29]. Subsequent efforts explored next-best-view planning for classification (NBV_Bench [27]), heuristic and data-driven view selection for grasp synthesis on YCB objects [24], and embodied 3D exploration in real indoor scenes (R3ED [25]). Simulation-based approaches such as the Robotic Vision Scene Understanding Challenge [26] and Active Object Search [30] have further expanded evaluation protocols for semantic SLAM and object detection tasks. More recently, multi-modal active perception has been addressed by ActiView, which tests an agent's ability to zoom and pan to answer vision-language queries [31].

In parallel, tactile sensing research has released a diverse set of characterization benchmarks (TIP Bench. [32]), texture and material recognition datasets (ActiveCloth [35], ViTac [37], SSVTP [13], GelFabric [38]), and cross-modal vision–touch benchmaks and datasets (VisGel [39], Touch and Go [36], PHYSICLEAR [40], FeelSight [19], TacBench [14]). Large-scale multimodal datasets such as Touch100k [42], TVL [41], and FoTa [33] now exceed tens of thousands of samples across vision, touch, and language modalities. Multisensory datasets like ObjectFolder [43, 44, 53] further integrate tactile, visual, and audio data. In a similar manner, efforts to standardize vision-based tactile simulation have led to platforms like TACTO [54] and Taxim [55], which enable high-resolution visuo-tactile data generation. Despite these efforts, only the datasets provided by ActiveCloth [35] and SSVTP [13] support downstream active tactile perception tasks. However, these works do not provide standardized evaluation protocols or benchmarks to systematically evaluate active perception approaches. A comparison of existing datasets and benchmarks is summarized in Table 1. To the best of our knowledge, our proposed Tactile MNIST Benchmark Suite is the first to introduce a dedicated and reproducible benchmark for *active tactile perception*, where tactile exploration is an integral component of the perceptual process.

Table 2: Overview of the environments, including task types, descriptions, and assets.

Suite	Environment	Task Type	Description	Assets
`ap_gym`	TinyImageNet	Classification	Classify natural images into 200 categories by moving a limited field-of-view glimpse.	Tiny ImageNet [3]
	CIFAR10	Classification	Classify natural images into 10 categories by moving a limited field-of-view glimpse.	CIFAR-10 [2]
	CircleSquare	Classification	Determine whether a given image contains a circle or a square using limited agent visibility.	Geometric shapes
	MNIST	Classification	Digit recognition task using standard MNIST digits.	MNIST [1]
	LightDark	Regression (2D localization)	Position estimation from brightness-dependent noisy observations, requiring movement to light.	None
	LIDARLocRooms	Regression (2D localization)	Navigate procedurally generated maps with ambiguous LIDAR readings to localize.	None
	LIDARLocMaze	Regression (2D localization)	Navigate procedurally generated mazes with ambiguous LIDAR readings to localize.	None
	TinyImageNetLoc	Regression (2D patch localization)	Localize a glimpse within a natural image by moving a limited field-of-view.	Tiny ImageNet [3]
	CIFAR10Loc	Regression (2D patch localization)	Localize a glimpse within a natural image by moving a limited field-of-view.	CIFAR-10 [2]
TMBS	TactileMNIST	Classification	Touch-based digit classification using a vision-based tactile sensor.	MNIST 3D digits
	TactileMNISTVolume	Regression (volume estimation)	Estimate volume of digits using a vision-based tactile sensor.	MNIST 3D digits
	Toolbox	Regression (object pose estimation)	Estimate 6D pose of tools (e.g., a wrench) using a vision-based tactile sensor.	3D tools
	Starstruck	Classification (counting)	Count stars among other objects using a vision-based tactile sensor.	3D shapes

3 Framework: Benchmarking Active Perception

In this section, we introduce both the *Active Perception Gym* (`ap_gym`), a framework for benchmarking active perception algorithms, and the *Tactile MNIST Benchmark Suite* (TMBS), which introduces environments for four active tactile perception tasks. Fig. 1 depicts an overview of our framework.

3.1 Active Perception

In active perception tasks, an agent's main objective is to gather information and make predictions about a desired property of the environment, e.g., the class label or pose of an object. Examples of such properties could be the location of an object in case of a search task or the class of an object for the agent in case of a classification task. To gather information, the agent must interact with the environment, e.g., by moving a sensor around a platform in case of TMBS.

We model active perception as an episodic process, where the agent can take a number of time-discrete sequential actions until the episode terminates and is reset. At every step, the agent obtains an observation (e.g., a tactile glimpse) from the environment that reveals some information but never the full state at once. Formally, that makes active perception problems a special case of Partially Observable Markov Decision Processes (POMDPs).

POMDPs are defined by the tuple $(S, A, T, R, \Omega, O, \gamma)$, consisting hidden states S, actions A, a transition function $T : S \times A \times S \to [0, 1]$, a reward function $R : S \times A \to \mathbb{R}$, a set of observations Ω, an observation function $O : S \times A \times \Omega \to [0, 1]$, and a discount factor $\gamma \in [0, 1]$. The objective of the agent in a POMDP is to maximize the expected cumulative reward over time by selecting actions based on its belief about the underlying state. Since the agent does not have direct access to the true state, it maintains a belief distribution over states, updating it using observations and the observation function. The environment evolves according to the transition function, where taking an action leads to a probabilistic transition to a new state, which in turn generates an observation based on the observation function.

In case of active perception problems, we assume that the hidden state S, the action A, the reward function R, and the transition function T have specific structures. First, we assume that the target property the agent is tasked to predict is part of the hidden state. Hence, S is defined as $S = S_{\text{base}} \times Y^*$, where S_{base} is the set of base (hidden) states of the environment and Y^* is the set of prediction targets. E.g., Y^* could be the set of classes in a classification task or the set of possible locations in a localization task, while S_{base} contains all the other hidden state information. To allow the agent to make predictions, the action space A is defined as $A_{\text{base}} \times Y$, where A_{base} is the base action space and Y is the prediction space. The base action space A_{base} contains all the actions the agent can take to interact with the environment, while Y is the set of possible predictions the agent can make. Crucially, environments are defined in a way that the agent's prediction never influences the hidden state of the environment. Thus, the transition function T is defined as $T(s, a, s') = T(s, (a_{\text{base}}, y), s') = T_{\text{base}}(s, a_{\text{base}}, s')$. An example of a base action could be a desired movement of the tactile sensor, while the prediction could be the logits of the agent's current class prediction.

Finally, the reward function is defined as $R(s, a) = R((s_{\text{base}}, y^*), (a_{\text{base}}, y)) = R_{\text{base}}(s_{\text{base}}, a_{\text{base}}) - \ell(y^*, y)$, where R_{base} is the base reward function and ℓ is a differentiable loss function. An example

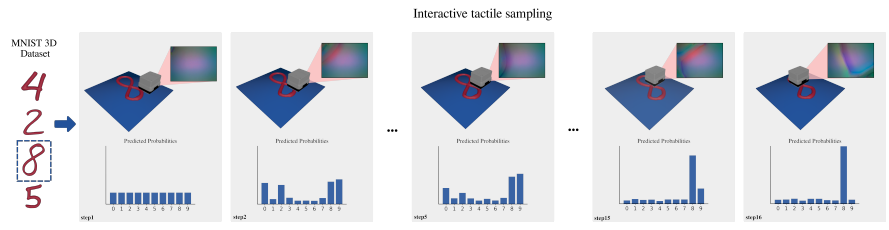

Figure 2: Illustration of the TactileMNIST classification task. In each episode of the TactileMNIST classification task, one random digit from the MNIST 3D dataset is selected and presented to the agent. It can then move the sensor around and touch the object 16 times before the episode terminates. Notably, it does not receive any visual input but has to rely solely on the readings from its tactile sensor. After every touch, it has to make a prediction about the class label, the digit's numeric value, and its performance is measured by the average prediction accuracy throughout the episode.

for a base reward could be an action regularization term, while the loss function ℓ could be a cross-entropy loss in a classification task. Hence, the agent has to make a prediction in every step, encouraging it to gather information quickly to maximize its prediction reward early on. A visualization of this process on the TactileMNIST digit classification task is shown in Fig. 2.

3.2 Active Perception Gym

`ap_gym` models active perception tasks as episodic processes in a way that is fully compatible with Gymnasium [21]. Each task is defined as a Gymnasium environment, bundled with the differentiable loss function $\ell(y^*, y)$ and the prediction target y^*. Since the loss functions need to convey gradient information to the learning algorithm, we currently provide them either JAX [56] or PyTorch [57] functions, but more autograd frameworks might be supported in the future. During roll-outs, `ap_gym` automatically computes task-dependent metrics, such as accuracy for classification, or Euclidean distance for regression.

`ap_gym` provides a family of lightweight environments designed to isolate core exploration and decision-making behaviors in active perception. As summarized in Table 2, `ap_gym` includes 11 environments spanning both classification and regression tasks. Four progressively harder image-based classification benchmarks (CircleSquare, MNIST, CIFAR-10, TinyImageNet) evaluate an agent's ability to select informative glimpses from natural or synthetic visuals. Two image-localization tasks (TinyImageNetLoc, CIFAR10Loc) require the agent to infer the position of a limited-field-of-view patch within a larger image. Finally, five non-visual regression tasks — LightDark and four LIDAR-based 2D localization environments (Rooms and Maze) — challenge agents to reduce state uncertainty by navigating procedurally generated maps or lighting fields. Crucially, in the self-localization environments, the agent influences the property it is trying to infer — its position — through its actions. Hence, the prediction target changes over time in these environments, which is an explicitly supported aspect of `ap_gym` environments.

For the image-based classification and localization tasks, `ap_gym` relies on a mix of third-party assets: Tiny ImageNet [3], CIFAR-10 [2], and MNIST [1] datasets. The LIDARLoc (rooms and maze) environments generate map layouts procedurally and require no external data. Whenever applicable, `ap_gym` defines two versions of each environment, one for training with the training split of the respective dataset, and one for evaluation with the test split.

By abstracting away complex contact models and dynamics, `ap_gym` environments enable rapid prototyping of active perception strategies and provide a controlled baseline for more complex scenarios in the TMBS suite. However, despite their simplistic appearance, all `ap_gym` environments impose significant challenges due to partial observability and non-immediate action payoffs. The tasks differ substantially from each other, testing the algorithm's capability to handle diverse scenarios.

3.3 The Tactile MNIST Benchmark Suite

Tactile sensing presents unique challenges for perception: as an interactive modality, touch can unintentionally shift objects during exploration, and modern vision-based tactile sensors, such as GelSight [22] and DIGIT [46], produce inherently high-dimensional observations. At the same time, tactile sensing provides highly localized information confined to points of contact, necessitating active perception.

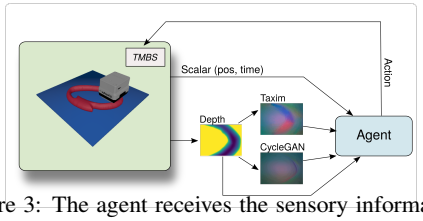

Figure 3: The agent receives the sensory information from the environment. This observation consists of scalar values and a tactile image. This can have three rendering modes, depth, Taxim, or CycleGAN.

The Tactile MNIST Benchmark Suite (TMBS) extends `ap_gym` to tactile tasks using vision-based tactile sensors. It contains four environments: *TactileMNIST*, where the agent must classify a 3D model of a hand-written digit (see Section 3.4), *TactileMNISTVolume*, which tasks the agent to infer the volume of a given digit, *Toolbox*, where the agent must estimate the pose of a tool, and *Starstruck*, in which the agent must count the number of stars among other shapes (see Table 2 for an overview). In each environment, the agent controls a simulated GelSight Mini tactile sensor [22] and is presented with one or more objects on a platform. By interacting with the object, the agent must infer task-dependent properties, such as the

class of the object, its pose, or its volume. Particularly, aside from tactile data and proprioception, the agent does not receive any additional sensory data, so it has to infer the required property from touch alone. Although, for simplicity and performance reasons, we do not simulate the physical interaction between the sensor and the object, we shift the objects around randomly to simulate unintended object movements.

To simulate the tactile sensor, we support three rendering modes:

Taxim: Taxim [55] computes an approximation of the gel deformation and afterwards applies a data-driven rendering algorithm.

CycleGAN: With data collected on 3D printed MNIST 3D objects (see Section 3.5), we train a CycleGAN [23] for a style transfer between a depth image and tactile image [58]. The resulting images are visually much more realistic than the Taxim renderings, and thus might be beneficial for sim-to-real transfer. However, this mode is currently only available for the *TactileMNIST* environment. For more details, refer to Appendix C.

Depth: Here, the agent receives a depth image clipped to the GelSight gel thickness (4.25mm).

A comparison of all rendering modes is shown in Fig. 3, a visualization of the TactileMNIST digit classification task is provided in Fig. 2, and an overview of all tasks in TMBS is given in Table 2.

3.4 The MNIST 3D Dataset

MNIST 3D[3] is a collection of 13,580 auto-generated 3d-printable meshes derived from a 500×500 - pixel high-resolution MNIST variant [59] and scaled to fit in a 10x10cm square. The MNIST 3D dataset poses an exciting tactile classification challenge, as it has significant variability in shape and size within the classes, while also being large enough to facilitate learning from data. A single touch is rarely enough to classify objects from this dataset, as segments of hand-written digits are usually ambiguous. Hence, even after finding the object, the agent has to apply some strategy (e.g., contour following) to gather enough information for a successful classification. In addition to tactile sensing, this dataset could also be used as a benchmark for 3D mesh classification methods. More details on the generation of the MNIST 3D dataset can be found in Appendix D.

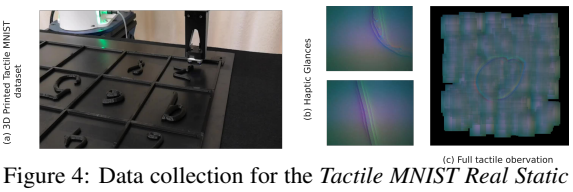

Figure 4: Data collection for the *Tactile MNIST Real Static* dataset. We mounted a GelSight Mini sensor on a Franka Research 3 (a) and collected 153,600 touches across 600 3D-printed MNIST digits. Examples of individual touches are visible in (b), and a collection of 256 touches overlayed based on their positions in (c).

3.5 A Large Dataset of Real Tactile Interactions

To complement simulated renders, TMBS includes a real–world, static tactile dataset captured with a GelSight sensor on 3D-printed MNIST 3D digits: the *Real Tactile MNIST Dataset*[4]. The dataset contains video sequences of 153,600 touches across 600 digits, which amounts to 256 touches per object collected in sequence. For data acquisition, we laid each 3D-printed MNIST digit in a 12x12cm grid on a rubber mat and used a Franka Research 3 robot arm [60], with a GelSight tactile sensor to press the sensor down at random locations in the cell. Once we measured a normal force exceeding 5N, we stopped pressing and registered the time stamp. To prevent degradation of the elastomer gel, we replaced the GelSight sensor's gel pad after every 76,800 touches (i.e., halfway through each dataset). Finally, we partitioned each dataset into training (90 %) and test (10 %) splits, ensuring uniform class distributions across each split. Note that we also provide two processed versions of this dataset, where we replaced the videos with still images at the time of contact, one in full resolution at 320x240px[5] and one scaled to 64x64px[6] for faster loading and training.

[3] https://huggingface.co/datasets/TimSchneider42/tactile-mnist-mnist3d

[4] https://huggingface.co/datasets/TimSchneider42/tactile-mnist-touch-real-seq-t256-320x240

[5] https://huggingface.co/datasets/TimSchneider42/tactile-mnist-touch-real-single-t256-320x240

[6] https://huggingface.co/datasets/TimSchneider42/tactile-mnist-touch-real-single-t256-64x64

We note that an additional challenge introduced during data collection is the possibility of the digit shifting slightly due to contact with the sensor. This variability makes the dataset more representative of real-world scenarios and provides an opportunity for methods to learn robustness to object movement and misalignment. Thus, the Real Tactile MNIST Dataset serves several key roles in the TMBS benchmark. First, it enables training of a CycleGAN for realistic simulation of tactile images and sim-to-real transfer. Second, it can be used for both pretraining and fine-tuning of learning-based perception models, enabling models to acquire basic tactile features before being deployed on a robot for active learning tasks and again facilitating sim-to-real transfer. Finally, the dataset provides a reproducible, offline benchmark for validating and comparing active perception algorithms under realistic sensor noise and material artifacts. For additional details on the data collection procedure, refer to Appendix E.

3.6 Evaluation Protocols

In `ap_gym` environments, there are two levels of exploration: (1) during an episode, the agent must explore to gather information, and (2), over the course of the training, the agent must explore the effects of its actions to optimize its model and policy. To disambiguate the measures of performance in these two levels, we will call the first one *exploration efficiency* and the second *sample efficiency*. Importantly, the former is a quality measure of the policy, while the latter is a quality measure of the learning algorithm. Here we borrow the term *sample efficiency* from the RL literature, where it refers to the number of environment interactions the agent needs in order to learn to solve the given task. By *exploration efficiency*, on the other hand, we refer to the efficiency with which the agent collects information within an episode.

Regarding *exploration efficiency*, we consider two measures: the *average* prediction accuracy and the *final* prediction accuracy. Here, *average* prediction accuracy means the prediction accuracy the agent exhibited throughout an episode on average, while *final* prediction accuracy means the accuracy the agent exhibited at the final step of the episode. Normally, the agent starts each episode with little to no information and then keeps gathering information as the episode progresses. Hence, for a rational agent, we expect the prediction accuracy to increase over the course of the episode and to be highest at the end of the episode. Thus, the *average* prediction accuracy could be seen as a measure of how quickly the agent explored, while the *final* prediction accuracy could be seen as a measure of how thoroughly the agent explored throughout the episode. In `ap_gym` environments, both of these measures are tracked for a number of environment-specific prediction accuracy metrics, such as classification accuracy, mean-squared-error, pose error, and others. For a detailed list of metrics for each environment, refer to Appendix F.

Approaches evaluating on `ap_gym` or the Tactile MNIST benchmark suite should report both *average* and *final* metric values over the course of the training. If applicable, the metrics should be computed on the test variants of the environments, which use the test split instead of the training split. The objective is to maximize both *sample efficiency* and *exploration efficiency*. Section 4 serves as an example for an evaluation report on `ap_gym` and Tactile MNIST environments.

4 Experiments

In this section, we highlight experiments across selected environments from `ap_gym` and TMBS for various baseline methods, including `TAP` [52] and `HAM` [8]. Both `TAP` and `HAM` are RL-based active perception methods and, thus, well suited for evaluation on TMBS. The main difference between them is that `TAP` employs an actor-critic RL approach in combination with a transformer architecture, while `HAM` relies on a REINFORCE gradient in combination with an LSTM model. `TAP` provides two variants: `TAP-SAC` and `TAP-CrossQ`, based on `SAC` [61] and `CrossQ` [62]. We additionally evaluate a baseline that uses `TAP`'s transformer model with PPO [63], which we call `TAP-PPO`.

We highlight experiments on four environments in total. In the CircleSquare environment, the agent can move a glimpse around an image and has to find and classify an object that can be either a circle or a square. The TactileMNIST environment tasks the agent to classify MNIST 3D models by touch alone, and in the Starstruck environment, the agent must count the number of stars (1-3) among other objects on the platform. In the Toolbox environment, the agent must find a tool and determine its 2D pose and orientation on the platform. These environments represent a diverse set of challenges, and solving them requires the agent to adopt efficient exploration strategies, which is made evident by the

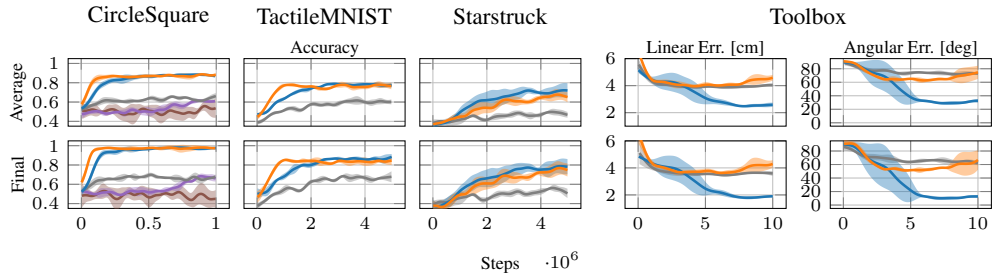

Figure 5: Average and final prediction accuracies for the baseline methods `TAP-SAC`, `TAP-CrossQ`, `TAP-PPO`, `HAM` [8], and a random baseline `TAP-RND` for the CircleSquare (`ap_gym`), TactileMNIST (TMBS), Starstruck (TMBS), and Toolbox (TMBS) environments. All methods were trained on 5 seeds for up to 10M. Shaded areas represent one standard deviation. Metrics are computed on evaluation tasks with unseen objects, except for Circle-Square and Toolbox, which have only two and one, respectively. For Starstruck, a correct classification requires predicting the exact number of stars. For Toolbox, we compute the linear and angular displacement between the prediction and the actual object pose as a metric. As `HAM` does not have a vision encoder, we evaluate it on non-tactile environments only. For `TAP-PPO`, we found it to be unstable in combination with a vision encoder, so we resort to only evaluating it on non-tactile environments as well.

consistently poor performance of `TAP`'s random baseline `TAP-RND` in Fig. 5. Further experiments and details for the training can be found in Appendix G.

As visible in Fig. 5 and Appendix G, `TAP`'s off-policy methods perform consistently better than the on-policy baselines `HAM` and `TAP-PPO`. This gap is likely due to on-policy methods generally being more sample-efficient than off-policy methods, as on-policy methods cannot reuse previously collected samples. However, despite the better performance of `TAP`, neither the `ap_gym` tasks nor the TMBS can be considered solved. `TAP` requires millions of environment interactions to learn viable exploration policies and falls short of perfect accuracy. More research in the area of sample-efficient RL and active perception is needed to improve sample efficiency to allow for the deployment of such methods in the real world.

5 Limitations and Conclusion

In this paper, we have introduced *Active Perception Gym* (`ap_gym`), a Gymnasium-compatible benchmark suite tailored for evaluating active perception tasks, and the *Tactile MNIST Benchmark Suite* (TMBS), which contains four tactile-specific tasks designed for robust exploration. To support these benchmarks, we have released a dataset comprising 13,580 high-resolution 3D digit models and an extensive real-world dataset of 153,600 tactile samples collected from 600 3D-printed digits using a GelSight Mini sensor. Provided as an open-source framework, `ap_gym` and TMBS offer a structured, standardized, and reproducible benchmark intended to facilitate advancements in active perception research, including efficient exploration strategies, sim-to-real adaptation through CycleGAN training, and the pretraining and fine-tuning of tactile models.

However, our benchmark has limitations, most notably the absence of online, real-world evaluation scenarios and metrics beyond the static dataset. While there are established datasets such as YCB [64] enable benchmarking of contact-rich manipulation tasks using real-world objects, our suite deliberately focuses on controlled, simulation-based exploration to ensure reproducibility and interoperability. In future work, we aim to address this limitation by designing carefully structured real-world tactile exploration experiments that extend the benchmark's relevance to physical robotic systems and support the study of sim-to-real transfer in active perception. Another key limitation is the absence of a physics engine in our simulation environment, which currently prevents modeling of more complex, contact-rich interactions. As a result, tasks such as grasping, 3D object reconstruction (where objects may tumble upon contact), object retrieval in cluttered scenes, and in-hand pose estimation remain out of scope. Extending our framework to incorporate physics engines would open the door to these richer interaction scenarios and significantly broaden the benchmark's applicability. Ultimately, we view this benchmark as a foundational resource for advancing active tactile perception and enabling the development of more robust, efficient algorithms for robotic tactile manipulation.

References

[1] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[2] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[3] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[4] 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.

[5] Susan J Lederman and Roberta L Klatzky. Hand movements: A window into haptic object recognition. *Cognitive psychology*, 19(3):342–368, 1987.

[6] Tony J. Prescott, Mathew E. Diamond, and Alan M. Wing. Active touch sensing. *Phil. Trans. R. Soc. B*, 366(1581):2989–2995, Nov 2011.

[7] A. Boehm, T. Schneider, B. Belousov, A. Kshirsagar, L. Lin, K. Doerschner, K. Drewing, C.A. Rothkopf, and J. Peters. What matters for active texture recognition with vision-based tactile sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[8] Sascha Fleer, Alexandra Moringen, Roberta L Klatzky, and Helge Ritter. Learning efficient haptic shape exploration with a rigid tactile sensor array. *PloS one*, 15(1):e0226880, 2020.

[9] Jingxi Xu, Shuran Song, and Matei Ciocarlie. Tandem: Learning joint exploration and decision making with tactile sensors. *IEEE Robotics and Automation Letters*, 7(4):10391–10398, 2022.

[10] Zhengkun Yi, Roberto Calandra, Filipe Veiga, Herke van Hoof, Tucker Hermans, Yilei Zhang, and Jan Peters. Active tactile object exploration with gaussian processes. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4925–4930. IEEE, 2016.

[11] Marten Björkman, Yasemin Bekiroglu, Virgile Högman, and Danica Kragic. Enhancing visual perception of shape through tactile glances. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3180–3186. IEEE, 2013.

[12] Niklas Funk, Erik Helmut, Georgia Chalvatzaki, Roberto Calandra, and Jan Peters. Evetac: An Event-Based Optical Tactile Sensor for Robotic Manipulation. *IEEE Trans. Rob.*, 40:3812–3832, July 2024.

[13] Justin Kerr, Huang Huang, Albert Wilcox, Ryan Hoque, Jeffrey Ichnowski, Roberto Calandra, and Ken Goldberg. Self-supervised visuo-tactile pretraining to locate and follow garment features. *arXiv preprint arXiv:2209.13042*, 2022.

[14] Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *8th Annual Conference on Robot Learning*, 2024.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[18] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13154–13164, 2023.

[19] 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.

[20] Amir-Hossein Shahidzadeh, Seong Jong Yoo, Pavan Mantripragada, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. Actexplore: Active tactile exploration on unknown objects. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3411–3418. IEEE, 2024.

[21] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

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

[23] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.

[24] Sabhari Natarajan, Galen Brown, and Berk Calli. Aiding grasp synthesis for novel objects using heuristic-based and data-driven active vision methods. *Frontiers in Robotics and AI*, 8:696587, 2021.

[25] Qianfan Zhao, Lu Zhang, Lingxi Wu, Hong Qiao, and Zhiyong Liu. A real 3d embodied dataset for robotic active visual learning. *IEEE Robotics and Automation Letters*, 7(3):6646–6652, 2022.

[26] David Hall, Ben Talbot, Suman Raj Bista, Haoyang Zhang, Rohan Smith, Feras Dayoub, and Niko Sünderhauf. The robotic vision scene understanding challenge. *arXiv preprint arXiv:2009.05246*, 2020.

[27] Pourya Hoseini, Shuvo Kumar Paul, Mircea Nicolescu, and Monica Nicolescu. Next best view planning in a single glance: An approach to improve object recognition. *SN Computer Science*, 4(1):51, 2022.

[28] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Kosecka, and Alexander C. Berg. A dataset for developing and benchmarking active vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[29] Phil Ammirato, Alexander C Berg, and Jana Kosecka. Active vision dataset benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2046–2049, 2018.

[30] Jie Wu, Tianshui Chen, Lishan Huang, Hefeng Wu, Guanbin Li, Ling Tian, and Liang Lin. Active object search. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 973–981, 2020.

[31] Ziyue Wang, Chi Chen, Fuwen Luo, Yurui Dong, Yuanchi Zhang, Yuzhuang Xu, Xiaolong Wang, Peng Li, and Yang Liu. Actiview: Evaluating active perception ability for multimodal large language models, 2024.

[32] Tianyi Liu and Benjamin Ward-Cherrier. The tip benchmark: A tactile image-based psychophysics-inspired benchmark for artificial tactile sensors. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 94–106. Springer, 2024.

[33] Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward H. Adelson. Transferable tactile transformers for representation learning across diverse sensors and tasks, 2024.

[34] Sudharshan Suresh, Zilin Si, Stuart Anderson, Michael Kaess, and Mustafa Mukadam. MidasTouch: Monte-Carlo inference over distributions across sliding touch. In *Proc. Conf. on Robot Learning, CoRL*, Auckland, NZ, December 2022.

[35] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward H Adelson. Active clothing material perception using tactile sensing and deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4842–4849. IEEE, 2018.

[36] Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and go: Learning from human-collected vision and touch. *arXiv preprint arXiv:2211.12498*, 2022.

[37] Shan Luo, Wenzhen Yuan, Edward Adelson, Anthony G Cohn, and Raul Fuentes. Vitac: Feature sharing between vision and tactile sensing for cloth texture recognition. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2722–2727. IEEE, 2018.

[38] Wenzhen Yuan, Shaoxiong Wang, Siyuan Dong, and Edward Adelson. Connecting look and feel: Associating the visual and tactile properties of physical materials. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5580–5588, 2017.

[39] Yunzhu Li, Jun-Yan Zhu, Russ Tedrake, and Antonio Torralba. Connecting touch and vision via cross-modal prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10609–10618, 2019.

[40] Samson Yu, Kelvin Lin, Anxing Xiao, Jiafei Duan, and Harold Soh. Octopi: Object property reasoning with large tactile-language models. *arXiv preprint arXiv:2405.02794*, 2024.

[41] Letian Fu, Gaurav Datta, Huang Huang, William Chung-Ho Panitch, Jaimyn Drake, Joseph Ortiz, Mustafa Mukadam, Mike Lambeta, Roberto Calandra, and Ken Goldberg. A touch, vision, and language dataset for multimodal alignment. *arXiv preprint arXiv:2402.13232*, 2024.

[42] Ning Cheng, Changhao Guan, Jing Gao, Weihao Wang, You Li, Fandong Meng, Jie Zhou, Bin Fang, Jinan Xu, and Wenjuan Han. Touch100k: A large-scale touch-language-vision dataset for touch-centric multimodal representation. *arXiv preprint arXiv:2406.03813*, 2024.

[43] Ruohan Gao, Yiming Dou, Hao Li, Tanmay Agarwal, Jeannette Bohg, Yunzhu Li, Li Fei-Fei, and Jiajun Wu. The objectfolder benchmark: Multisensory learning with neural and real objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17276–17286, 2023.

[44] Ruohan Gao, Zilin Si, Yen-Yu Chang, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. Objectfolder 2.0: A multisensory object dataset for sim2real transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10598–10608, 2022.

[45] Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. *arXiv preprint arXiv:2109.07991*, 2021.

[46] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020.

[47] Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023.

[48] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023.

[49] Kenneth Y Goldberg and Ruzena Bajcsy. Active touch and robot perception. *Cognition and Brain Theory*, 7(2):199–214, 1984.

[50] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. Revisiting active perception. *Autonomous Robots*, 42:177–196, 2018.

[51] Cristiana De Farias, Naresh Marturi, Rustam Stolkin, and Yasemin Bekiroglu. Simultaneous tactile exploration and grasp refinement for unknown objects. *IEEE Robotics and Automation Letters*, 6(2):3349–3356, 2021.

[52] Tim Schneider, Cristiana de Farias, Roberto Calandra, Liming Chen, and Jan Peters. Active perception for tactile sensing: A task-agnostic attention-based approach, 2025.

[53] Ruohan Gao, Zilin Si, Yen-Yu Chang, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. Objectfolder 2.0: A multisensory object dataset for sim2real transfer. In *CVPR*, 2022.

[54] 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*, 7(2):3930–3937, 2022.

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

[56] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[58] Weihang Chen, Yuan Xu, Zhenyang Chen, Peiyu Zeng, Renjun Dang, Rui Chen, and Jing Xu. Bidirectional sim-to-real transfer for gelsight tactile sensors with cyclegan. *IEEE Robotics and Automation Letters*, 7(3):6187–6194, 2022.

[59] Cédric Beaulac and Jeffrey S. Rosenthal. Introducing a new high-resolution handwritten digits data set with writer characteristics. *SN Computer Science*, 4(1), November 2022.

[60] Franka Inc. Homepage, January 2025. [Online; accessed 28. Jan. 2025].

[61] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[62] Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. Crossq: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. *arXiv preprint arXiv:1902.05605*, 2019.

[63] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claim of the paper is the introduction of the first benchmarking protocol for active tactile perception. Section 2 supports the claim we are the first with a comprehensive literature review. Furthermore, our two benchmark suites and the datasets are introduced in Section 3, with additional details in the appendix and are provided online.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 5 we discussed limitations and future directions of the work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: [NA] .

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Throughout Section 3 we present our benchmark suites. Throughout the appendix, and particularly in Appendix F and Appendix H we present detailed description on how each of our environments work as well as implementation details. We add that our code is also available online with detailed documentation. Furthermore, for the dataset collection, we fully describe the steps taken to generate the MNIST 3D Meshes (in Section 3.4 and Appendix D) and the real tactile data collection (in Section 3.5 and Appendix E).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide all the code for our benchmark environments with extensive documentation.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details both in Section 4, Appendix G and Appendix H

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our results in Section 4 and Appendix G show one standard deviation for each learning curve.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All implementation and performance details for the experiments are present in Section 4, Appendix G and Appendix H

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research fully conforms to the NeurIPS Code of Ethics. No human subjects or participants were involved in the creation of the datasets used, so there are no related privacy or consent concerns. All data and code are released publicly on GitHub and Hugging Face under permissive licenses (MIT for code, CC-BY for data), ensuring long-term accessibility, transparency, and reproducibility. We have verified that none of the datasets used are deprecated and that they comply with licensing terms. Our work poses no foreseeable risks related to safety, security, discrimination, or environmental harm.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA] .

Justification: This work introduces a benchmark for active tactile perception in robotic systems. It is foundational in nature and not tied to any direct applications or deployments. While it may contribute to progress in robotic manipulation over the long term, we do not foresee any immediate societal impact from this work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA] .

Justification: None of our data or benchmark suites has any significant risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, all external assets used in this work are properly cited in both the main paper and the appendix. We explicitly list the main assets used in our environment in Table 2, along with their associated licenses. All listed assets are released under a CC-BY license, and we have ensured that their terms of use are fully respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: All the code, models, and datasets we have released come with detailed documentation. Additionally, Section 3, Appendix F, Appendix D, Appendix E and Appendix H provide details of the released assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA] .

Justification: -

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing or research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA] .

Justification: This research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.