

©Copyright 2026

Carolina Higuera Arias

Towards Tactile Intelligence: Grounding Robot Manipulation Models in Multisensory Contact

Carolina Higuera Arias

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:

Byron Boots, Chair

Dieter Fox

Siddhartha Srinivasa

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Towards Tactile Intelligence: Grounding Robot Manipulation Models in Multisensory Contact

Carolina Higuera Arias

Chair of the Supervisory Committee:

Byron Boots

Computer Science and Engineering

Touch comes before sight, before speech. In today’s AI landscape, this Margaret Atwood quote is playing out in reverse; despite touch being a crucial modality for physical interaction, its integration into robot manipulation remains secondary. While significant progress has been made in spatial and semantic understanding for robot manipulation, achieving physical robustness, reactivity, and dexterity remains a fundamental challenge. Current paradigms often rely on exocentric vision to infer the complex dynamics of hand-object-environment interactions. This task is frequently ill-posed due to visual occlusions and the latent nature of contact characteristics such as friction, stiffness, and force. To achieve truly fluent manipulation skills, robots must not only understand the visual cues of their environment but also learn to speak the primary language of interaction: *contacts*.

Unlike vision, which is largely captured through structured images, touch is inherently multisensory. It can be perceived through elastomer deformations, magnetic field perturbations, high-frequency vibrations, pressure, and other signals. This richness introduces significant complexity in abstracting meaningful contact information, which is often sparse, for downstream tasks. This thesis proposes a unified framework for tactile perception for incorporating contact feedback into robot decision-making. The research is organized into three hierarchical thrusts. First, we address the observability gap in contact-rich tasks by introducing Neural Contact Fields (NCF), a method for tracking arbitrary extrinsic contact

interactions that are typically occluded from vision. We demonstrate that providing robots with explicit spatial awareness of contact enables robust zero-shot sim-to-real transfer for complex insertion tasks.

Second, to address the lack of standardized tactile encoders and the prohibitive cost of labeled data collection, we introduce the **Sparsh** family of models. We adapt self-supervised learning techniques to the tactile domain to learn general-purpose representations from unlabeled data using masked objectives. These representations capture both static contact properties (such as material, 3-axis forces) and dynamic interaction cues (such as slip, grasp stability, and pose), offering a rich latent space that can be leveraged by policies in the same manner as pre-trained vision encoders. Our framework comprises Sparsh for vision-based tactile sensors, Sparsh-Skin for magnetic skins, and Sparsh-X, a multisensory backbone that fuses image, vibration, motion, and pressure signals into a shared latent space.

Finally, we integrate these tactile priors into a generative framework through Visuo-Tactile World Models (VT-WM). By grounding a robot’s internal “imagination” in local tactile feedback, we demonstrate that the model is less prone to hallucinations, maintains object permanence under heavy occlusion, and adheres to physical laws during planning. Together, these contributions provide a path toward data-efficient, contact-aware robot manipulation, demonstrating that the synergy between global context and local interaction is essential for reliable, fine-grained dexterity in the real world.

TABLE OF CONTENTS

	Page
List of Figures	iv
Chapter 1: Introduction	1
Chapter 2: Extrinsic Contact Tracking for Robot Manipulation	5
2.1 Introduction	5
2.2 Tracking Extrinsic Contact with Tactile Sensing	7
2.3 Improving Insertions Policies via Extrinsic Contact Tracking	17
2.4 Related Work	23
2.5 Discussion	25
Chapter 3: Self-Supervised Touch Representations for Vision-based Tactile Sensing	27
3.1 Introduction	27
3.2 Touch Representations via Self-Supervised Learning	30
3.3 TacBench: Tactile Sensing Benchmark	33
3.4 Comprehending Tactile Properties	36
3.5 Enabling Physical Perception	43
3.6 Enabling Manipulation Planning	49
3.7 Related Work	50
3.8 Discussion	52
Chapter 4: Self-Supervised Perception for Tactile Skin Covered Dexterous Hands	55
4.1 Introduction	55
4.2 Self-Supervised Representations for Tactile Skins	57
4.3 Training and Online Probing	61
4.4 Evaluation of Sparsh-skin for Contact Perception	63
4.5 Policy Learning: plug insertion	71
4.6 Related Work	73
4.7 Discussion	74

Chapter 5:	Multisensory Touch Representations for Robot Manipulation	76
5.1	Introduction	76
5.2	Learning Multisensory Touch Representations	78
5.3	Inferring Physical Properties with Sparsh-X	81
5.4	Sparsh-X for Policy Learning	83
5.5	Related Work	87
5.6	Discussion	90
Chapter 6:	Visuo-Tactile World Models	92
6.1	Introduction	92
6.2	World Models that Understand Contact	93
6.3	Contact Perception of Visuo-Tactile World Model in imagination	99
6.4	Causal Compliance	100
6.5	Zero-Shot Planning with Visuo-Tactile World Model	103
6.6	Related Work	106
6.7	Discussion	108
Chapter 7:	Conclusion	110
Bibliography		114
Appendix A:	Sparsh supplementary material	136
A.1	TacBench evaluations via fine-tuning	136
Appendix B:	Sparsh-skin supplementary material	140
B.1	Force Estimation	140
B.2	Joystick State Estimation	140
B.3	Pose Estimation	140
Appendix C:	Sparsh-X supplementary material	145
C.1	Dataset for SSL Pretraining	145
C.2	Datasets for Downstream Tasks	146
C.3	Benchmarking Sparsh-X for physical properties comprehension	148
C.4	Sparsh-X and Policy Learning	149
Appendix D:	Visuo-Tactile World Model supplementary material	153
D.1	Training Visuo-Tactile World Model	153

D.2	Contact perception with Visuo-Tactile World Model	154
D.3	Zero-shot planning with World Models	156

LIST OF FIGURES

Figure Number	Page
2.1 NCF architecture block diagram	8
2.2 Contrast between simulation and real tactile images for DIGIT sensor	9
2.3 Simulation environments and grasp configurations for the Extrinsic Contact dataset.	11
2.4 Snapshots of extrinsic contact predictions on simulated testing trajectories at key timesteps	13
2.5 Ablations over inputs for tracking extrinsic contact with NCF	15
2.6 Qualitative NCF performance with unseen objects in simulation and zero-shot real world.	16
2.7 Block diagram of policies to ablate the utility of extrinsic contact in insertion tasks	18
2.8 Success rate of mug-in-cupholder policies with and without extrinsic contacts	19
2.9 Evaluation of mug-in-cupholder policies in sim and real-world	20
2.10 Success rate of bowl-in-dishrack policies with and without extrinsic contacts	21
2.11 Evaluation of bowl-in-dishrack policies in sim and real-world	22
2.12 Snapshots of execution of insertion policies in real world	23
3.1 Touch representation learning via SSL diagram	30
3.2 Visualization of reconstructed tactile images from Sparsh embeddings	33
3.3 Real labeled data collection setup for TacBench tasks	35
3.4 Attentive probe architecture for TacBench	36
3.5 Summary of Sparsh performance across TacBench	37
3.6 Friction cone of test data and RMSE (mN) for force estimation task with DIGIT sensor.	37
3.7 Force field visualizations from Sparsh	41
3.8 Data collection setup for TacBench force and slip tasks	42
3.9 Contrast between Sparsh (VJEPA) and E2E in slip detection	43
3.10 Illustration of robustness of Sparsh (VJEPA) in slip detection under noisy ground truth labels	44

3.11	Visualization of class agreements between the ground truth and the Sparsh decoder for pose estimation task	46
4.1	Sparsh-skin block diagram for self-supervised learning of skin representations	59
4.2	Illustration of Xela signal corruption via masking for SSL prediction task .	61
4.3	Visualization of Xela signal reconstruction from Sparsh-skin embeddings. .	62
4.4	UMAP visualization of Sparsh-skin representations colored by object . . .	63
4.5	Decoder architectures for Sparsh-skin downstream tasks.	64
4.6	Hardware setup used for downstream evaluation of Sparsh-skin	66
4.7	Data collection protocol for force task with Xela sensors	66
4.8	Summary of results comparing Sparsh-skin on all tasks	68
4.9	Policy architecture for plug insertion task	71
5.1	Sparsh-X multisensory touch transformer architecture	78
5.2	Pretraining data collection setup for Sparsh-X	80
5.3	Performance of Sparsh-X on inferring physical properties	81
5.4	Plug-insertion task via imitation learning with Sparsh-X	84
5.5	Block diagram of tactile adaptation for sim-trained policies	86
5.6	Performance of tactile adaptation with Sparsh-X in in-hand rotation task .	88
6.1	Visuo-Tactile World Model illustration	94
6.2	Samples of tactile images from Digit 360 sensors used in visuo-tactile world model	95
6.3	Visuo-Tactile World Model architecture	96
6.4	Object permanence of visuo-tactile world model	100
6.5	Snapshots of VT-WM preserving object permanence	101
6.6	Causal compliance of visuo-tactile world model	102
6.7	Snapshots of VT-WM and causal compliance between contact and object motion	103
6.8	Success rate of plans via CEM with VT-WM and V-WM on real robot . . .	104
A.1	Performance of Sparsh ablations across TacBench	139
B.1	Correlation between ground truth and Sparsh-skin force estimation	141
B.2	Force sequence reconstruction from Sparsh-skin	142
B.3	RMSE convergence rates between Sparsh-skin and E2E in state estimation task	143
B.4	Reconstructed object pose trajectory via Sparsh-skin	143

B.5	Comparison of pose estimation accuracy of Sparsh-skin with and without palm sensing.	144
C.1	Distribution of SSL data for Sparsh-X training	145
C.2	Visualization of Digit 360 tactile modalities	146
C.3	Experimental setup for material-quantity classification task with Sparsh-X	147
C.4	Experimental setup and data distribution for the normal force regression with Sparsh-X	147
C.5	Confusion matrix for object-action-surface classification with Sparsh-X	149
C.6	Confusion matrix for material-quantity estimation with Sparsh-X	150
C.7	Real-world policy deployment architecture with Sparsh-X	151
C.8	Architecture overview of the plug insertion policy	151
D.1	Multitask Vision-Tactile Dataset for Visuo-Tactile World Model	153
D.2	Action controllability of visuo-tactile world model	155
D.3	Relationship between visual and tactile state prediction of visuo-tactile world model	156
D.4	Comparison of visuo-tactile world model rollouts with ground truth sequence	157
D.5	Visualization of world model plans for Reach Button task.	159
D.6	Visualization of world model plans for Push Fruits task.	160
D.7	Visualization of world model plans for Reach & Push task.	161
D.8	Visualization of world model plans for Wipe Cloth task.	162
D.9	Visualization of world model plans for Stack Cubes task.	163

ACKNOWLEDGMENTS

Doing a PhD has been one of my biggest dreams since I first started engineering school. At the time, I wasn't entirely sure what that path entailed or what I truly wanted to get out of it, but my family never wavered. They carried that dream with me, encouraging me every step of the way. To my mom and dad: thank you for being my biggest cheerleaders, for sharing your passions with me, and for always letting me fly free.

Along this journey, I have been fortunate to encounter people who saw my potential even when I didn't. In Colombia, pursuing graduate studies is not always the most straightforward path; the more one grows academically, the less appealing they can seem to a traditional workforce. That logic never made sense to me. I believe that as our society and economy grow, we need people with the academic and technical skills to drive that growth. I lived in this spirit during my master's, and I want to thank Fernando for the insightful conversations and for pushing me to be the small fish in the big tank.

I always aspired to be a Fulbright scholar, not just for the support, but because I believe in its mission of uniting nations through academic exchange. In 2020, a year that felt like the world was standing still, the Fulbright program put my dreams on wheels. I still have the notebook where I listed the schools and labs I hoped to reach out to; looking at those names now feels surreal.

I remember my first conversation with Byron over Zoom. I was a mix of nerves and pure excitement. After reading some of the Robot Learning Lab's publications, I was convinced I wanted to be part of the team, although knowing very little about the recruitment process or academic life in the U.S. Byron, I am deeply grateful to you for believing in my potential and for playing such a fundamental role in this life-changing experience. Your advising and your insistence on critical thinking have seeded a path for my future success.

I arrived in Seattle in September 2021 to a city that was sunny, warm, and green—little

did I know what the weather had in store for me later! Selest gave me my first tour of the Gates Center and the robotics lab. He probably didn't realize it, but he was showing me my version of Disneyland. Thank you, Selest, for your help in the lab, but especially for being a person who brings joy.

At UW, I found my academic family in the Robot Learning Lab, the best vibes in robotics. I cherish the memories of whiteboard presentations, Slack standup updates, and our outings to Fremont Brewery. Thank you to Jake, Sandesh, Kay, Adam, Anqi, Mohak, Yuxiang, Boling, Rosario, Tyler, Kevin, Rohan, Sanghun, Mateo, Harine, and former post-docs Siyuan, Sasha, Alex, and Guanya.

While I joined the lab focused on RL, I didn't account for Siyuan showing me a demo of his GelSlim sensors. As I began working on robot manipulation, it suddenly made sense: touch was the missing key. That first year was a whirlwind of catching up: a new language, a new culture, a new research area and learning how to control the Franka. Siyuan, thank you for being my first mentor and for your incredible patience.

Later, I had the fortune of working with Mustafa. You have played a vital role in my growth, always open to chatting about ideas and letting us shape the vision of truly dexterous manipulation. Thank you for your encouragement, advising, and for opening doors to amazing opportunities at Meta. My time with the GUM team at FAIR was where the dream of a unifying language of touch really took hold. I want to thank Tess, Jessica, Zeo, Suddhu, Haozhi, Franzie, Taosha, and Sergio for the brainstorming sessions and technical guidance. To Mike and Tingfan, thank you for developing awesome tactile hardware and saving me whenever the robots broke. Francois, thank you for the mentoring and the life conversations that helped me gain perspective. Mustafa and Jitendra, thank you for your inclusive vision, and Akash, I am so thankful we got to push this vision to the sky together.

Another surreal chapter of my research took place at Boston Dynamics. Alberto, thank you for your trust; that internship made me believe I was ready for new challenges. Thank you for the coffee chats in Spanish and the technical guidance. Working on Atlas was a dream, and I want to thank Jan and Saumya for their support. Suddhu, thank you for

the fun collaboration, sharing the excitement of seeing our policies (almost!) work was a highlight. To Lev, Shreyas, Phil, Eric, and Hojun: you guys rock.

Beyond the research, there were moments when I just needed a friend. This achievement belongs to Mohak, Daphne, Mateo, Julio, and Melanie as well. Thank you for being there through the happy and the stressful times. I am lucky to call such great researchers my true friends. Rosario, you are the answer to one of the most important questions I have in life, and for that, I will always be grateful.

Finally, my heart goes back to Colombia. Sonia, Pao, and Natis: you are my sisters. Thank you for reminding me how strong I am since our high school days. And most importantly, Mami, Papi, Richi, Franco, and Barto: you are my fuel. This achievement is yours as much as it is mine. Thank you for your love and your trust.

DEDICATION

to my family: papá, mamá, richi, franco y bartolomé

Chapter 1

INTRODUCTION

The last two years have marked a paradigm shift in robotics reasoning, bringing us into an era of semantic success. By integrating web-scale vision-language knowledge, robots have gained an unprecedented ability to follow natural language instructions and reason about the world at a high level. However, semantic understanding is not a substitute for physical robustness. While a robot may now know that an object is fragile or heavy based on visual heuristics or language priors, guessing physical properties from pixels is insufficient for the safety and precision required in real-world deployment. To interact with the physical world reliably and in a truly dexterous manner, robots must speak the primary language of manipulation: *contacts*.

Touch offers a direct, unambiguous window into the physics of the hand-object interaction. As the author Margaret Atwood noted, “*Touch comes before sight, before speech.*” In the robotics domain, however, this natural hierarchy is often reversed, with tactile feedback treated as an optional modality. To move beyond pick-and-place primitives and achieve true dexterity, robots must do more than just learn from visual experience; they must have the intrinsic capability to interpret and leverage the rich, high-frequency stream of touch feedback.

This thesis proposes learning-based approaches to inject robot-object-environment contact into physics-grounded actors that are not just spatial, but specifically *contact-aware*. Our framework spans the tracking of extrinsic contacts, the development of general touch representations that capture both static and dynamic properties, and the grounding of world dynamics in the physical reality of touch.

1.0.1 *The Duality of Perception for Robot Manipulation: Vision and Touch*

The challenge of modern robot manipulation lies in the reconciliation between exocentric “seeing” and egocentric “feeling.” While vision is an extraordinary tool for global context and learning broad manipulation behaviors, relying solely on pixels forces a model to indirectly infer the underlying physics of hand-object interaction. Properties such as friction, stiffness, and force profiles, as well as critical contact events like incipient slip, are often invisible to exocentric cameras, yet they are completely observable through the tactile modality. By omitting these signals, vision-only models are forced to assume the dynamics of contact, leading to brittle policies that cannot recover from the physical uncertainties of the real world.

There is no single, universal way to perceive touch. Human tactile sensing relies on a diverse set of mechanoreceptors that capture skin deformation, vibrations, and pressure at varying rates and resolutions. Robotics reflects this complexity through a plethora of specialized hardware. Vision-based tactile sensors capture elastomer deformation as high-resolution images to extract geometry and shear force profiles, while magnetic-skin sensors offer flexible form factors by measuring perturbations in a magnetic field. Supplementary modes, such as contact microphones for high-frequency vibrations or IMUs for finger motion, further enrich this sensory stream. This multisensory richness, however, introduces a significant computational hurdle: determining how a downstream policy or dynamics model can effectively parse the contact information embedded in such heterogeneous raw signals.

This complexity leads to a significant representation gap in current learning-based robotics. While incorporating vision is now straightforward thanks to pre-trained backbones like ResNet [81], CLIP [150] or DINO [22, 140], there is no established recipe for how a model should leverage tactile data. Because touch is inherently local and possibly high-dimensional, models trained from scratch often suffer from modality collapse, where the learner ignores the subtle, sparse tactile signal in favor of more dominant visual or proprioceptive variations. Furthermore, tactile features are often learned task-dependent; a geometry-heavy task may require pattern recognition from tactile images, whereas tight insertions or reorientation tasks benefit more from explicit extrinsic contact tracking.

Ultimately, the goal of this thesis is to bridge this gap by demonstrating that the adaptation of self-supervised representation learning to the tactile domain offers a scalable path across different sensors and tasks. By complementing the high-level semantic success of exocentric vision with force and contact states, we provide the physical grounding required for true reactivity. This synergy between global context and local interaction motivates the primary research statement of this thesis:

A diverse set of learning-based models designed to instill contact-awareness in robot actors. By addressing extrinsic contact tracking, general touch representations, and visuo-tactile world dynamics as distinct yet complementary pillars, this work provides the necessary perception tools for achieving data-efficient and physically grounded dexterous manipulation.

1.0.2 Thesis Overview and Contributions

Guided by the vision of contact-grounded perception, this thesis proposes general algorithmic frameworks aimed at enabling robots to perform fine-grained manipulation. The work is organized based on the nature of the tactile information and the level of sensory integration, moving from explicit geometric tracking to unified multisensory representations and, finally, to generative world models.

Chapter 2: Spatial Awareness via Contact Estimation. We first address the "observability" gap in contact-rich manipulation. We introduce Neural Contact Fields (NCF), a method to track arbitrary multi-modal extrinsic contacts—interactions between objects and the environment that are typically occluded from vision. By leveraging neural fields to generalize across object shapes, we demonstrate that providing a robot with explicit spatial awareness of contact enables zero-shot sim-to-real transfer for complex insertion tasks, outperforming "blind" proprioceptive baselines.

Chapter 3: Scaling Vision-Based Tactile Representations. To move beyond task-specific tactile extraction and towards scalability, we introduce Sparsh, a framework for learning general-purpose touch representations. By training on over 460k tactile images and adapting masking-based self-supervised learning techniques, we demonstrate that these pre-trained encoders provide a scalable "backbone" for tactile intelligence that outperforms

end-to-end models across a wide range of perception and manipulation tasks.

Chapter 4: Extension to Magnetic Skin Sensing. To enable dexterity on complex embodiments like multi-fingered hands, we extend the SSL paradigm to non-visual sensors. We present Sparsh-Skin, a pre-trained encoder for magnetic-skin sensors. This work overcomes the inherent noise and uninterpretability of magnetic flux signals, providing a standardized latent space for fast, high-frequency tactile feedback.

Chapter 5: Multisensory Fusion for Policy Learning. Touch sensing is inherently multifaceted. We introduce Sparsh-X, a unified backbone that fuses tactile image, vibrations, motion, and pressure signals into a rich latent space. By training on diverse manipulation atomic actions (e.g. tapping, sliding, rotating) and a diversity of material, we demonstrate that integrating multisensory tactile embeddings into robot policies leads to superior performance and adaptability to changes in tactile properties in fine-grained tasks like in-hand rotation.

Chapter 6: Contact Grounding for World Models. Finally, we integrate these tactile priors into a generative framework. We introduce the Visuo-Tactile World Model (VT-WM). By grounding a robot’s “imagination” in both global vision and local touch, we show that the model maintains object permanence under occlusion and adheres to physical laws. This grounding enables more reliable zero-shot planning for ubiquitous manipulation skills like wiping and stacking.

Chapter 2

EXTRINSIC CONTACT TRACKING FOR ROBOT MANIPULATION

2.1 Introduction

Contact is inevitable; is estimation optional? During mundane activities like inserting a mug in a holder we often leverage our sense of touch to infer when and where the mug is in contact with the holder. We use this feedback, for example, when the mug’s handle is stuck on top of the holder to guide it towards a position that allows for correct insertion. In this work, we study how robot manipulation policies can benefit from increased spatial awareness through estimation of extrinsic contacts.

Extrinsic contacts characterize the interactions between an object being manipulated and the environment. Such interactions are prevalent in contact-rich manipulation tasks, such as peg insertion [40, 158, 57], packing [42], and tool manipulation [189, 165]. Access to extrinsic contact information offers several advantages. For example, it provides better observability for manipulation policies, improves planning over extrinsic contacts, and aids in controlling collisions. Despite the valuable information that an extrinsic contact tracker can provide about object-environment interaction, for example for extrinsic dexterity [221, 34], this problem is understudied in literature. Prior approaches for localizing environmental contacts are characterized by making hypothesis about the geometry of the contact. For example, Ma et al. [127] formulate the problem as constrained optimization, where different constraints must be defined to solve for parameters that localize external contact, such as position of a point, direction of a line, or normal direction of a plane. Kim et al. [101] uses factor graphs and tactile measurements to actively estimate the contact line between the object and its environment for a peg-in-hole insertion task, while having a controller that enforces the line contact. Although these works have started to explore the problem of extrinsic contact localization, several challenges remain. For instance, it is not clear: (i) how to track multi-modal contact patches without assuming a contact type; (ii) how to track contact to

no-contact interactions, and vice versa; and (iii) how to generalize to complex object shapes and new environments.

In this chapter we address these challenges with Neural Contact Fields (NCF), the first method that can track arbitrary multi-modal extrinsic contact from tactile perception. Our key insight is to leverage neural fields to generalize across different object shapes, given the local motion produced by the contact, which are well-captured by vision-based tactile sensors [106]. Our method estimates the probability of external contact for any 3D point on an object surface given a sequence of tactile images and the most recent history of end-effector poses and external contact probabilities. We train NCF to track extrinsic contact on three categories of objects with simulated tactile data. In experiments, we find that NCF is able to localize multiple contact patches without making any assumptions about its geometry and can capture contact/no-contact transitions, on unseen shapes in unseen environments.

We further demonstrate it can be used to learn an insertion policy in simulation which can then be transferred to the real world. We comprehensively study, both in simulation and in real-world experiments, the advantages of estimating extrinsic contacts with policy learning and inference for insertion tasks. Our evaluation framework involves mug-in-cupholder and bowl-in-dishrack insertion tasks. We train the policies in simulation and transfer them to the real world without fine-tuning. We evaluate four policies with different observation inputs: using only the robot’s proprioception, with tactile data, with NCF, and with ground truth extrinsic contacts. We find that the policy with NCF benefits from spatial awareness, exhibiting better performance than policies without extrinsic contact. Estimation of extrinsic contacts provides valuable information about object-environment interactions that the policy can leverage. In real-world experiments, policies with NCF outperform those without extrinsic contacts, achieving 33% higher success and $1.36\times$ faster execution on mug-in-cupholder, and 13% higher success and $1.27\times$ faster execution on bowl-in-dishrack.

2.1.1 Contributions

To summarize, this chapter makes the following key contributions:

- Section 2.2 presents the NCF (Neural Contact Field) architecture and a supervised training pipeline using simulated data to facilitate robust neural field learning.
- Section 2.2.4 evaluates the zero-shot transferability of NCF, demonstrating its efficacy when deployed in real-world environments without fine-tuning.
- Section 2.3 demonstrates that integrating extrinsic contact estimation significantly enhances the performance of robot insertion policies across both simulated and physical platforms.

2.2 Tracking Extrinsic Contact with Tactile Sensing

Robot manipulation tasks, including object insertion, packing, and tool use, inherently involve interactions with the environment through extrinsic contacts. For example, the subtle shear forces felt at the fingertips when misaligning a mug with its holder serve as critical feedback. Consequently, tactile sensing emerges as a compelling modality for implicitly capturing these extrinsic contact cues. This work investigates two key research questions: (a) *How can extrinsic contacts be effectively tracked using vision-based tactile sensing?* and (b) *How can this extrinsic contact information be leveraged to improve performance of insertion policies?*

2.2.1 Neural Contact Fields

We focus on the problem of tracking extrinsic contact between object and environment. Assuming that the grasp pose is fixed (same for all objects of the same category), our goals are threefold: i) to track multiple contact patches without assuming the contact type, ii) capturing contact to no-contact and vice-versa transitions, and iii) to generalize contact tracking to complex and unseen object shapes and environments. Our proposed Neural Contact Fields consists of three modules, as illustrated in Fig. 2.1:

- A *PointNet Encoder + Neural Point Descriptor Fields* that allow us to generalize across different object shapes.

Figure 2.1: NCF architecture for tracking extrinsic contact (between object and environment) given tactile sensing (between robot hand and object) inputs. For a set of query 3D points, NCF estimates the probability of any point in extrinsic contact.

- *Tactile Representations* following an autoencoder approach from a history of tactile images.
- *Contact Regressor* for estimating, in latent space, the probability of contact for any 3D point on an object surface.

Neural Point Descriptor Fields [164]. This model represents an object as a function that maps a 3D coordinate $\mathbf{x}$ to a spatial descriptor. It builds upon the architecture of Occupancy Networks [133], where the spatial descriptor $\mathbf{z}$ for a 3D point $\mathbf{x}$ is formed by concatenating the activations from all layers of the occupancy network. To ensure that these spatial descriptors are grounded in the object’s category, the model is conditioned on different low-dimensional latent codes. These latent codes are obtained from a PointNet encoder $\mathcal{E}$ [146] that takes the object’s point cloud $\mathcal{P}$ as input.

In our work, we consider scenarios where an object is rigidly grasped, implying that any motions induced by contact constraints are negligible. Consequently, the object’s configuration in the world frame can be described by a rigid body transformation $(\mathbf{R}, \mathbf{t}) \in SE(3)$, which is directly determined by the end-effector pose. Neural Contact Fields operates within

the latent space of spatial descriptors $\mathbf{z} \in \mathbb{R}^n$:

$$\mathbf{z}_i = g(\mathbf{x}_i|\mathbf{P}) = g(\mathbf{R}\mathbf{x}_i + \mathbf{t}|\mathbf{R}\mathbf{P} + \mathbf{t}) \forall i \in \{0, N\} \quad (2.1)$$

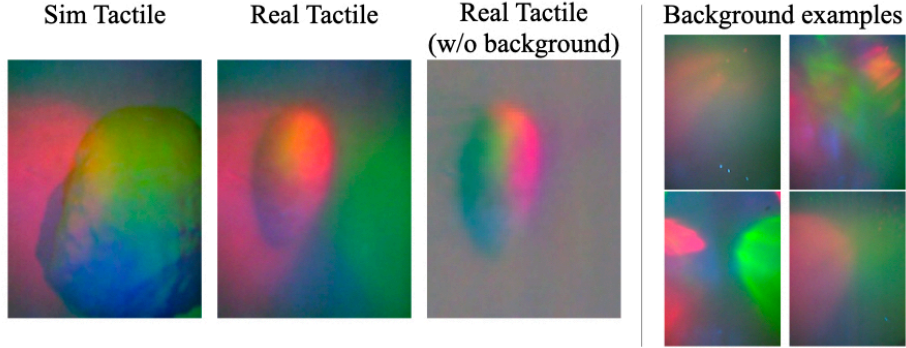


Figure 2.2: Contrast between simulation and real tactile images. The tactile simulator (TACTO [185]) aims to recreate the optical response of the sensor. However, the tactile imprint typically differs from real images. One contributing factor is the absence of simulation for contact dynamics, such as the deformation of the sensor’s elastomer. Also, non-contact images or backgrounds can differ among real sensors. Artifacts that resemble bubbles and shadows arise due to defects in the acrylic window of the sensor. Ensuring robustness to these diverse sources of variation is crucial for an effective tactile encoder.

Tactile Representations. In our setup, a parallel gripper equipped with DIGIT tactile sensors [106] on its fingers grasps the object. When the grasp is rigid and no external contact occurs, the object’s pose is fully determined by the end-effector pose. However, external contacts can induce slight pivoting or slipping of the object, motions that are effectively captured by vision-based tactile sensors.

To facilitate representation learning of tactile images under diverse, randomized contact interactions with the environment, we leverage the TACTO simulator [185] for scalable data collection. Nevertheless, a significant reality gap exists between simulated and real DIGIT images, as shown in Fig. 2.2. For example, real devices often exhibit optical artifacts due to manufacturing variations or elastomer wear, and no-contact images differ across devices. To harness the benefits of simulation for learning tactile embeddings while enhancing robustness

to the sim-to-real gap, we preprocess the images to directly extract static shear information. This preprocessing involves subtracting the corresponding no-contact state image. This step also effectively removes image artifacts that do not convey contact information but could otherwise confound pixel-level encoders.

We train a variational autoencoder (VAE) [103] to learn a regularized embedding space, which we have empirically found to be beneficial for sim-to-real transfer. After obtaining the embedding for each tactile image through the VAE, we concatenate the T most recent embeddings and input this resulting vector into a Multi-layer Perceptron (MLP). In our experiments, we set $T = 5$. Given our sensor’s sampling rate of 30 fps, this corresponds to approximately 0.17 seconds of tactile signal history.

Contact Regressor Our pipeline, illustrated in Fig. 2.1, aims to represent the extrinsic contact of an object as a function that maps a 3D point $\mathbf{x}$ to a probability of contact. This mapping is conditioned on the object’s motion, captured by a sequence $\mathbf{G}$ of tactile images, and a canonical point cloud description of the object $\mathbf{P}$:

$$f(\mathbf{x}, \mathcal{G}, \mathcal{P}) : \mathbb{R}^3 \times \mathbb{R}^{k \times 2 \times 320 \times 240 \times 3} \times \mathbb{R}^{3 \times N} \rightarrow [0, 1] \quad (2.2)$$

Here, k represents the number of time steps in the tactile image sequence, and the dimensions $2 \times 320 \times 240 \times 3$ correspond to the two sensors, image height, image width, and RGB channels, respectively. The point cloud $\mathbf{P}$ consists of N points in 3D space.

Neural Contact Fields (NCF) enable the tracking of extrinsic contacts by estimating, within the latent space of spatial descriptors, the probability that a 3D point $\mathbf{x}$ is in extrinsic contact. This estimation is conditioned on the implicit representation of the object’s motion resulting from contact, which is encoded from a history of tactile images:

$$f(\mathbf{z}, \mathcal{E}(\mathbf{G})) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow [0, 1] \quad (2.3)$$

Here, $\mathbf{z}$ is the spatial descriptor for point $\mathbf{x}$, and $\mathcal{E}(\mathbf{G})$ represents the tactile sequence embeddings.

Furthermore, we provide the NCF model with temporal information about the object’s state history, specifically the change in end-effector pose with respect to the current pose,

denoted as $\Delta \mathbf{e}_{t-1,t-2}$, for the preceding two timesteps. These inputs are fed into a temporal aggregation network to predict the contact probability for each query point. We explore a recurrent ResNet, Multi-layer Perceptron (MLP) and Transformer architectures as aggregation networks. The MLP model comprises three layers with output dimensions [512, 128, 1], with batch normalization and ReLU activation functions applied after the first two hidden layers. The Transformer model utilizes an encoder-only structure with two hidden layers and two attention heads. Its feature dimension is 512, and its output is passed through two linear layers with output dimensions [128, 1]. Finally, and for all variations of contact regressor, a sigmoid activation function is applied to the output to estimate the probability of contact for each 3D coordinate $\mathbf{x}$.

2.2.2 NCF training dataset



Figure 2.3: For collecting our Extrinsic Contact dataset, we simulate six kitchen cabinet environments (left) and utilize fixed, rigid grasp poses for all bottles, bowls, and mugs (right). Objects in these environments can move as the robot interacts with them during data collection.

Our training and evaluation datasets consist of six kitchen cabinet scenarios with YCB objects in it on different configurations, as presented in [Fig. 2.3a](#), four for collecting training data and two for testing the model. A Franka Panda arm with a parallel gripper and DIGIT

sensors [106] mounted on both fingers is interacting with these simulated environments in PyBullet [33]. The trajectories that the robot follows are randomly generated. We use TACTO [184] for simulating the sensors. For collecting extrinsic contact data, we use three categories of objects: mugs, bottles and bowls. We use several shapes for collecting training trajectories, obtained from the 3D Warehouse dataset. In order to evaluate the generalization of our model on unseen shapes for the grasped object, we collect the testing trajectories using new shapes for the three objects in testing scenarios. The initial grasp pose for each object is fixed and rigid and an example is shown in Fig. 2.3b.

The dataset provides for every contact event, access to a history of DIGIT images for each finger, end-effector pose, a reference point cloud of the grasped object, the set of query surface points and their ground truth probabilities of external contact.

2.2.3 NCF evaluation in simulation

We use a pretrained implementation of Neural Point Descriptor Fields to get the spatial descriptors for 3D coordinates in the point cloud for mugs, bottles and bowls. We trained separately the tactile encoder to learn the embedding of DIGIT image sequences. These sequences have length $k = 5$ and contain the images for both fingers for a history of 0.3s. We train the contact regressor on the training trajectories, which contain in total 4500 contact events. We use negative log-likelihood as loss function and Adam optimizer [102] with learning rate $1e^{-4}$.

We evaluate NCF-v1 offline with a GPU Nvidia RTX 3080, for which the forward pass takes ~ 66 ms. In Fig. 2.4 we show qualitative results of our pipeline tracking extrinsic contact. We show key snapshots of a test trajectory collected with unseen mugs, bottles, bowls, and new scenarios in simulation. From a qualitative point of view, we demonstrate the ability of NCF to track extrinsic contacts for the three novel objects. For example, with the mug trajectory, we show that the NCF-v1 can transition from patch to break contact to a new contact location. With the snapshots for the trajectories with the bottle and bowl, we show that NCF-v1 can localize multiple contact patches produced during complex contact interactions.

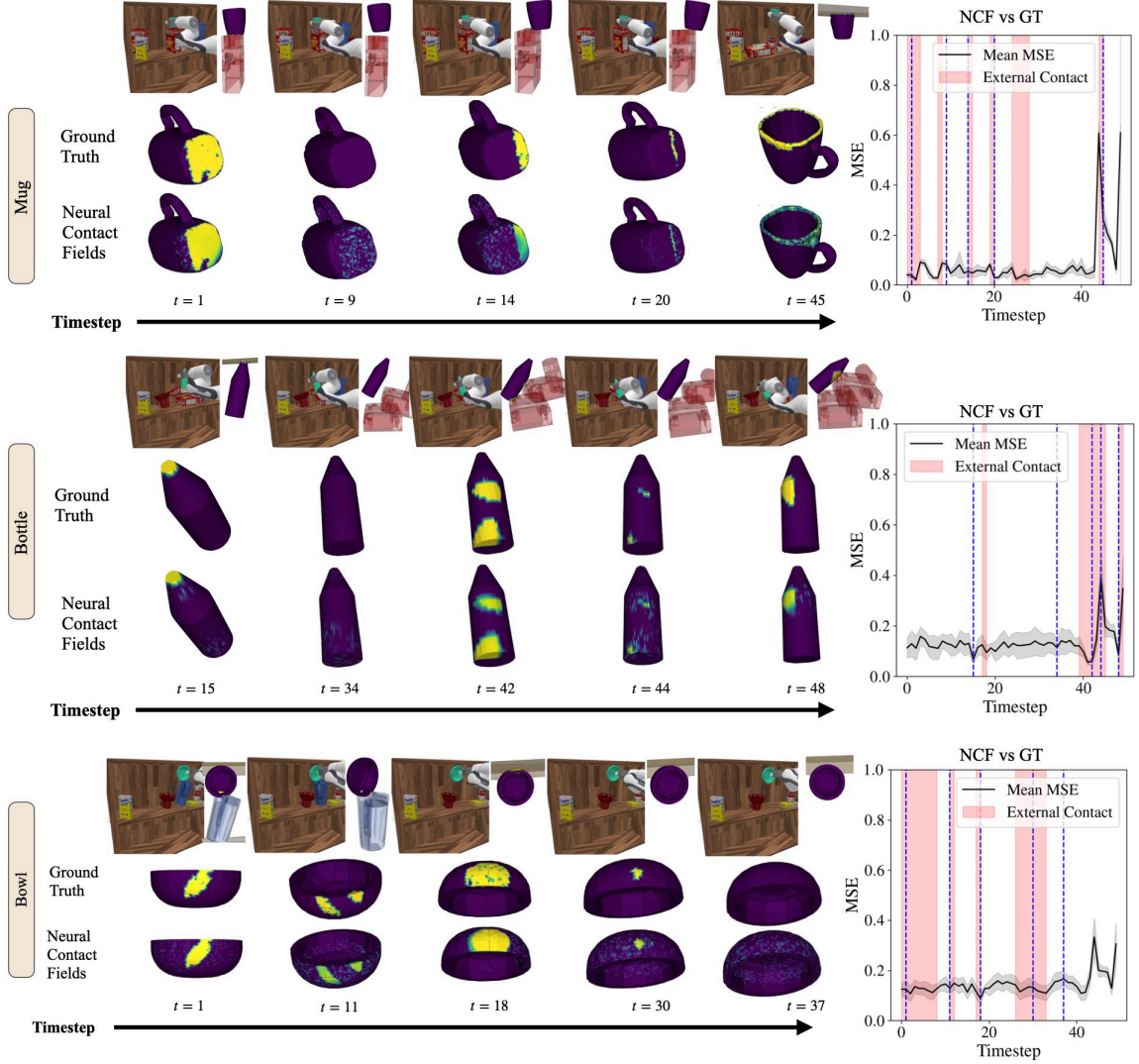


Figure 2.4: Snapshots of extrinsic contact predictions on simulated testing trajectories at key timesteps (blue dotted lines) for mugs, bottles and bowls. For each row: **[top]** Pybullet frame and zoom-in view of the contact interaction, **[middle]** extrinsic contact ground truth probabilities, **[bottom]** Neural Contact Fields prediction, and **[right]** MSE between the ground truth and estimated extrinsic contact probabilities for the trajectory over time.

For a quantitative evaluation, we analyze the mean squared error (MSE) between the ground truth external contact probability and the predictions. At each timestep we plot the MSE after running the model 10 times and applying Monte Carlo Dropout to compute

a 95% confidence interval. NCF is close to the ground truth during the majority of the trajectories. From these results, we identify cases that might induce a spike in error. For example, when the object transitions from no-contact to contact or when the shape of the contact patch changes drastically due to a non-smooth change in end-effector pose.

In addition, we perform ablations over the inputs of our NCF to quantify their contribution on the performance of the model. We compare four models: 1) NCF; 2) NCF without a history of contact probabilities and end-effector poses; 3) NCF considering only the current tactile frame rather than a history of images; and 4) NCF without a history of tactile images, contact probabilities and end-effector poses. In [Fig. 2.5](#) we plot the MSE over time for each model and we highlight the timesteps with external contact events. The contribution of a history of tactile images for tracking extrinsic contact is noticeable. This is expected, given the intuition that the sequences of tactile images contain indirect information about the external contact. This results also suggest that considering only a history of contact probabilities instead of the sequence of tactile images generates a loss of information about how the external contact has been evolving over time.

2.2.4 NCF zero-shot transfer to real-world

We examine how the NCF models transfer zero-shot to the real world. We evaluate three different variations, our former NCF-v1 model [\[84\]](#), and NCF-v2 with tactile autoencoder and MLP or Transformer contact regressor architectures [\[85\]](#).

For evaluation in real world, the robot executes predefined trajectories involving an unseen mug and bowl shape within a cupholder and dishrack environment, respectively (see [Fig. 2.6](#)). These trajectories mimic attempts to blindly insert the object. We generate pseudo-ground-truths by replaying the trajectory executed by the robot in simulation. While this doesn't precisely capture the physics and dynamics of real-world contacts, it serves as a realistic approximate reference for facilitating qualitative comparisons. We show in [Fig. 2.6](#) (middle and right), a visualization of the pseudo-ground-truth and the extrinsic contact patch estimated by the three models at key timesteps of the trajectories.

For the case of the unseen mug ([Fig. 2.6](#), middle), NCF-v2 (Transformer) effectively

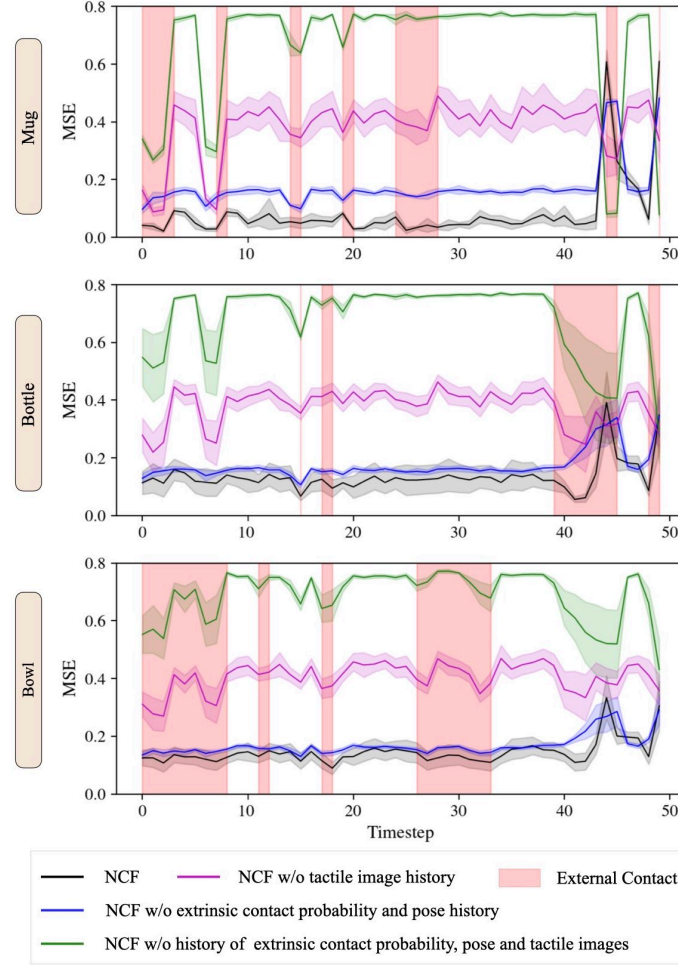


Figure 2.5: Ablations over inputs for tracking extrinsic contact with NCF. We compare MSE with respect to ground-truth external contact probabilities over representative trajectories. We evaluate four models: NCF, without extrinsic contact probability and end-effector pose history, without tactile images history (considering only the current frame), and without history of tactile images, extrinsic contact probability and end-effector pose. From previous timesteps considering extrinsic contact and sequences of tactile measurements significantly improves performance of the model.

identifies the contact points on the handle (first two rows) and detects contact on the foot of the mug when it is positioned on the edge of the cupholder (last two rows). NCF-v2 (MLP) captures the contact on the handle but struggles during contact transitions. On the other hand, NCF-v1 cannot make correct predictions on real-world data, mostly predicting zero contact everywhere. The predicted contact probabilities exhibit noise, and the model

Figure 2.6: Qualitative study with unseen bottle (left), mug (middle), and bowl in simulation (right) and real world (middle and right). The robot executes a predefined trajectory to make the object collide with the environment. We show estimated extrinsic contact patch from NCF-v1, NCF-v2 (MLP), and NCF-v2 (Transformer) on the object surface. In simulation, all models exhibit comparable performance in estimating extrinsic contact on unseen object shapes. For the real world, we show a pseudo-ground-truth for reference, that is recreated in simulation thus approximating the real contacts. NCF-v2 (Transformer) shows the best sim-to-real performance in estimating extrinsic contacts.

faces difficulties in accurately localizing contacts.

We also evaluate the performance of the models to unseen shape *and* unseen environment (Fig. 2.6, right) for the bowl-in-dishrack trajectory. NCF-v2 (MLP) and NCF-v2 (Transformer) effectively capture the transition from no-contact to contact. Both models capture the contacts as the bowl descends between the dividers. NCF-v2 (Transformer) demonstrates higher qualitative consistency and similarity to the pseudo-ground-truth, particularly towards the end of the trajectory when the bowl contacts the rack base while still situated between the dividers. We find similar results for NCF-v1 in that it cannot detect and track real-world contact locations.

We find that the proposed preprocessing stage and the VAE tactile autoencoder play an important role in facilitating the transfer, but they alone are not enough. The combination

with a contact regressor with enough representation power is crucial for making NCF robust to the challenges of real data. As NCF-v2 (Transformer) consistently outperforms other variants in estimating extrinsic contacts on real data, we use it in subsequent experiments for learning policies for insertion tasks in Section 2.3.

2.3 Improving Insertions Policies via Extrinsic Contact Tracking

Insertion tasks with tight tolerances pose a significant challenge, as precise alignment between the object and the receptacle is crucial for success. Extrinsic contact information plays a vital role in achieving this alignment. Neural Contact Fields (NCF) offer a practical learning-based method to perceive and track these extrinsic contacts in real-world scenarios using vision-based tactile sensing. In this section, we demonstrate that by providing observability over extrinsic contacts through NCF, robot insertion policies can achieve significantly higher success rates and faster execution times.

To have a clear understanding of the advantage of estimating extrinsic contacts for policy learning, we train four policies with different observation spaces using reinforcement learning and compare their performance, as illustrated in Fig. 2.7.

- **Proprioception Only**, π_{prop} . This policy only receives end-effector pose information at t , $t-1$, and $t-2$, with the rotation expressed as quaternion. We define these inputs as $\mathbf{e}_{t-2:t} \in \mathbb{R}^{21}$.
- **w/ Tactile**, π_{tac} . We augment the proprioceptive observations by incorporating the embedding of the sequence of tactile images. We use the same tactile VAE described in Sec.2.2.1. This policy allows evaluating if directly incorporating tactile sensing is sufficient when compared to the intermediate estimation of extrinsic contact.
- **w/ NCF-v2**, π_{NCF} (ours): We augment the proprioceptive observations by including the estimated contact patch $\mathbf{p}_t$ using NCF-v2. This contact patch corresponds to the points on the object with a contact probability above a threshold. The contact patch is processed through an MLP and its feature embedding is extracted using max-pooling.

- **w/ Ground Truth Contact, π_{gtc}** (privileged policy): We augment the proprioceptive observations with the point cloud of the ground truth contacts. This is an upper bound for simulation evaluation.

Figure 2.7: Overview of policies trained with different inputs to study the utility of estimating extrinsic contact in insertion tasks.

For each variant, we jointly optimize the policy π and the downsampling MLPs using PPO [159]. We train one policy for each task. For training and testing, we use the same object shape for all policies.

2.3.1 Task 1: *mug-in-cupholder*

Experimental Setup. A Franka Panda arm with a parallel gripper and DIGIT sensors on each finger starts with a mug grasped above a table. The goal is to insert the mug into a cupholder on the table. The cupholder features a slot on the side, as depicted in Fig. 2.12 (top). Successful completion requires precise alignment ensuring the mug’s handle fits into the cupholder’s slot.

The action space of the robot consists of 6-DOF transformations relative to the current end-effector pose, with the rotation expressed as axis-angle. These actions are set as the

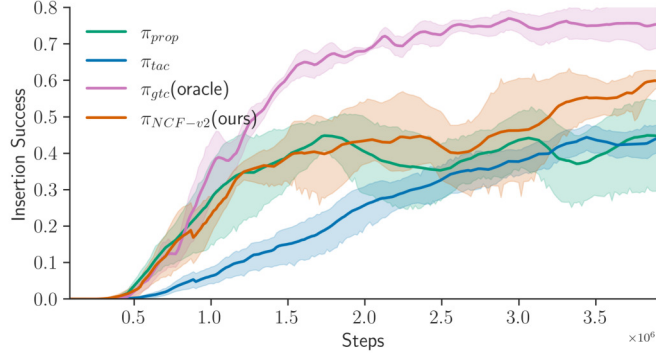


Figure 2.8: Success rate mean and 95% confidence interval for mug-in-cupholder task across 256 parallel simulated environments. Each policy is trained with 5 seeds. π_{NCF} (ours) outperforms policies using proprioception only or directly using tactile data.

target for the joint-space IK controller. We use IsaacGym for simulating the tasks, using the Factory environments [138] as reference, and use TACTO for simulating the touch sensors. For training, we run 256 environments in parallel, introducing randomization to the initial end-effector pose and running each episode for 250 timesteps.

Evaluation. Learning an insertion policy for this task is inherently challenging due to the narrow tolerance between the mug and the cupholder. While the robot can reduce keypoint misalignment, successful insertion ultimately hinges on the handle fitting into the slot.

Fig. 2.8 illustrates the progression of the success rate during policy training. These learning curves strongly affirm the advantage of incorporating extrinsic contacts in learning policies for contact-rich tasks. With access to privileged information in the form of ground truth extrinsic contacts, the policy achieves about an 80% success rate in correctly inserting the mug. However, relying solely on the history of the robot’s end-effector pose provides limited information, leading to early convergence to a low-performance policy. While π_{prop} is good at aligning the objects, it struggles with orienting the mug correctly. Integrating tactile information into the observations proves beneficial, but it demands extensive exploration to effectively map tactile signals to actions. π_{tac} requires nearly double the number of steps

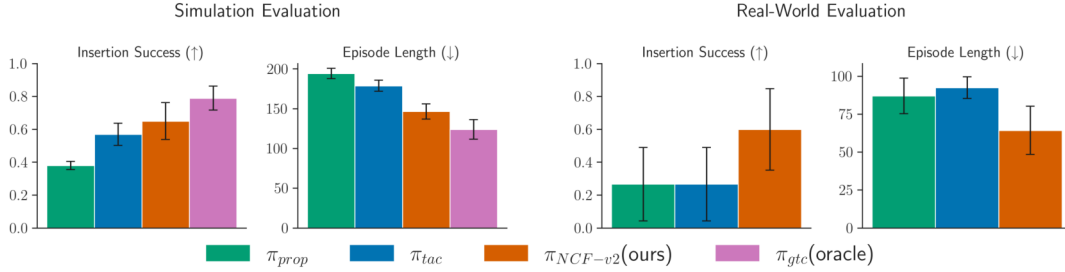


Figure 2.9: Evaluation of mug-in-cupholder policy. We show insertion success and episode length of 100 trials in simulation (left) and 15 trials in real (right). π_{NCF} (ours) performs better and faster compared to other policies that do not leverage extrinsic contact. Note different y-axes are used for episode length.

to achieve the same success rate as the proprioception-only policy. In contrast, a policy leveraging extrinsic contacts, like π_{NCF} (ours), demonstrates spatial awareness, meaning a relatively straightforward ability to align and orient the mug. This is facilitated by the availability of task-specific information, such as contacts on salient features of the object like the handle, body, and foot.

We test all policies in simulation, running 100 repetitions. We also perform real-world experiments to evaluate π_{prop} , π_{tac} , and π_{NCF} (ours), running 15 repetitions without performing any fine-tuning. For each test, both in simulation and on the physical robot, we randomize the initial end-effector pose, altering the initial position and orientation of the grasped mug.

Fig. 2.9 presents the performance metrics for both simulation and real-world experiments. In the simulation experiments, policies that incorporate extrinsic contact information demonstrate the highest success rates. Moreover, leveraging extrinsic contacts provides the additional benefit of learning policies capable of inserting the mug faster in terms of the number of timesteps required to complete the task.

Ultimately, the real-world metrics are the strongest evidence of the utility of extrinsic contacts for the downstream task. We do not deploy π_{gtc} (privileged policy) since access to ground truth extrinsic contact is only feasible in simulation. NCF enables us to navigate this limitation, making it possible to estimate extrinsic contacts using real-world data. π_{NCF}

(ours) has an average success rate of 60% over 15 repetitions, surpassing π_{prop} and π_{tac} at 27%. This represents an absolute improvement of 33%. Moreover, π_{NCF} (ours) proves to be $1.36\times$ faster, finishing the task on average in 23 fewer steps than its counterparts, with each trial limited to a maximum of 100 steps.

2.3.2 Task 2: bowl-in-dishrack

Experimental Setup. The robot starts with the bowl held above the dishrack. The goal is to insert the bowl into the second slot from the left, as shown in Fig. 2.12 (bottom). Successful completion requires approaching the target slot while maintaining awareness of the dividers.

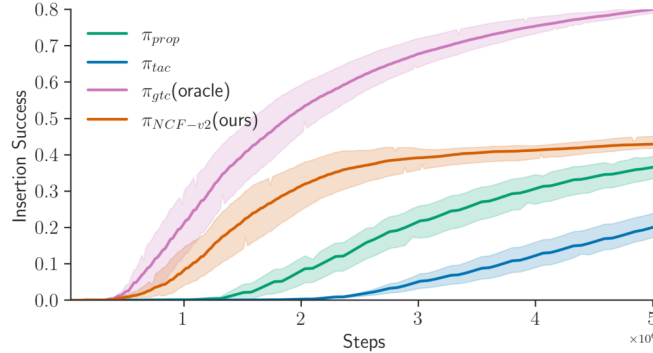


Figure 2.10: Success rate mean and 95% confidence interval for bowl-in-dishrack task across 256 parallel simulated environments. Each policy is trained with 3 seeds. π_{NCF} (ours) outperforms policies using proprioception only or directly using tactile data.

Evaluation. Fig. 2.10 shows the progression of the success rate during policy training. Having observability over extrinsic contacts proves again to be helpful for this task, especially when comparing the progression of the success rate for the privileged policy. Policies with only end-effector information or with raw tactile feedback require intensive exploration. Even with longer training, π_{prop} cannot reach performance comparable to the privileged policy.

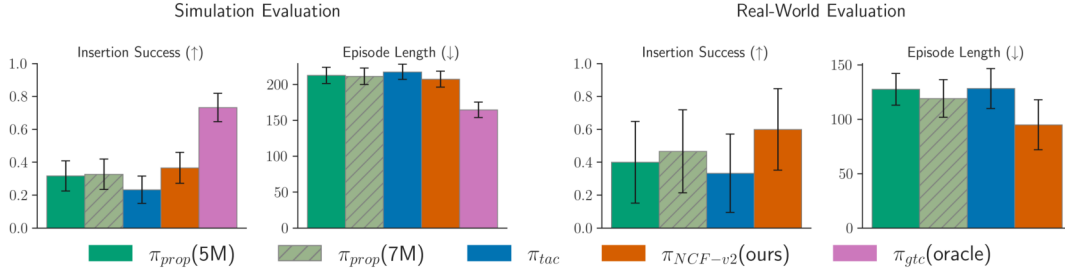


Figure 2.11: Evaluation of bowl-in-dishrack. We show success and episode length of 100 trails in simulation (left) and 15 trials in real (right). π_{NCF} (ours) performs better and faster compared to other policies that do not leverage extrinsic contact. Note different y-axes are used for episode length.

The performance gap between π_{gtc} (privileged policy) and π_{NCF} (ours) is accentuated by the fact that NCF-v2 predicts extrinsic contacts in an unseen environment. Additionally, since not all policies have converged after 5×10^6 steps, and given the minimal difference in success rates observed after 5×10^6 steps between π_{prop} and π_{NCF} (ours), we extend the training of π_{prop} to 7×10^6 steps to better discern the performance differences when deploying the policies.

We test the policies in simulation and in the real world without fine-tuning, running 100 and 15 repetitions, respectively. Fig. 2.12 shows the experimental setup and key snapshots during the execution of the policies. For each test, we randomize the initial end-effector pose. According to Fig. 2.11, in simulation, the privileged policy correctly inserts the bowl into the desired slot in the dishrack around 75% of the time. However, the other policies exhibit significantly lower performance, below 50% average success rate. It is worth noting that in simulation, we could not completely control the initial pose of the bowl due to undesired collisions with the gripper. Therefore, there was also a high degree of randomization in the initial bowl pose. While the privileged policy could implicitly get information about the bowl’s pose through the contact point cloud, the other policies could not. Even though NCF can estimate the contact patch, it also relies on the end-effector to infer the object’s pose.

In contrast, in the real world experiments, the initial pose of the bowl is controlled, and

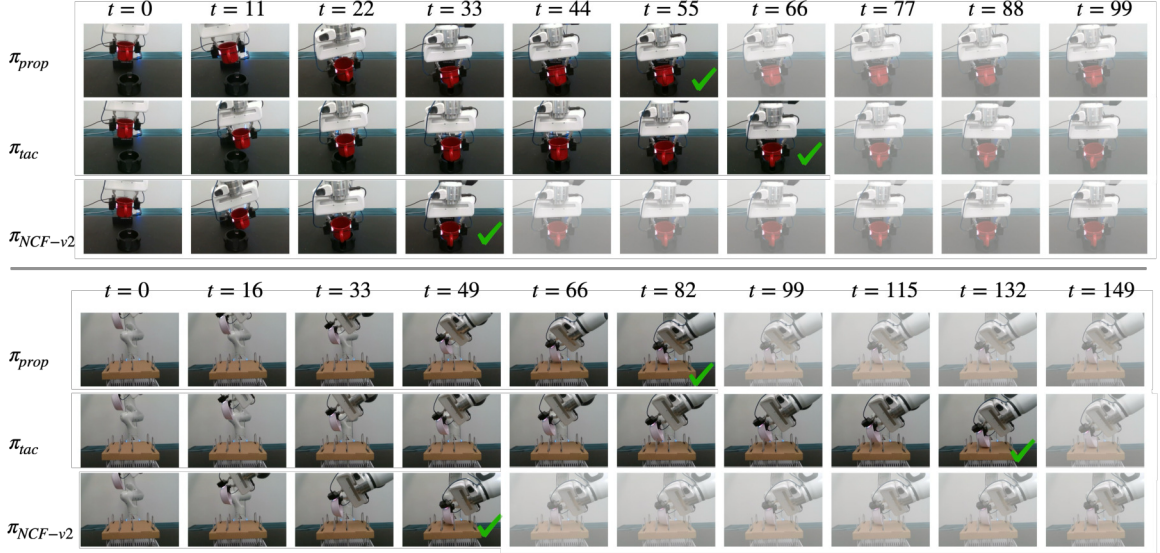


Figure 2.12: Execution of insertion policies in real world. Successful policy executions on mug-in-cupholder (top) and bowl-in-dishrack (bottom) insertion tasks in real world. On average, π_{NCF} (ours) is 13 – 33% more successful and $1.27 - 1.36\times$ faster at completing the tasks, compared to policies that do not leverage extrinsic contact. t refers to the step of the corresponding frame in a trajectory with 100 steps.

all policies exhibit better performance. π_{NCF} (ours) has the highest insertion rate, highlighting that NCF is generalizing to a new environment. A policy with only proprioceptive information is not able to gather the spatial understanding that this task requires, even if trained for a longer time. π_{NCF} (ours), on average, achieves an absolute improvement of 13% in success rate with respect to π_{prop} and completes the bowl insertion $1.27\times$ faster.

2.4 Related Work

External contact localization. Although localizing external contacts can provide rich information to respond correctly in downstream tasks, such as insertion, packing, or assembling, this important problem has received little attention. Ma et al. [127] discuss a theoretical basis for localizing environmental contacts. They formulate the problem as a constraint-based estimation subject to the kinematic constraints imposed by tactile measurements of the object local motion, as well as the kinematic and frictional constraints

imposed by rigid-body mechanics. This work presents a method to estimate three possible hypothesis of contact: point, line and patch contact. For each mode, different constraints must be formulated to solve for the parameters that allow to localize the external contact on the object, such as position of a point, direction of a line, or normal direction of a plane. To solve the problem, some assumptions must be guaranteed such as the object remains in contact with the environment and that the grasp is stable. The work acknowledges that a key limitation lies in having to assume a particular contact type to formulate the constraints and that a stronger implementation should consider multiple hypothesis for contact formations. In addition, [101] follows a similar approach, using factor graphs and tactile measurements to actively estimate the contact line between the object and its environment for a peg-in-hole insertion task. This method allows to parameterize the contact line, assuming that the object is always making contact, and generalizes well for objects with basic shapes and flat surfaces.

A learning based approach is presented in [35], in which a collision model is learned from point cloud of the object and its environment. The model predicts the likelihood that the object collides with the scene but does not provide further information about where is the contact located on the object. This method generalizes across four objects categories (mugs, cylinders, boxes, and bowls) with different scenes.

Our Neural Contact Fields generalize across three object categories, can localize multi-modal external contacts without making any assumptions about the contact type, and can capture contact/no-contact transitions.

Extrinsic contact in manipulation policies. Insertion tasks heavily rely on interactions with the environment and have been extensively studied by the robotics community. The inherent complexity of these tasks arises from their partial observability, as significant occlusions can occur during object alignment, even with the aid of vision systems. Hand-designed closed-loop control strategies for peg-in-hole insertion have been proposed in prior work [66]. Furthermore, the integration of vision-based tactile sensing with Reinforcement Learning (RL) policies for insertion has been extensively explored, often with a focus on direct training on real robotic systems [40, 56, 143].

To leverage the data efficiency of simulation, sim-to-real strategies for insertion tasks have also been developed. Some of these approaches involve an offline phase where robot motion and control parameters are learned using RL, followed by an online phase to learn a residual policy based on real force measurements [216]. Other sim-to-real techniques employ domain randomization across physical, optical, and task domains to enhance the transfer of learned policies to real-world scenarios [24, 173].

Precise placement has been identified as an application area that benefits from the prediction of extrinsic contacts [141]. This work utilizes force and vision-based tactile sensing to estimate extrinsic contact, but specifically on flat surfaces. Neural Deforming Contact Field (NDCF) [182] offers a method to jointly model object deformations and contact patches using implicit representations, leveraging a partial point cloud of the object and end-effector wrench information. This approach enables contact tracking on a single deformable tool (a sponge) across different environments. In contrast to these methods, our NCF allows for tracking extrinsic contacts on objects with complex geometries (such as mugs, bottles, and bowls) in diverse environments through vision-based tactile sensing.

2.5 Discussion

We presented in this chapter Neural Contact Fields (NCF), a novel method for tracking extrinsic contact. Our model outputs the probability of external contact for a set of query points on the 3D surface of an object based on vision-based tactile sensing. Our method works in the latent space of spatial descriptors, which allow us to generalize within categories of objects with different shapes. Our experiments in simulation demonstrate the capability of NCF in localizing and tracking complex contact interactions, such as multiple contact patches, and breaking and making of contact. We believe NCF can be used in downstream robot manipulation tasks and leave its applications for future work.

Additionally, we have demonstrated the utility of estimating extrinsic contacts from touch for learning robot insertion policies. Building upon prior work on NCF, we first made improvements to the tactile embedding and contact regressor to enable sim-to-real transfer of extrinsic contact estimation during real robot interactions. Through simulation and real-world evaluations, we verified our hypothesis that estimating extrinsic contact aids

policy learning for two insertion tasks, mug-in-cupholder and bowl-in-dishrack. Policies with access to estimated/ground-truth extrinsic contacts achieve higher success rates than a baseline policy with access to only proprioception and raw tactile data. In real-world experiments, policies with NCF outperform the baselines without fine-tuning, achieving 33% higher success and 1.36 times faster execution on mug-in-cupholder, and 13% higher success and 1.27 times faster execution on bowl-in-dishrack.

Limitations. The current NCF is restricted to tracking extrinsic contacts for three specific object classes (mugs, bottles, and bowls), assuming a fixed relative pose during grasping. To assess the performance of NCF on real-world data, we generate pseudo-ground truths by replicating the robot’s trajectory in a simulation. These pseudo-ground truths serve as references for evaluation but may not perfectly reflect actual contact interactions.

NCF assumes rigid grasp, although this could be relaxed by including an object-pose tracker in the loop. To get the correct reference point cloud, prior knowledge of the object class is necessary. To overcome these constraints, integrating NCF into a system that learns an implicit representation of various object shapes through vision can be advantageous.

Chapter 3

SELF-SUPERVISED TOUCH REPRESENTATIONS FOR VISION-BASED TACTILE SENSING

3.1 Introduction

Touch comes before sight, before speech. In today’s AI landscape, this Margaret Atwood quote is playing out in reverse despite touch being a crucial modality for humans to physically interact with the world. Touch provides a direct window into information like forces and contact during hand-object interactions, enabling dexterity. Vision-based tactile sensors [207, 44, 107, 115] have emerged as the leading form factor capable of capturing images of physical interactions at the sensor-object-environment interface, often inaccessible through vision. These images contain properties such as contact geometry, texture, and forces and have been leveraged across tasks like insertion [41, 86], pushing [124], grasping [20], localization [170], and pose and shape estimation [11, 168].

The prevailing approach to incorporating vision-based tactile sensors in robot tasks is to train custom models using labeled data [86, 30, 193, 121] to estimate useful states. However, this can be inefficient and results in repeated effort across different type of sensors like GelSight 2017 [207] (with markers) and DIGIT [107] (without markers) or different variety of tasks. For example, feature extractors trained on GelSight with markers may not transfer to other sensors, and encoders optimized for texture recognition [212] may not be suitable for tasks that require reasoning about forces or slip [209]. Supervision for building large general models is prohibitive as collecting large scale real world data with ground truth labels is challenging. For instance, properties like forces [126] and slip [16] require careful and expensive instrumentation in lab settings, while other properties like tracking deformations [49] or extrinsic contact [86] can be infeasible. To address this fragmentation in the literature across custom solutions, there is a need for touch representations that are broadly applicable to many tasks and many sensors, along with a benchmark of standardized

tasks useful in measuring progress. Taking inspiration from self-supervised learning (SSL) methods in computer vision, we extend these approaches to the tactile sensing domain and build a benchmark for evaluation.

In this chapter, we introduce a family of touch representations for vision-based tactile sensors trained with SSL. Specifically, we provide a recipe to adapt masking-based objectives from computer vision to the tactile domain, and train general-purpose touch encoders by curating a new Touch-Slide dataset and existing datasets of tactile images, namely YCB-Slide [170], Touch-and-Go [198], and ObjectFolder [63]. Pulling together additional unlabeled data points from the existing datasets we train our models on a total of 460k+ tactile images. Finally, we construct **TacBench**, a benchmark consisting of six touch-centric tasks that cover the space of relevant problems on tactile properties such as force estimation and slip detection, on perception such as pose estimation and grasp stability, and on robot manipulation such as policies for solving a bead maze.

In evaluations on **TacBench**, we find that **Sparsh** with SSL pre-training yield on average 95.1% improvement over task and sensor specific end-to-end models under limited labeled data budget (33%-50% of the collected amount) for any task. Additionally, we find **Sparsh** (DINO) and **Sparsh** (IJEPA) to be the most competitive outperforming **Sparsh** (MAE), indicating the merits of learning in latent space over pixel space for tactile images.

3.1.1 Contributions

To summarize, this chapter makes the following key contributions:

- [Section 3.2](#) introduces the recipe for learning general touch representations, **Sparsh** pre-trained with SSL on 460k+ tactile images,
- [Section 3.3](#) presents **TacBench** a benchmark of standardized tasks to evaluate touch representations and models
- [Sections 3.4](#) to [3.6](#) present a thorough comparison between different SSL paradigms and classical end-to-end approaches across the benchmark.

3.1.2 Review of SSL methods

In this research, we consider three SSL paradigms, namely **Sparsh (MAE)**, **Sparsh (DINO)**, and **Sparsh (IJEPA) & Sparsh (VJEPA)**.

Sparsh (MAE) is based on the principle of masked image modeling, where an encoder model is tasked with learning the contextual representations of substantially masked images, such that it enables reconstruction of the masked regions via a lightweight decoder. We use a ViT encoder and decoder for Sparsh, and the MAE loss corresponds simply to a L2 reconstruction loss:

$$\mathcal{L}_{\text{MAE}} = \|\mathbf{I}_{\text{target}} - \mathbf{I}_{\text{recon}}\|_2^2 \quad (3.1)$$

Sparsh (DINO) is based on the principle of self-distillation between two identical networks, where a student network learns to track the output predictions of a EMA teacher network. Cross-entropy loss is employed between the predictions of the student and teacher network, both of which consume different crops of the same input. Specifically, feature representations from each branch are passed through a MLP head, producing probability vector over an arbitrarily chosen number of classes. These scores are normalized to produce $\mathbf{p}_s$ and $\mathbf{p}_t$ for the student and teacher respectively.

$$\mathcal{L}_{\text{DiNO}} = - \sum \mathbf{p}_t \log \mathbf{p}_s \quad (3.2)$$

Sparsh (IJEPA) and Sparsh (VJEPA) share similarities with both masked image modeling and self-distillation. Here, we employ two identical networks termed context and target networks. The context network corresponds to a student network, which is tasked to predict the features from a EMA target network (teacher network), through a small predictor network. In this case, a L_2 loss over features is used to enforce similarity between the two branches. Specifically, the context network observes M global masks of an image to produce contextual features, which are then passed through a predictor to predict target network features of B_i local crops of the same image $\hat{\mathbf{s}}_{y_j}$. On the other hand, the target network consumes local crops of the image to produce $\mathbf{s}_{y_j}$.

$$\mathcal{L}_{\text{jepa}} = \sum_{i \in M} \sum_{j \in B_i} \|\hat{\mathbf{s}}_{y_j} - \mathbf{s}_{y_j}\|_2^2 \quad (3.3)$$

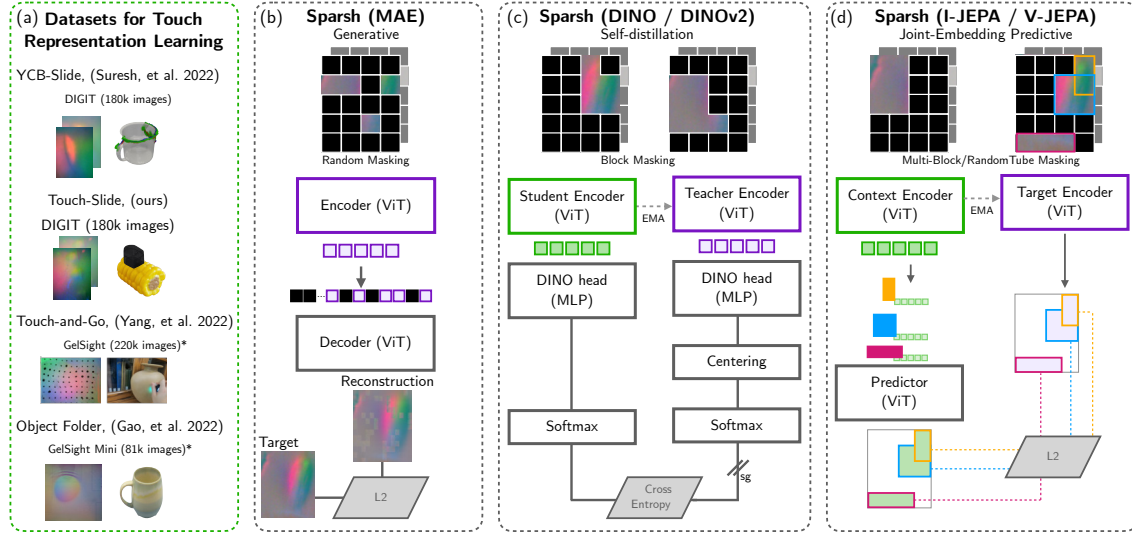


Figure 3.1: (a) We curate new and existing datasets of vision-based tactile sensors to train touch representations by adapting state-of-the-art SSL vision methods to the tactile domain, namely (b) Masked Autoencoder (MAE) [80], (c) DINO/DINOv2 [22, 140], and (d) Image/Video Joint-Embedding Predictive Architecture (JEPA) [7, 10].

3.2 Touch Representations via Self-Supervised Learning

Current approaches incorporating vision-based tactile sensors in robotic tasks lean on custom task and sensor specific solutions. This can be inefficient, and there is a growing need for general-purpose touch representations that can be more broadly useful. We envision the following guiding principles for such general touch representations: (i) provide performance benefits across many tasks including real-time robot manipulation, (ii) generalize across multiple types of sensors built on a similar operating principle, like vision-based tactile sensors, and (iii) improve performance by leveraging computation and diverse data at scale without the need for manual labels. Self-supervised learning (SSL) is promising in this regard, as it offers data-agnostic objectives based on wide-reaching concepts such as analysis-by-synthesis to train generalist models. This motivates the question of whether vision techniques such as masked image modeling (MIM) [80, 7] and self-distillation [22, 140]

*Without need for labels we can sample more images than reported in Touch-and-Go [198] and Object Folder [63].

can be extended to the domain of vision-based tactile sensors.

To this end, inspired from advances in self-supervised learning (SSL) in computer vision, we introduce **Sparsh**, a family of touch representation trained with SSL across multiple sensors such as DIGIT [107], GelSight 2017 (with markers [207]), and GelSight Mini (without markers). The tactile domain, however, imposes several challenges that impede a straightforward application of SSL approaches from vision towards touch representations.

Tactile sensors inherently provide local information; thus, images can be ambiguous when observed independently and can vary across grasp forces, materials, and shapes. Therefore, we investigate the optimal space for training SSL encoders. Specifically, we are interested in the efficiency of pixel reconstruction, latent reconstruction, or clustering approaches to learning representations in the presence of aforementioned ambiguities. We hypothesize that latent reconstruction and clustering could be more efficient in learning representations, as they focus model capacity away from fine reconstruction details [113]. Tactile images contain distractors, such as markers and light placement variations, which can significantly vary due to manufacturing discrepancies. To increase robustness to distractors, we perform background subtraction for both DIGIT and GelSight Mini (markerless). This process provides the model with a reference to no-contact, which conveys static shear information when a perpendicular force is applied to the elastomer. We find empirically that background subtraction helps models generalize across the same type of sensor.

To address the scarcity of labeled and even unlabeled data in the tactile domain that limits the training of large encoder models, we curate together new and existing vision-based tactile sensor datasets [170, 198, 63], totaling $\sim 661\text{k}$ images as illustrated in Fig. 3.1 and itemized in Table 3.1. 70% or 462.7k images are used for SSL pre-training which is an order of magnitude larger than any prior work on touch representations [21, 197, 48] (the rest are held out for monitoring training using online probes).

Tokenizing tactile images appropriately for SSL is important because many tasks such as slip detection and relative pose estimation require temporal reasoning. For SSL methods that operate on images, we concatenate two tactile images with a temporal stride of 5 samples across the channel dimension, $I_t \oplus I_{t-5} \rightarrow x \in \mathbb{R}^{h \times w \times 6}$. For a sensor operating at 60FPS, this corresponds to an inference window of approximately 80 *ms*, the reaction

time that humans need to adjust the grip force when detecting partial slip [213]. For SSL methods that operate on video (e.g. V-JEPA), we generate clips with 4 frames at $[t, t-2, t-4, t-6] \in \mathbb{R}^{4 \times h \times w \times 3}$ corresponding to an inference window of ~ 100 ms. Currently **Sparsh** is limited by data streaming rates, and not by inference time, as the models support inference rates of upto 112FPS (measured on an Nvidia RTX3080).

Dataset	Sensor	Size	Collector
YCB-Slide [170]	DIGIT	180k	Human
Touch-Slide	DIGIT	180k	Human
Touch and Go [198]	GelSight (markers)	> 13.9k	Human
ObjectFolder 2.0 [64]	GelSight	> 66k	Robot

Table 3.1: Tactile Datasets used for learning tactile representations. For all datasets, we incorporate images corresponding to transitions from non-contact to contact.

3.2.1 Training details

All encoder models are Vision Transformers (ViT) [46]. Although the main encoder models use ViT-B/14 as the standard architecture, following [7] we use a small ViT as the predictor network. All the models are pretrained without a [cls] token. For DINO, which decodes the [cls] token into classes, we repurpose ViT registers [36] to predict classes. In Table 3.2 we report the number of parameters for each encoder and their respective inference times.

	Sparsh (MAE)	Sparsh (DINO)	Sparsh (IJEPA)	Sparsh (VJEPA)
N. parameters	86254848	86255616	86386944	86537472
FPS	104	112	112	60

Table 3.2: Number of parameters and inference time for **Sparsh** backbones

We train all models on 8 Nvidia A-100 (80G) GPUs. In addition to training losses, to monitor training progress, we rely on online probes. Specifically, we find that for joint em-

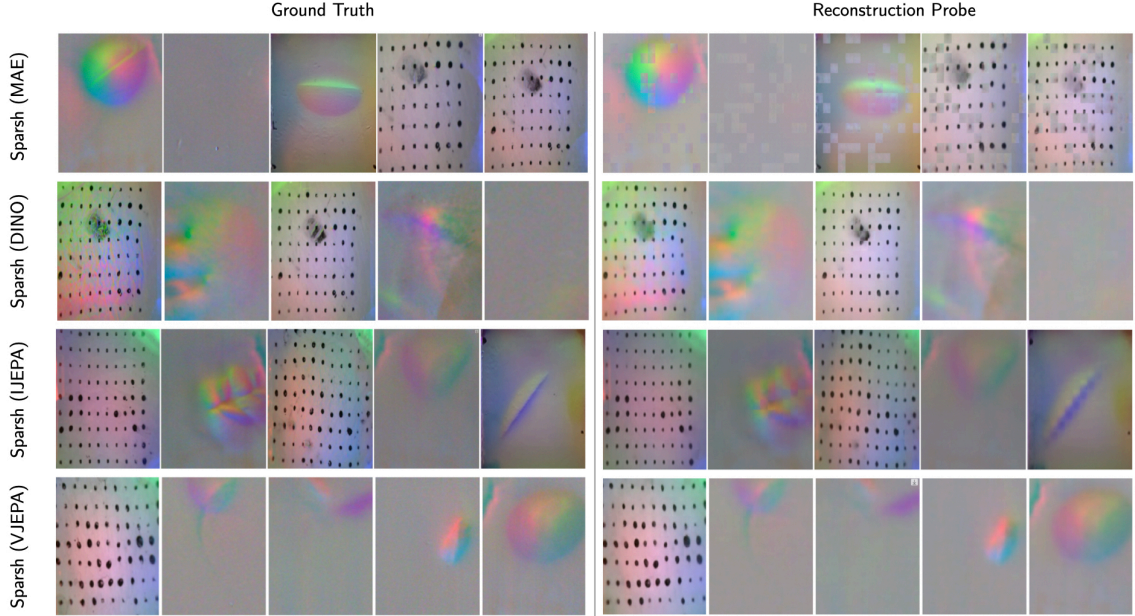


Figure 3.2: Visualization of reconstructed tactile images using the online probe to monitor SSL training of **Sparsh** models.

bedding predictive architectures, the training losses are not indicative of model convergence during optimization; therefore, proxy metrics such as reconstruction quality are helpful. For all methods, we utilize DPT [154] based decoders to decode the tactile representations back into tactile images. See Fig. 3.2 for some examples of tactile reconstructions from **Sparsh** embeddings. All encoder models are trained for 150 epochs. We use AdamW optimizer and use a linear rampup followed by a cosine schedule as the learning scheduler. Further, we find that tuning momentum value as well as the weight decay factor was important in observing training convergence without collapse. Additional information of hyperparameters is detailed in Table 4.1.

3.3 TacBench: Tactile Sensing Benchmark

We introduce **TacBench**, a collection of touch-centric tasks, and labeled datasets for standardized evaluation for vision-based tactile sensing. We compile data for all tasks from various sensors to evaluate the generalization of representations. These tasks are catego-

	Arch.	EMA decay	LR	Batch size
Sparsh (MAE)	ViT-B/14	N/A	1e-4	100
Sparsh (DINO)	ViT-B/14	0.998	1e-4	150
Sparsh (IJEPA)	ViT-B/14	0.996	6.25e-4	150
Sparsh (VJEPA)	ViT-B/14	0.996	6.25e-4	150

Table 3.3: **Training hyperparameters for Sparsh models.** All models run for 150 epochs with optimizer AdamW, a weight decay cosine schedule from 0.04 to 0.4, and a learning rate warmup of 30 epochs.).

rized under three main questions.

1. Do the representations comprehend tactile properties? Tactile sensing informs finger-object contact interaction properties like forces and slip that are crucial for robot manipulation. In [Section 3.4](#), we evaluate learned representation on estimating instantaneous normal and shear forces [T1] and visualizing force fields [T1A] [[209](#), [174](#), [128](#), [126](#)], and detecting slip [T2] [[209](#), [43](#), [183](#), [16](#)].

2. Do the representations enable perception? Tracking and accumulating slip states is essential for tasks like finger-gaiting and in-hand reorientation [[149](#), [199](#)]. In [Section 3.5](#), we evaluate the ability of the representations to track SE(2) pose changes of the object relative to the sensor [T3] [[11](#)], prediction of the stability of a grasp [T4] [[20](#)], and textile recognition [T5] [[210](#)].

3. Do the representations enable manipulation planning? Pre-trained representations can provide tactile features to a manipulation policy, improving training efficiency and test performance by eliminating the need to extract the states from raw sensor data. In ??, we design a bead maze [T6] manipulation problem as illustrated in [Fig. 3.3](#) (c), where the robot using tactile sensing is tasked to move a bead along a curved wire.

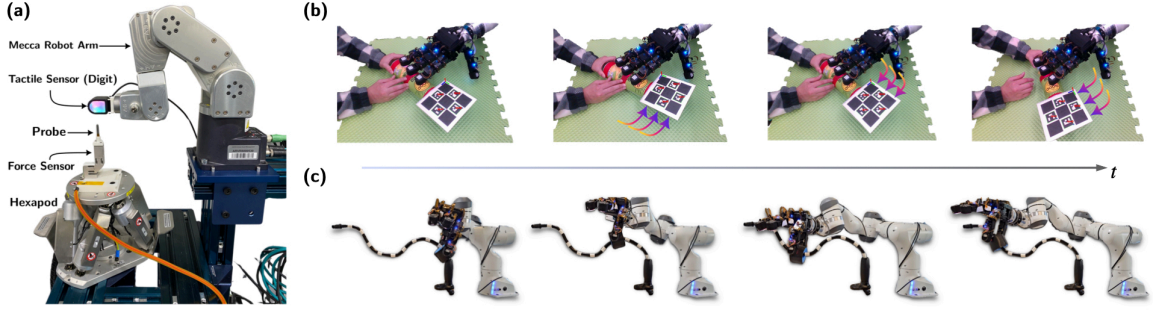


Figure 3.3: Real labeled data collection setup for **TacBench** tasks (a) [T1] Force estimation and [T2] Slip detection, (b) [T3] Pose estimation, and (c) [T6] Bead maze.

Task	Dataset	Sensor	Size	Collector	Label
[T1] Force estimation	Shear load	DIGIT	75k	Robot	3-axis force
	(indenter: sphere, flat, sharp)	GelSight Mini	75k	Robot	3-axis force
[T2] Slip detection	Shear load (indenter: sphere)	DIGIT	125k	Robot	Friction cone
[T3] Pose estimation	Object sliding	DIGIT	49k	Human	Object pose $\mathbf{SE}(2)$
[T4] Grasp stability	Feeling of Success [20]	GelSight 2017	9.3k	Robot	Success (yes/no)
[T5] Textile recognition	Clothing Dataset [210]	GelSight 2017	120k	Robot	Textile ID
[T6] Bead maze	Demonstrations	DIGIT	34k	Human	Joint angles

Table 3.4: Datasets in **TacBench** for evaluating representations on downstream tasks.

Evaluation protocol. We adopt a frozen evaluation procedure with an encoder-decoder architecture. Specifically, we *freeze* the pre-trained **Sparsh** encoder weights and train the parameters of an attentive decoder [10, 25] to assess what touch representations have captured from self-supervised pre-training alone. All tasks in the benchmark, except force field visualization and policy learning, train an attentive decoder containing a cross-attention module and a two-layer MLP using labeled datasets from Table 4.2. For tasks that require dense reasoning, we use DPT decoders [154] to decode patch representations into full input resolution quantities such as normal and shear force fields, and reconstructed tactile images. See Fig. 3.4 for a visual illustration of the probe architectures.

We also include an end-to-end (E2E) baseline with identical model capacity where the same encoder and decoder probe are initialized with random weights and all the parameters

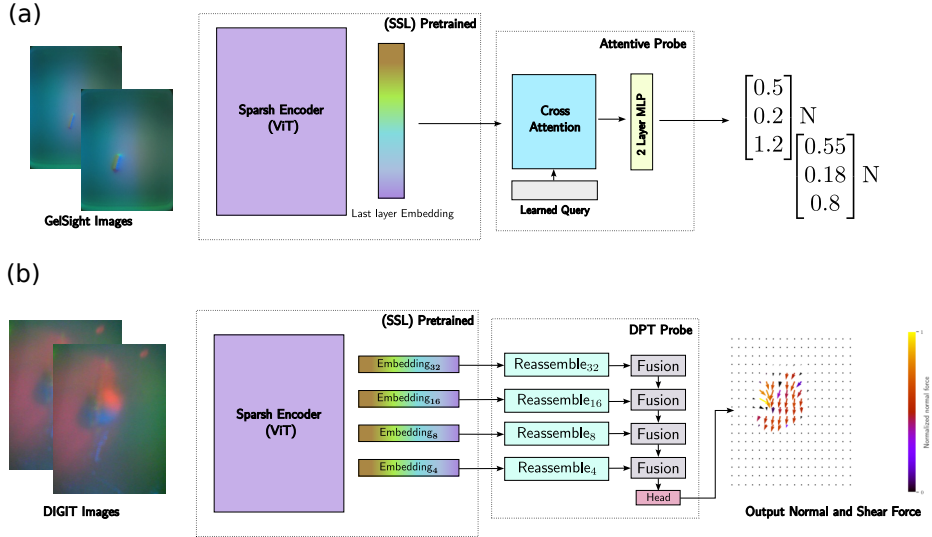


Figure 3.4: (a) Attentive probe architecture consists of a cross attention layer followed by a linear layer to regress *resultant* output quantities such as resultant force or slip state (b) Dense Prediction Transformer (DPT) [154] consists of multiple reassemble and fusion layers to decode features from intermediate layers of the Sparsh backbone to produce dense outputs such as normal and shear fields.

(both encoder and decoder) are trained. Further, we train downstream decoders with different amounts of labeled data to evaluate task performance under progressively low labeled data regimes.

In the following sections, we describe the design, metrics and results of each task in TacBench.

3.4 Comprehending Tactile Properties

3.4.1 [T1] Force estimation

Task. Force estimation is defined as the prediction of 3-axis normal and shear forces applied on the sensor’s elastomer. Fig. 3.3(a) shows our data collection setup. We use three different indenter shapes to collect force-labeled data: hemisphere, sharp, and flat. Our dataset contains 75k time-aligned samples of 3-axis force measurements, end-effector poses, and tactile images from DIGIT at 60fps and GelSight at 25fps. We train the decoder

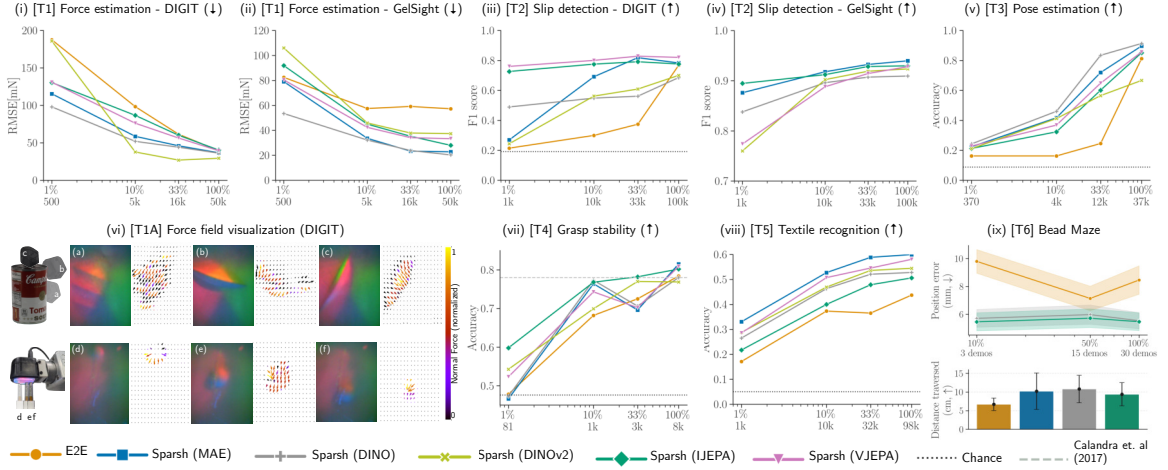


Figure 3.5: Summary of results comparing Sparsh and E2E on [T1]-[T6] tasks in TacBench across varying amounts of labeled data. Pre-training with SSL yields general touch representations that work across several tasks and sensors outperforming task and sensor specific models particularly under limited labeled data budget.

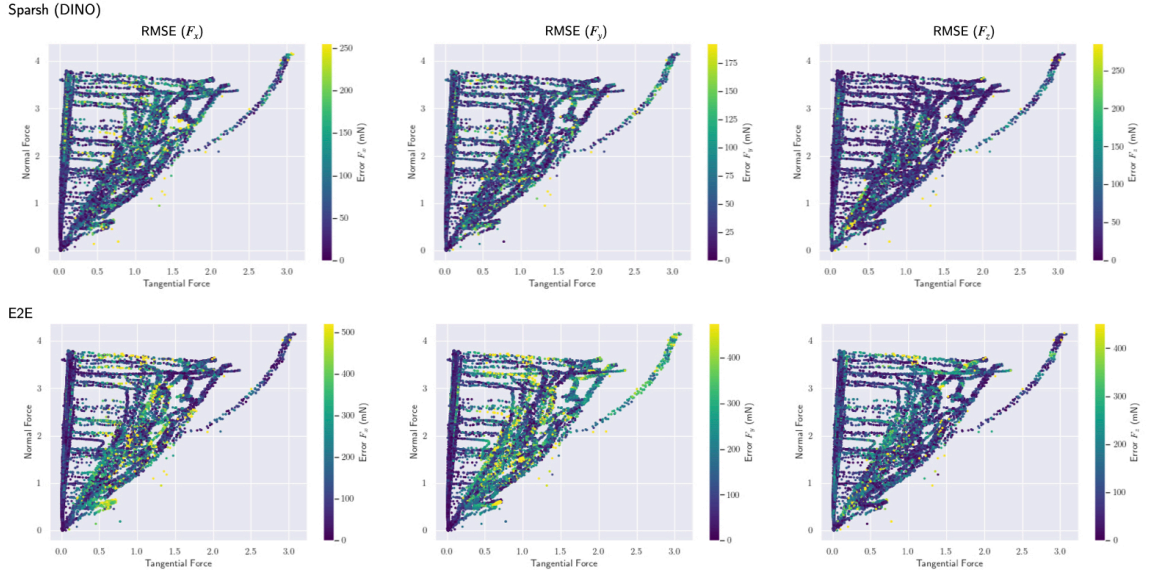


Figure 3.6: Friction cone of test data and RMSE (mN) for force estimation task with DIGIT sensor.

using normalized force measurements scaled between $[-1, 1]$, supervised using $\mathbf{L}_1$ loss and optimized using Adam until convergence. We compare performance using the average root

Model	Full dataset (50k)	1/3 dataset	1/10 dataset	1/100 dataset
E2E	39.34	61.42	98.22	187.51
	[39.21, 39.48]	[61.12, 61.72]	[97.61, 98.84]	[185.51, 188.51]
Sparsh (MAE)	36.61	45.96	58.55	115.39
	[36.51, 36.71]	[45.80, 46.12]	[58.31, 58.79]	[114.69, 116.09]
Sparsh (DINO)	36.09	44.03	51.89	97.95
	[36.01, 36.17]	[43.87, 44.19]	[51.69, 52.10]	[97.36, 98.52]
Sparsh (DINOv2)	29.31	26.85	37.66	185.86
	[29.14, 29.46]	[26.70, 26.99]	[37.45, 37.86]	[184.94, 186.78]
Sparsh (IJEPA)	40.27	60.04	86.57	130.37
	[40.16, 40.38]	[59.72, 60.34]	[86.06, 87.08]	[129.59, 131.15]
Sparsh (VJEPA)	39.38	56.34	76.11	130.83
	[39.30, 39.47]	[56.07, 56.62]	[75.67, 76.55]	[130.29, 131.38]

Table 3.5: Root Mean Squared Error (mN) and 95% confidence interval for force estimation with DIGIT data. All models were evaluated on flat indenter data over 25k test samples.

mean squared error (RMSE) across all three axes.

Results. GelSight Mini images are of higher resolution (HD) compared to DIGIT (320×240) resulting in smaller contact regions against the background. For this reason, we observe that when sufficient supervised data is available for DIGIT, it is possible to train a model from scratch to achieve high accuracy, but for GelSight Mini the end-to-end model does not perform well. Fig. 3.5 (i)-(ii) shows across the board that our frozen Sparsh representations can estimate forces with low error. Specifically, we find Sparsh (DINO) to be robust even when access to labeled data is sparse, a common scenario in tactile sensing.

We compute the average RMSE over 25k samples of unseen DIGIT data in Table 3.5 and for GelSight in Table 3.6. We report metrics for each Sparsh model and the E2E approach, under four different budgets of training data. We also provide a 95% confidence interval to ground the error ranges of each model. In Fig. 3.6 we plot the friction cone from the

Model	Full dataset	1/3 dataset	1/10 dataset	1/100 dataset
E2E	57.21 [56.44, 57.98]	59.09 [58.15, 60.04]	57.43 [56.44, 58.42]	82.42 [80.98, 83.86]
Sparsh (MAE)	22.72 [22.27, 23.17]	23.28 [22.83, 23.72]	33.56 [33.04, 34.08]	78.98 [77.74, 80.21]
Sparsh (DINO)	20.25 [19.85, 20.65]	23.79 [23.40, 24.18]	32.17 [31.67, 32.67]	53.43 [52.69, 54.17]
Sparsh (DINOv2)	37.30 [36.71, 37.88]	37.79 [37.22, 38.37]	45.86 [45.14, 46.59]	105.95 [104.28, 107.62]
Sparsh (IJEPA)	27.91 [27.37, 28.44]	35.20 [24.57, 35.82]	44.93 [44.13, 45.73]	91.81 [90.76, 92.86]
Sparsh (VJEPA)	33.26 [32.67, 33.84]	34.07 [33.39, 34.75]	42.35 [41.60, 43.10]	80.36 [79.26, 81.47]

Table 3.6: Root Mean Squared Error (mN) and 95% confidence interval for force estimation with GelSight Mini data. All models were evaluated on flat indenter data over 25k test samples.

test DIGIT data, where the colormap represents the error in mN for each axis. Note that E2E exhibit larger errors (around 500mN) for the tangential component and they are more predominant as the normal force increases. In contrast, the top model Sparsh (DINOv2) estimates with low error ($< 100\text{mN}$) in general across the whole range of tangential and normal forces.

3.4.2 [T1A] Force field visualization

We qualitatively evaluate the representations for rendering normal and shear force fields to understand sensor-object interactions. Although obtaining a shear field for sensors with markers nowadays is trivial via marker tracking [43], it is challenging and underexplored for markerless sensors.

Depth estimation and optical flow are analogous to the estimation of normal and shear

force fields, areas where the computer vision community has proposed several unsupervised methodologies [167, 94, 122, 95, 67]. We borrow ideas of unsupervised monocular depth estimation, where from two tactile images I_t and I_{t-n} , we learn a pose estimator for getting the transform between frames. With the sensor intrinsic K , we map image I_t from pixel space to camera plane, translate estimated depth D_t , apply transform from t to $t - n$, and transform back to image plane to get $\hat{I}_{t-n}$. We supervised based on the reprojection error, MSE between I_{t-n} and predicted $\hat{I}_{t-n}$. To reconstruct the shear field, we transfer ideas from unsupervised optical flow, where we warp the features of image I_t to I_{t-n} based on the estimated flow and compute a photometric consistency loss that encourages the estimated flow (shear) to align image patches with a similar appearance. This loss is a linear combination of the Charbonnier loss and the structural similarity (SSIM) between I_{t-n} and $\hat{I}_{t-n}$. We also add a smoothness loss that acts as a regularization term, encouraging the shear field to align the boundaries with the visual edges in the tactile image.

We train a CNN decoder using the reassemble-fusion approach for dense predictions [153] unsupervised, since we do not have access to ground truth for markerless sensors. We frame normal field estimation as depth estimation [67] and shear field estimation as optical flow [167, 94, 122, 95]. Fig. 3.5 (vi) shows visualizations for the top-performing model **Sparsh** (DINO) in [T1] that provides directional information about the relative motion of the contact patch. For instance, sliding motion (a, c, e, f), torsional slip (b), and divergence field upon contact (d). Refer to Fig. 3.7 for a higher resolution view of the snapshots.

3.4.3 [T2] Slip detection

Task. Shear and slip are closely related. Using the same setup as force estimation, we collect strokes where a hemispherical probe slides over the sensor, producing trajectories with both sticking and slipping samples. Slip is labeled using the friction cone model with an empirically estimated static friction coefficient (see Fig. 3.8). The dataset, with a notable imbalance between no-slip and slip classes, contains 125k samples with 13% slip instances. We train two decoders: one for slip detection and another for normalized force changes (Δ) as we find that predicting the two correlated quantities jointly enhances slip detection.

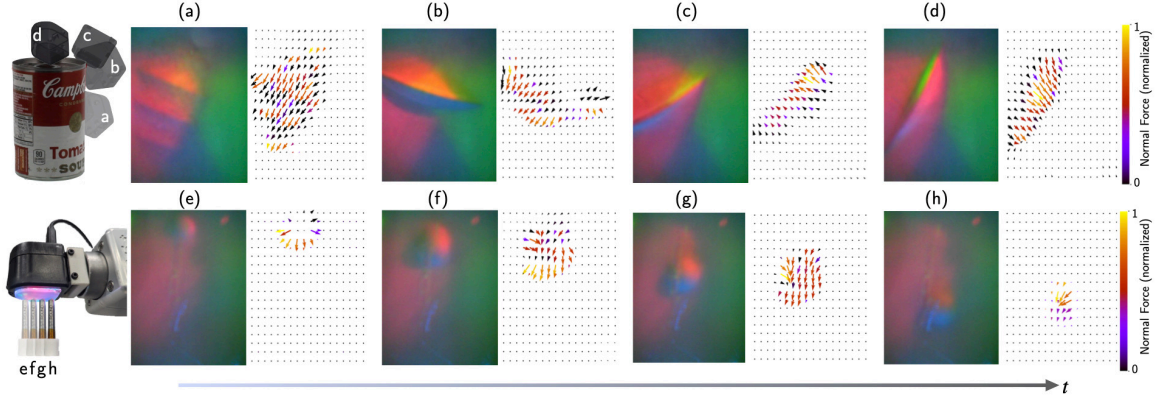


Figure 3.7: Normalized tactile flow (unitless) visualizations using **Sparsh** (DINO). Top row shows predicted force field for four key-frames from a representative YCB-Slide trajectory and bottom row shows interaction with the spherical probe. Arrows represent the tangential forces, while the colors depict the normal forces. These visualizations provide directional information about the relative motion of the contact patch. For instance (a) shows torsional motion resulting from rotating along the edge, (b, c, d) show sliding on the edge, (e) shows a diverging field when making contact with a spherical probe, and (f, g, h) show forces produced by sliding the probe top-down.

The MLP decoders use cross-entropy for slip detection and mean absolute error for Δ force regression, reserving 25k samples for evaluation.

Results. Table 3.7 shows F1 score instead of accuracy due to the imbalance in the slip labels in the dataset. Fig. 3.5 (iii)-(iv) illustrates the advantages of frozen **Sparsh** features trained under a JEPA paradigm for slip detection, particularly challenging for DIGIT sensor, even when using only 1% of the training dataset. In particular, **Sparsh** (VJEPA) achieves the highest F1 score among the models. Although all models detect slip from the 80 ms history of tactile data, **Sparsh** (VJEPA) benefits from a detailed temporal perspective, as its encoder processes a video clip with four frames spanning this window. **Sparsh** backbones also show better performance than the E2E model when labeled training data is significantly reduced.

In Fig. 3.9 we contrast the predictions over time for a sample trajectory between **Sparsh** (VJEPA) and E2E models trained with 33% of the data. Note that for **Sparsh** (VJEPA) the

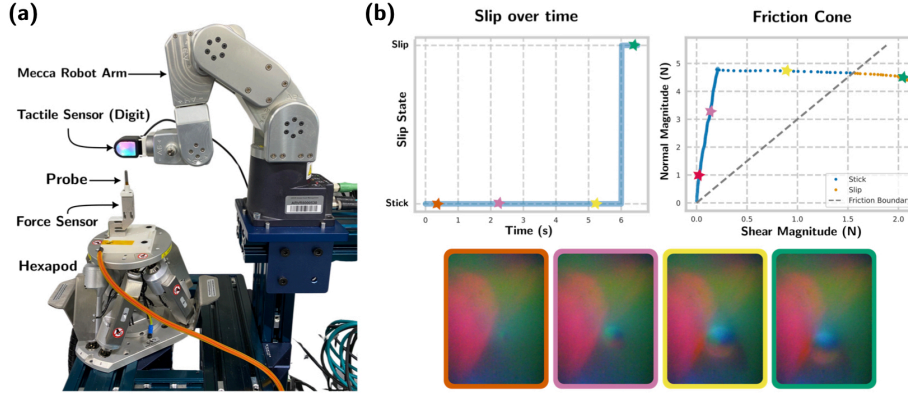


Figure 3.8: (a) Data collection setup for [T1] **Force Estimation** and [T2] **Slip Detection**. The Mecca Robot Arm with DIGIT / Gelsight is pressed against a static probe with random normal force. The arm then slides the sensor over the probe which induces shear forces. (b) Slip states over one representative stroke. When the sensor is pressed against the probe the normal force increases. The gel sensor initially resists sliding due to friction, but gives in, which results in a slight drop in normal force while the magnitude of shear force increases.

Model	Full dataset	1/3 dataset	1/10 dataset	1/100 dataset
E2E	0.767	0.238	0.299	0.214
Sparsh (MAE)	0.783	0.818	0.691	0.269
Sparsh (DINO)	0.685	0.561	0.548	0.489
Sparsh (DINOv2)	0.687	0.601	0.561	0.243
Sparsh (IJEPA)	0.776	0.791	0.775	0.726
Sparsh (VJEPA)	0.820	0.828	0.800	0.760

Table 3.7: Performance of models on slip detection task under different budgets of training data. We use F1 score as metric, given that it ensures the model accurately identifies slip events without favoring the majority class. A high F1 score indicates effective and reliable slip detection.

errors are around the friction boundary, where the probe is starting to slide. Also, it is worth noticing that a poor estimation of changes in shear and normal forces is reflected in

the accuracy of distinguishing between slip and no-slip. In Fig. 3.10, we illustrate a failure case for **Sparsh** (VJEPA), as its results do not align with the ground truth. However, it is important to note that slip labeling is prone to errors due to its reliance on an experimental coefficient of friction. Despite the inaccuracies in the friction boundary for this trajectory, **Sparsh** (VJEPA) successfully detects the slip samples.

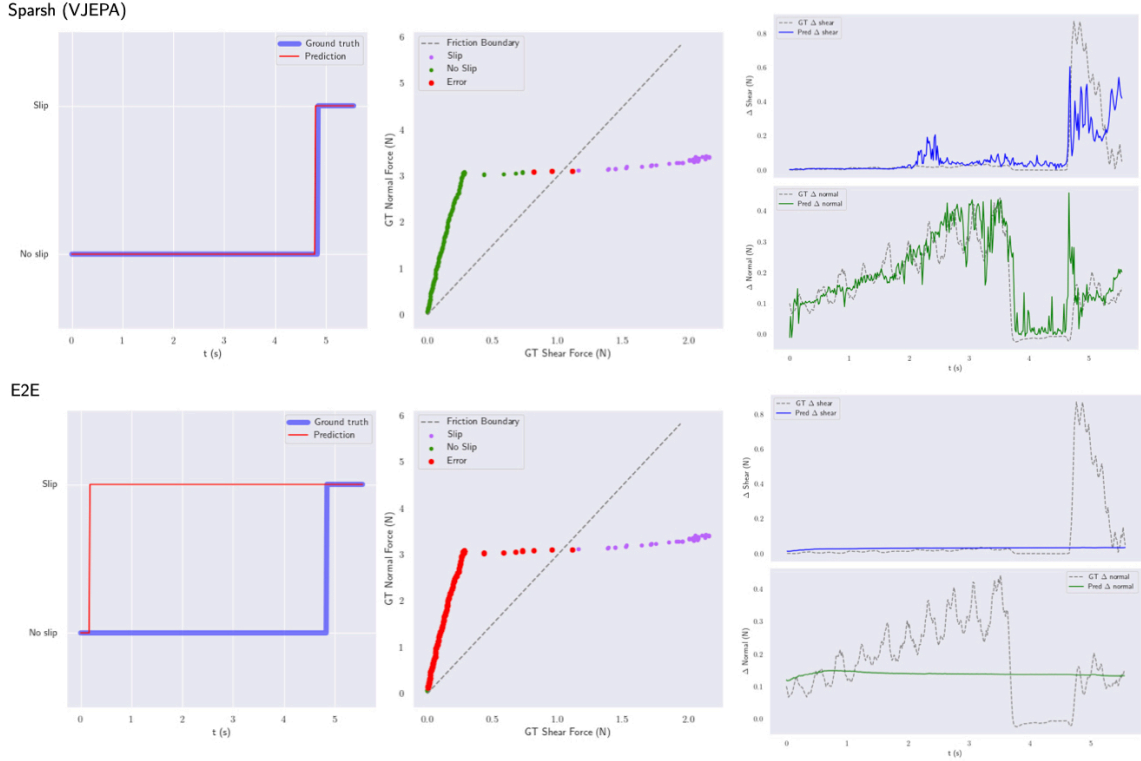


Figure 3.9: Contrast between **Sparsh** (VJEPA) and **E2E** for a test trajectory with a spherical probe sliding on the DIGIT sensor. **Sparsh** (VJEPA), even though trained only on 33% of the data, can detect slip accurately, correlated with its ability to estimate changes in normal and shear forces.

3.5 Enabling Physical Perception

3.5.1 [T3] Pose estimation

Task. Estimating object pose changes can help tasks such as tracking object drift for in-hand translation [100], rotation [149, 199], and pushing [124], among others. Given that

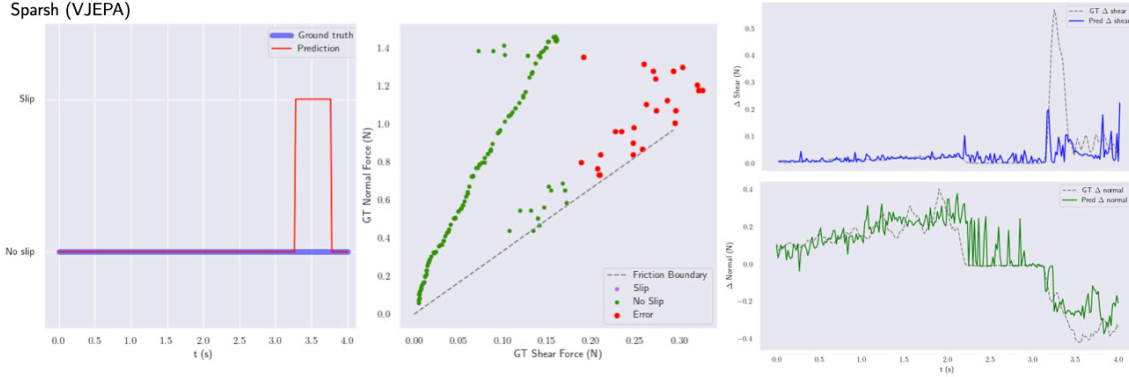


Figure 3.10: Failure case where the ground truth does not reflect slip since it relies on an experimental coefficient of friction. Despite the inaccuracies in the friction boundary for this trajectory, **Sparsh** (VJEPA) successfully detects slip samples.

Model	Full dataset	1/3 dataset	1/10 dataset	1/100 dataset
E2E	0.812 [0.811, 0.813]	0.245 [0.244, 0.247]	0.162 [0.160, 0.164]	0.162 [0.160, 0.164]
Sparsh (MAE)	0.896 [0.896, 0.897]	0.719 [0.718, 0.721]	0.417 [0.414, 0.420]	0.223 [0.221, 0.225]
Sparsh (DINO)	0.913 [0.912, 0.914]	0.834 [0.832, 0.836]	0.460 [0.457, 0.461]	0.242 [0.240, 0.245]
Sparsh (DINOv2)	0.665 [0.658, 0.673]	0.565 [0.559, 0.570]	0.411 [0.408, 0.415]	0.210 [0.209, 0.211]
Sparsh (IJEPA)	0.851 [0.850, 0.852]	0.601 [0.599, 0.603]	0.323 [0.321, 0.325]	0.212 [0.210, 0.215]
Sparsh (VJEPA)	0.856 [0.854, 0.857]	0.648 [0.646, 0.651]	0.368 [0.367, 0.370]	0.228 [0.225, 0.231]

Table 3.8: Accuracy and 95% confidence interval for pose estimation task following the regression-by-classification paradigm. Relative pose between object and ring finger. Metrics computed over 630 test samples.

tactile images capture local changes between sensor-object, we evaluate **Sparsh** representations to estimate $\mathbf{SE}(2)$ transformations of the object relative to the sensor. Fig. 3.3 (b) illustrates the data collection procedure. The dataset consists of time-synchronized pairs of DIGIT observations $\mathbf{z}_t \in \mathbb{R}^{h \times w \times 3}$ and object poses $\mathbf{T}_t \in \mathbf{SE}(3)$. $\mathbf{T}_t$ are then preprocessed to produce relative pose changes on the sensor gel as $\mathbf{S}_t^{t-1} \triangleq (\Delta x, \Delta y, \Delta \theta) \in \mathbf{SE}(2)$. We follow the regression-by-classification paradigm for this task [11, 100]. Relative object poses are binned into a grid, capturing translations with a resolution of $\pm 5\text{mm}$ and rotations with a resolution of $\pm 2^\circ$. For each degree-of-freedom (DOF), we train a head to predict probability distribution over the discretized grid using cross-entropy loss and Adam optimizer.

Results. Multiclass accuracy (Table 3.8) reveals that E2E approaches perform well with ample data, but drastically decline when labeled data is reduced, as shown in Fig. 3.5 (v). Fig. 3.11 shows a test trajectory over time with its ground truth labels. The colors on the plot represent the class agreement between the pose decoders trained with **Sparsh** (DINO) (using 33% of the data) and the ground truth. Darker colors indicate no error, while brighter colors indicate greater misclassification. Small datasets make it difficult to distinguish between close categories, such as orientation changes from $[0.5^\circ, 1.0^\circ]$ to $[1.0^\circ, 2.0^\circ]$. Pre-trained representations, however, maintain good performance even with only a third of the data. In low data scenarios, decoders using tactile representations often revert to extreme values, reducing estimation resolution and accuracy.

3.5.2 [T4] Grasp stability

Task. Grasp stability is well-studied in the tactile sensing literature for parallel jaw grippers [197, 96, 163, 104]. We evaluate whether representations aid in predicting grasp success given a short history of tactile images from a single finger. Specifically, we take inspiration from [20] and adapt the Feeling of Success dataset. Each sample consists of a triplet of tactile images corresponding to ‘before’, ‘during’, and ‘after’ grasping a set of objects. The dataset consist of 64% successful grasps and 36% failed grasps. We pass to the SSL model the ‘before’ and ‘during’ as tactile history. Since [20] does not specify an official train/test split, we create our randomized split with all objects, using approximately 8k grasps for

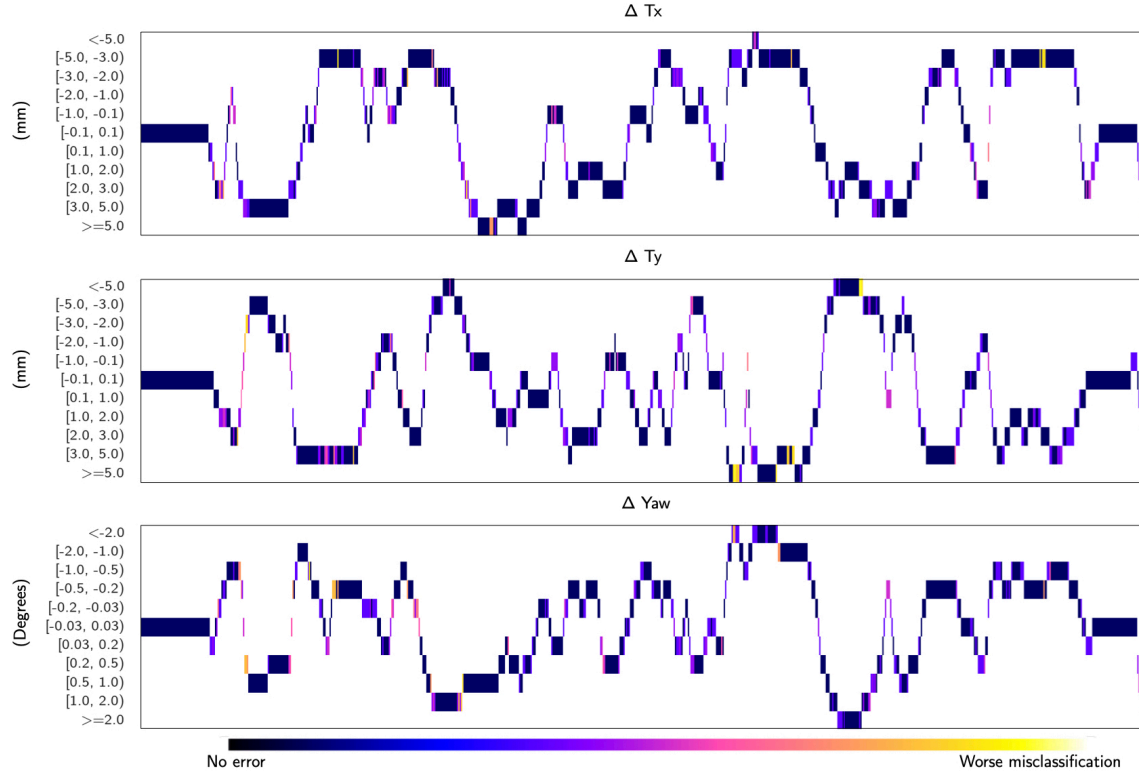


Figure 3.11: Ground truth relative pose classes for T_x , T_y , and Yaw for a test trajectory. The colormap represents the class agreements between the ground truth and the pose decoder, with darker colors indicating no error and brighter colors indicating greater misclassification.

training and the remaining 1.3k grasps for evaluation.

Results. Training with the full dataset, all models achieve similar accuracy ([Table 3.9](#)). **Sparsh** (IJEPA) or **Sparsh** (VJEPA) reach $\sim 80\%$ classification accuracy, surpassing results from [20] that combined tactile and vision modalities as shown in [Fig. 3.5](#) (vii). Our model, relying solely on touch from a single finger, shows competitive performance even with only 33% and 10% of the data. However, with just 80 training samples, performance drops significantly.

Model	Full dataset	1/3 dataset	1/10 dataset	1/100 dataset
E2E	0.784 [0.783, 0.785]	0.725 [0.722, 0.728]	0.682 [0.680, 0.684]	0.478 [0.472, 0.482]
Sparsh (MAE)	0.815 [0.813, 0.817]	0.696 [0.691, 0.702]	0.764 [0.761, 0.768]	0.466 [0.461, 0.471]
Sparsh (DINO)	0.780 [0.777, 0.782]	0.706 [0.702, 0.710]	0.773 [0.772, 0.775]	0.473 [0.467, 0.479]
Sparsh (DINOv2)	0.770 [0.767, 0.771]	0.770 [0.768, 0.772]	0.699 [0.697, 0.701]	0.543 [0.539, 0.546]
Sparsh (IJEPA)	0.802 [0.800, 0.804]	0.782 [0.779, 0.784]	0.768 [0.766, 0.770]	0.598 [0.597, 0.601]
Sparsh (VJEPA)	0.809 [0.805, 0.813]	0.702 [0.700, 0.704]	0.743 [0.740, 0.746]	0.523 [0.519, 0.527]

Table 3.9: Accuracy and 95% confidence interval for grasp stability classification over different budget sizes of training data, using Feeling of Success dataset. Results over 1.3k grasps.

3.5.3 [T5] Textile recognition

Task. Vision-based tactile sensors are broadly used for material property recognition, since their compliant gel and high resolution cameras make them effective at discriminating different materials by surface texture [210, 55]. Specifically, we take the task definition from [210] and adapt the Clothing Dataset. The dataset consist of 4467 short video clips (10-25 frames), of a robot with a GelSight 2017 (markers) grasping several types of textile (20 classes), such as leather, cotton, polyester, etc. We follow the train-test split provided in the metadata of the dataset.

Results. As shown in Table 3.10, training an E2E specialist model for textile recognition using the full dataset can be challenging, as noted in [210]. By leveraging pre-trained

touch representations, as shown in Figure 3.5 (viii) the performance of the task can be significantly improved, even when training with only 10% of the labeled data. **Sparsh** (MAE) is particularly effective, as it heavily relies on pixel-level features.

Model	Full dataset	1/3 dataset	1/10 dataset	1/100 dataset
E2E	0.437	0.365	0.373	0.171
Sparsh (MAE)	0.599	0.588	0.527	0.330
Sparsh (DINO)	0.527	0.520	0.463	0.264
Sparsh (DINOv2)	0.544	0.536	0.469	0.288
Sparsh (IJEPA)	0.506	0.478	0.399	0.217
Sparsh (VJEPA)	0.580	0.545	0.507	0.285

Table 3.10: Accuracy for textile classification over 20 classes using GelSight with markers under different budget of labeled data. Results over 26k tactile images, where accuracy of chance is 0.05.

Additionally, we evaluate the cross-sensory ability of the representations. Since **Sparsh** is trained on multiple GelSight-like data, we investigate whether SSL training enables cross-sensory representations or if it helps downstream tasks trained for one sensor quickly adapt to another. To study this, we use as a baseline the decoder trained for [T5] Textile recognition, which was supervised with labeled data from GelSight with markers.

We collect new data using a DIGIT sensor for 10 out of the 20 textiles. Our dataset contains 11 samples for both training and testing. We load the trained decoders for [T5] using **Sparsh** (DINO) and E2E and perform zero-shot evaluation as well as 1-shot, 5-shot, and 10-shot training and subsequent evaluation using the DIGIT data. Table 3.11 reports accuracy on 110 samples of test data. Zero-shot evaluation with DIGIT performs close to chance, while with very few samples (10-shot) **Sparsh** (DINO) classifier quickly adapts and significantly outperforms E2E. This experiment empirically demonstrates the value of cross-sensor representations.

	zero-shot	1-shot	5-shot	10-shot
Sparsh (DINO)	9.1	19.1	28.2	61.8
E2E	3.6	0.0	15.5	10.9

Table 3.11: Accuracy of n -shot evaluation of [T5] Textile recognition on DIGIT data to study how **Sparsh** facilitates cross-sensory adaptation to the downstream task.

3.6 Enabling Manipulation Planning

3.6.1 [T6] Bead maze

Task. The bead maze is a children’s toy to enhance fine motor skills. We adapt this task to robot policy learning, where the goal is to guide a bead from one end to another following the path of the wire (maze). Given a small history of tactile images $(\dots, \mathbf{z}_{t-1}, \mathbf{z}_t)$, and robot proprioception $(\dots, q_{t-1}, q_t)$, we train a diffusion policy [28] to predict changes in joint angles as actions $\mathbf{a} \triangleq (\Delta q_t, \Delta q_{t+1}, \dots)$; $\Delta q \in \mathbb{R}^7$, to make progress on this task. This task is fundamentally tactile-focused, as the robot needs to react to resistance encountered by changes in the maze pattern and the subtle local movement of the bead are difficult to perceive from vision even when not occluded by the hand. A prior version of the bead maze task has been explored in robotics relying solely on tactile feedback [89]. In our setup, illustrated in Figure 3.3 (c), we assume that the robot starts with an initial stable grasp. We collect a dataset of 50 demonstrations on different maze patterns with a mix of VR-based and manual kinesthetic-based teleoperation, corresponding to a total of $\sim 34\text{k}$ training pairs of tactile images and robot joint angles.

Following the guidelines in Diffusion Policy, we use an observation horizon of 2 and an action prediction horizon of 8. We adhere to the official implementation for policy architecture and training hyper-parameters. For conditioning on tactile input, we modify the CNN encoder from Diffusion Policy and replace it with **Sparsh** backbones with fixed parameters. For training an end-to-end policy, the encoder corresponds to a ViT-Base encoder with randomly initialized weights.

Results. We evaluate **Sparsh** (DINO) and **Sparsh** (IJEPA) for policy learning, as these representations exhibit the best performance across the rest of the benchmark. For completeness, we also consider **Sparsh** (MAE), and E2E which trains a tactile encoder and policy end-to-end. Due to covariate shift [162] in behavior cloning, prediction errors can accumulate over time; therefore, we report position error between the predicted trajectory and a demonstration trajectory from a held-out maze sequence over small chunks of 3cm followed by the robot corresponding to 15 timesteps of action predictions. Figure 3.5 (ix) shows the position error over access to different number of demonstrations for training. We find that **Sparsh** (DINO) and **Sparsh** (IJEPA) produce significantly (a difference of $\sim 16\%$) lower trajectory errors compared to training the policy E2E.

Additionally, we evaluate real rollouts of the learned policies (using all 50 demonstrations) over a set of 10 randomized novel starting locations on the maze. In Table 3.12 and Fig. 3.5 (ix) we report distance traversed (in cm) before failure. We find that policies using **Sparsh** representations outperform E2E by $\sim 20\text{-}53\%$. We note that given the high precision nature of this task and the considerations for real system deployment for the policy, none of the models succeeds in completing the full maze on real robot rollouts. We expect that increasing the diversity of training data with different maze patterns will highlight the generalist capabilities of touch representations, and that temporal ensembling will aid in improving the smoothness of the policy [218].

(cm)	Sparsh (DINO)	Sparsh (IJEPA)	Sparsh (MAE)	E2E
Pre-trained	10.80 ± 3.68	9.4 ± 3.1	10.2 ± 4.9	6.70 ± 1.67
Fine-tuned	8.45 ± 3.21	10.02 ± 5.37	11.25 ± 3.85	6.70 ± 1.67

Table 3.12: Mean and variance of distance traversed (in cm) before failure for policies based on **Sparsh** and E2E. Results over 10 randomized novel starting locations on the bead maze.

3.7 Related Work

Self-Supervised Learning with its success in natural language processing and computer vision, has become the new learning paradigm. In the last three years, a variety of general-

purpose frameworks [69, 80, 23, 26] have been proposed for learning representations. We refer to [142, 9] for a comprehensive survey on SSL frameworks and their categorization based on pretext tasks and learning algorithms. In Appendix ??, we expand on Masked Image Modeling (MIM), self-distillation, and Joint-Embedding Predictive Architecture (JEPA), as we explore them in this study.

Traditionally, tactile sensing has relied on preprocessing tools like marker tracking and finite element method models to extract contact properties, such as shear forces [209, 174], dense normal estimation [128, 174], and contact area prediction [111]. From a learning perspective, a trend is to use custom encoder architectures tailored for specific tasks and sensors, which are either pre-trained or trained end-to-end [78, 30, 27, 193, 121, 86, 170]. Nevertheless, there is an increasing interest in representation learning for vision-based tactile sensors. For instance, MAE has shown effectiveness at material classification and texture recognition [21]. Fine-tuning convolutional encoders for BioTac, RoboSkin, and GelSight performs well on fabric decomposition tasks [212]. Even nearest-neighbor retrieval over pretrained representations, for the XELA [178] uskin sensor, can enable some success in dexterous manipulation [74]. Crucially, the current state of standardization in learning touch representations and the wide variety of tactile sensors available, has made it challenging to develop and share pre-trained models across the research community in this domain.

Another direction is exploring the alignment of visual and tactile modalities in latent space using multimodal datasets [198, 62, 61, 64, 20, 55] and techniques like contrastive coding and cross-sensory retrieval [197, 206, 38], yielding promising results in tasks like material classification, grasp stability, and tactile-driven image stylization. However, current approaches [197, 48, 99, 38] primarily focus on texture and visual properties and overlook physical contact properties, such as forces, slippage, and poses, which are essential for dexterous manipulation.

The works closest and concurrent to ours are T3 [217] and UniT [194]. T3 trains sensor-specific encoders to capture shared latent information through a shared trunk, using both the MAE objective and labeled task-specific data as supervision. UniT is a VQGAN [181] model with a patch-based discriminator for representation learning only for GelSight Mini (markers). On the other hand, we introduce a family of models trained with the latest SSL

algorithms for the three most commonly used families of tactile sensors: DIGIT, GelSight 2017, and GelSight Mini. Similar to T3 and UniT, we evaluate touch representations for policy learning, however we also introduce a standardized benchmark to comprehensively evaluate representations and their ability to solve several relevant touch-centric tasks along tactile properties, physical perception, and manipulation planning.

3.8 Discussion

Task	Best SSL vs E2E	DINO vs IJEPA	MAE vs Best	VJEPA vs Best
Force estimation (DIGIT)	28.31%	26.67%	−4.38%	−27.96%
Force estimation (GelSight)	59.74%	32.41%	1.72%	−64.23%
Slip detection	242.70%	29.08%	−1.21%	0.00%
Pose estimation	235.89%	−37.91%	−13.81%	−22.33%
Grasp stability	5.14%	8.45%	−10/17%	−7.83%
Bead maze	19.72%	−5.26%	-	-
<i>Average</i>	98.75%	8.91%	−5.57%	−24.47%

Table 3.13: Performance of **Sparsh** across TacBench and comparison between SSL approaches.

In [Table 3.13](#) we summarize the performance of **Sparsh** across the benchmark. We find that with respect to an E2E approach, with **Sparsh** we can achieve an improvement of 98.75% on average. **Sparsh** (DINO) and **Sparsh** (IJEPA) are in general the best models across the board, showing the benefits of learning touch representations in latent space. **Sparsh** (MAE), which relies on pixel space supervision, is still competitive, although it was not evaluated on the policy task.

Overall **Sparsh** excels on all tasks. In particular, we find **Sparsh** (DINO) is well suited for physics-based tasks like force and pose estimation, while **Sparsh** (IJEPA) performs better at touch semantic understanding like slip state, stability of a grasp, and textile recognition. On average **Sparsh** (DINO) outperforms **Sparsh** (IJEPA) by 5.6% across the benchmark. Both models perform similarly in bead maze test demonstrations, which require implicit knowledge of shear forces and slip. However, this did not translate to real robot performance due to lack of force control and system-level confounding variables not captured during training.

These include the high precision required to keep the bead in place, the impossibility of error recovery once grip is lost, and trajectory drift due to local decision-making. Specialist policies or models trained from scratch exhibiting better robot rollout performance is due to the narrow task domain setting that leads to overfitting, a trend similarly observed when studying pre-trained vision models [28, 218, 79].

Learning touch representations in latent space is more advantageous than in pixel space, as these representations can filter out and generalize over noise or lighting differences. Tasks traditionally challenging for markerless sensors (like DIGIT and GelSight Mini), such as shear force (and field) estimation and slip detection, become solvable with our general touch representations. On average, **Sparsh** achieves a 95.1% improvement compared to an end-to-end approach when all models have access to only 33% – 50% of the labeled dataset per downstream task. Using as little as 10% or 1% of the labeled data for force estimation and slip detection still yields acceptable results (e.g. force error below 0.1N with **Sparsh** (DINO)). Fine-tuning **Sparsh** encoders is another method of assessing the quality of pre-trained representations. We provide in Appendix A.1 experiments with partial and full fine-tuning. Notably, models pre-trained in latent space perform better in downstream tasks when fully fine-tuned, especially in regression tasks like force and pose estimation. In contrast, partial fine-tuning offers minor improvements, aligning closely with the performance of frozen models. We also evaluate **Sparsh** decreasing the model capacity, finding the biggest impact in performance for regression-like tasks when training with limited amount of labeled data (see Appendix A.1.1).

Sparsh is a significant step towards a general pre-trained backbone for vision-based tactile sensors. Our aim is to enable efforts to compile larger tactile datasets that include additional vision-based tactile sensors and leverage the benefits of scaling up SSL backbones, as seen in computer vision and natural language processing. **TacBench** serves as an initial benchmark for evaluating these representations, and additional tasks can be incorporated based on the needs of the tactile sensing and manipulation community. For instance, further exploration of pre-trained touch representations in tactile policy learning, or tracking dynamic object properties like changing mass during pouring.

Limitations. Open-source tactile datasets we considered in this study predominantly feature discrete contact interactions. We believe that incorporating data rich in shear interactions can further improve the representations. We do not ablate the length of tactile image history for learning the representations. Such ablations could provide guidance on improving their quality for downstream tasks. Our bead maze policies with pre-trained touch representations deployed on the real robot are only able to complete the maze partially before compounding error leads to the bead falling out of the fingers. Further research is needed to understand how to effectively leverage pre-trained touch representations in behavioral cloning for robot manipulation tasks.

Chapter 4

SELF-SUPERVISED PERCEPTION FOR TACTILE SKIN COVERED DEXTEROUS HANDS

4.1 Introduction

Touch is inconspicuous, but plays a crucial role in dexterous manipulation, like when playing the guitar or plugging a cord into a socket when vision is impaired. The robotics community has leveraged touch to enhance robot learning [149, 155, 87, 199], but has so far largely limited their attention to vision-based tactile sensing. Recently, sensors such as the DIGIT [107], GelSight [207], GelSlim [44] and others [155, 186], have gained popularity due to their high-resolution output, human-interpretable signals, and accessibility. Capturing touch as images is attractive, as advances in computer vision can be leveraged with minimal friction. Nevertheless, these sensors are slow compared to human skin’s touch receptors, come in bulky form factors precluding large area sensing, and are often custom-designed for specific manipulators [155], making reproducibility a challenge.

Magnetic skin-based sensors such as uSkin (Xela) [179, 180], ReSkin [12], and others [13, 145], offer an alternative for tactile feedback. They provide fast response times (~ 100 Hz), lower dimensionality and flexible form factors that can be adapted to complex embodiments, such as multifinger robot hands providing richer states for dexterous manipulation. Despite their potential, the widespread use of these sensors is primarily limited by their complexity: these sensors are difficult to interpret, difficult to model due to hysteresis and other factors, and are primarily hindered by a lack of infrastructure.

Self-supervised learning of general touch representations offers a potential solution: it can learn priors from unlabeled data, making subsequent learning on specific tasks (downstream learning) sample efficient. However, while previous research [74, 191], has applied self-supervision for tactile learning, these approaches often use techniques from computer vision such as treating temporal signals as images [74] and employing masked reconstruction

objectives [191, 80] that may be ill-suited for signals like noisy magnetic flux.

We present **Sparsh-skin**, a pre-trained tactile encoder model trained using self-supervised learning (SSL) for magnetic skin-like sensors covering a multifinger robot hand. **Sparsh-skin** directly learns in-hand contact priors from tactile history and hand configuration using a classification objective. Our tactile encoder simplifies downstream task use, by introducing standardized magnetic time-series data, and reducing the need for real-world labeled data, which is difficult to collect and oftentimes infeasible. For instance, we do not yet have hardware to annotate spatially distributed ground truth force fields. By combining our representation learning algorithm, tactile signal tokenization, and a fully-sensorized multi-fingered hand, we achieve state-of-the-art tactile representations for magnetic-skin sensors, outperforming end-to-end training by $\sim 56.37\%$ and prior works by $\sim 41.04\%$ on average in both performance and sample efficiency for downstream tasks.

4.1.1 Contributions

To summarize, this chapter makes the following key contributions:

- [Sections 4.2 and 4.3](#) extend the recipe for learning general representations for magnetic-skin based tactile sensors, following a self-distillation SSL paradigm.
- [Section 4.4](#) present a thorough evaluation of the representations for the Xela tactile sensors covering an Allegro robot hand.

4.1.2 Preliminaries on magnetic-skin tactile sensors

Tactile sensors can be broadly categorized into vision-based (e.g. DIGIT [107], GelSight [207], GelSlim [44] and others [155, 186]), pressure-based (e.g. force sensitive resistors), impedance-based (e.g. BioTac [54]), and magnetic-based (e.g. uSkin (Xela) [179, 180], ReSkin [12], and others [13, 145]) sensors. Vision-based sensors commonly used in robot manipulation capture finger-object-environment interactions as images [107, 207]. However, their bulky form factor, low-frequency feedback and high bandwidth requirement limit their application in tasks that require large areas coverage. Impedance-based tactile sensors offer high temporal

resolution, but are difficult to interpret, and currently do not provide full-hand coverage solutions either. Pressure-based sensors can offer a wide coverage area, but lacks capabilities in shear force sensing.

Magnetic tactile sensors, on the other hand, provide a thin skin-like alternative with options such as ReSkin [12], AnySkin [13], and Xela [179, 180] being popular choices. They provide low-dimensional but high-frequency signals. However, when these sensor pads are distributed on all contact interfaces of a robot hand, the total output is high-dimensional. These sensors primarily use hall-effect sensing for force measurement. Xela [180] in particular works by transducing displacements of permanent magnets embedded in an elastomer arranged in a grid pattern to magnetic flux changes, essentially capturing 3-axis shear and normal forces. ReSkin and AnySkin [12, 13] magnetize the entire elastomer layer continuously, instead of using discrete magnets. This sensing modality has been explored for various contact-rich applications, including planar pushing [118], surface material classification [188], grasp stability [59], and policy learning tasks [117, 144].

4.2 Self-Supervised Representations for Tactile Skins

Sparsh-skin is a self-supervised learning (SSL) framework designed to extract generalizable tactile features from unstructured "random-play" data. Specifically tailored for dexterous hands equipped with magnetic-skin tactile sensors, the model adopts a self-distillation paradigm. This choice is informed by the performance benchmarks of **Sparsh** —as detailed in the preceding chapter—which demonstrated the efficacy of distillation in similar high-dimensional contexts.

However, a critical distinction lies in the modality of the input signal. Unlike the image-based inputs typical of vision-centric SSL, raw data from magnetic skins is fundamentally temporal. Consequently, we adapt the tokenization scheme to better capture these temporal correlations, ensuring both an efficient representation of tactile dynamics and a compatible interface for the Transformer architecture.

4.2.1 Robot setup and data collection

Our setup consists of the Allegro hand sensorized with Xela uSkin, attached to a Franka Panda robot arm. The Allegro hand is equipped with 18 Xela uSkin sensing pads, consisting of 4 curved fingertip sensors with 30 individual sensors, 11 (4×4) grid sensors pads attached to the finger phalanges, and 3 (4×6) sensing pads attached to the palm, resulting in a total of 368 individual sensors.

We collected a dataset of the hand performing various atomic manipulation actions with 14 household objects and toys, including squeeze, slide, rotation, pick-and-drop, circumduction, pressing, wiping, and articulation. Using a VR-based teleoperation system with Meta Quest 3, which builds on the inverse kinematics-based re-targeting method proposed in [77], we recorded 11 sequences (approximately 2 minutes each) for each object, totaling around 4 hours of varied interactions. The dataset includes top/left camera views, Xela signals, and robot and hand joint states, covering a range of rigid and deformable objects with diverse tactile properties.

4.2.2 Sensor Tokenization

We perform baseline subtraction on Xela signals to account for their uncalibrated nature and consistent biases. A single baseline signal is collected with the Allegro hand in a resting configuration (palm up and flat) and used for all downstream tasks, unlike prior work [12, 144], which collects a new baseline signal per training sequence. We also resample Xela signals to a consistent 100Hz frequency. as the sensor data rate fluctuates between 80Hz to 100Hz, unlike prior work [74] that subsamples data to match modalities at lower frequencies.

We note that for representation learning, tactile data can be temporally correlated, and instantaneous signals cannot provide context for contact changes, therefore we choose to learn representations for chunks of 100ms of data. First, inputs to `Sparsh-skin` are formatted corresponding to a brief history of 0.1 seconds of the sensor signal $x_{1:10} \in \mathbb{R}^{10 \times 368 \times 3}$ concatenated with the history of sensor position $p_{1:10} \in \mathbb{R}^{10 \times 368 \times 3}$ computed from the forward kinematics of the Allegro hand. Inputs are then tokenized through a linear projection

f_{linear} to the dimension d of the representation $z_i = f_{\text{linear}}(x_{1:10}|p_{1:10}) \in \mathbb{R}^{368 \times d}$. Finally, a learnable token is added to each sensor according to the three types of Xela sensing pads (see 4.2.1) on the Allegro hand. We do not add additional positional embedding and instead rely on the sensor position to provide 3D positional information to the transformer network.

4.2.3 Architecture and SSL training

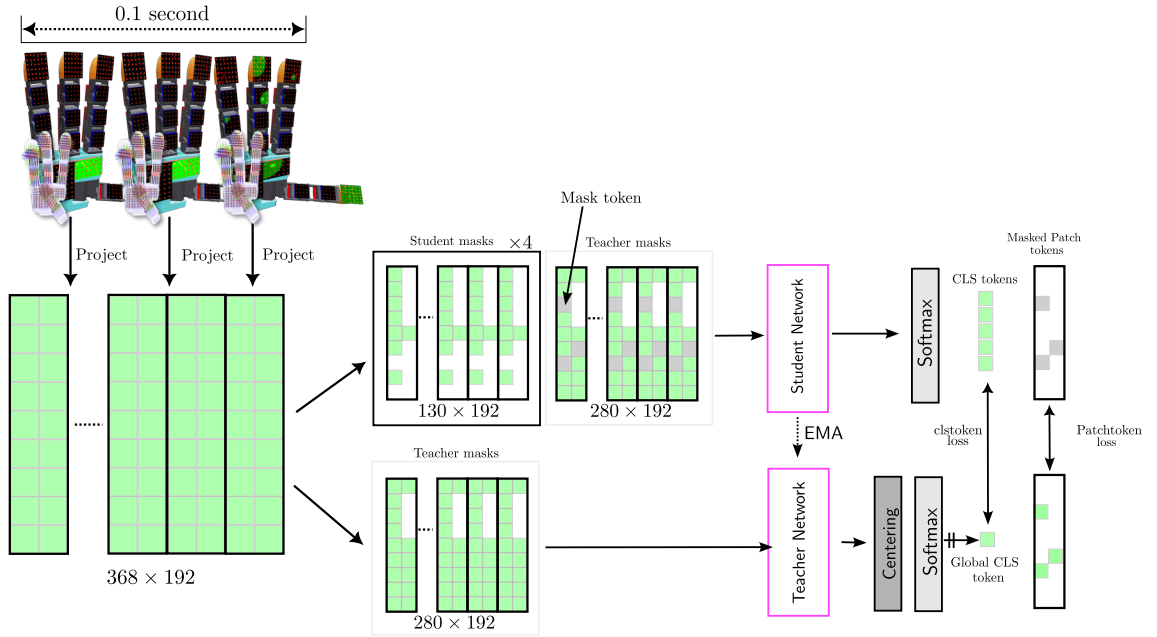


Figure 4.1: **Sparsh-skin** block diagram for self-supervised learning of skin representations. Our approach follows the student-teacher framework and loss functions used in self-distillation. However, we adapt the transformer input tokenization to accommodate time-series Xela data.

Our encoder model is a modified version of Vision Transformers [46]. Specifically, we adapt the tokenization of the time-series Xela with sensor pose data. After flattening the 3D-axis magnetic reading per magnetometer (368) and concatenating their corresponding pose in chunks of 0.1 second, the inputs $x \in \mathbb{R}^{10 \times 368 \times 6}$ are tokenized through a linear projection to the dimension d of the representation $f_{\text{linear}}(x) \in \mathbb{R}^{368 \times d}$. We use a tiny model with $d = 192$. We add a learnable embedding to identify different types of xela pads (palm, phalanges and fingertips). Then, we construct different cropped view of the data, two global views and

eight local views. We mask sensor data from contiguous blocks by removing those sensors from the input. For the local view we retain between 10% and 40% of the tactile signal, whereas for the global views we retain 40% to 100%. An illustration of the masking and diagram block of the pipeline for self-supervised learning of Xela representations is shown in Fig. 4.1.

The student and teacher share the same encoder and projector head architecture, both initialized with the same weights. The projector head corresponds to a 3-layer MLP with an output dimension of $k = 65536$. We use the projection head for the proxy prediction task to distill knowledge to match output distributions over k dimensions between student and teacher networks. The student network is updated via back-propagation, while the teacher network is updated at a lower frequency via exponential moving average (EMA) on the student weights. We pass the global and local views to the student encoder, while the teacher only has access to the global views. The register tokens from global/local views are passed through the projection head. For the teacher only, the output is also centered and sharpened via softmax normalization.

SSL prediction task Although cropping and resizing images is a common technique for signal corruption in the image domain, applying this method to magnetic flux readings alters the shear profile. Therefore to avoid any untoward data augmentation that changes the semantic meaning of the signal, we use block masking [6] to corrupt signals that are input to the encoding networks. Specifically, input data is masked after sensor tokenization in a cross-taxel manner i.e., given tokenized data from 368 sensors, we mask sensor data from local contiguous blocks including sensors even from neighboring sensor islands by removing those sensors from the input (see Fig. 4.2). The masked sensor tokens are subsequently transformed through the student and teacher network as $\mathbf{E}_\theta(\bar{z}_i)$ and $\mathbf{E}_{\hat{\theta}}(z_i^*)$ respectively.

For the prediction task, we use classification by defining a set of prototype classes as in [22, 140], which is robust to sensor noise compared to masked auto-reconstruction. The sensor tokens after transformation are converted into prototype logits through a classification head f_{class} as $\bar{p}_i$ and p_i^* respectively for the student and teacher networks. We use both the class token and the patch level cross entropy objective between the student and

teacher logit predictions to enforce local-to-global correspondence learning in the sensor representation.

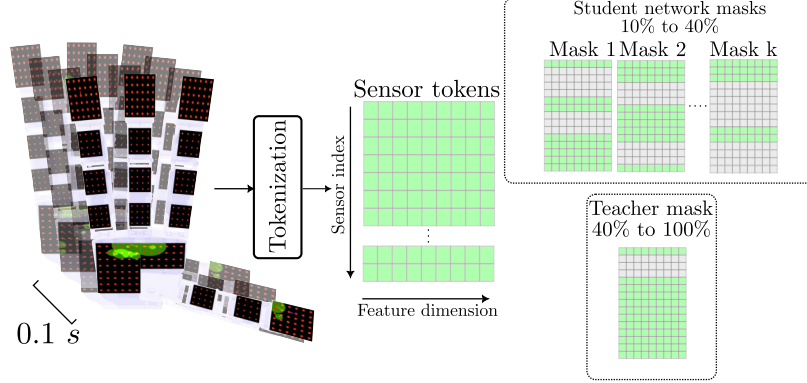


Figure 4.2: Illustration of Xela signal corruption via masking for SSL prediction task. Once a 100(ms) window of tactile measurements and sensor positions are tokenized, block masking is applied to corrupt the signal, . For each data sample, the student network receives k different masks, each randomly retaining 10% to 40% of the data denoted $\bar{z}_i$. The teacher network, in contrast receives 1-2 masks each retaining 40% to 100% of the data denoted z_i^* .

4.3 Training and Online Probing

We train **Sparsh-skin** on 8 Nvidia A-100 (80G) GPUs. To monitor learning, we use reconstruction online probe and classification via linear probing. We use AdamW optimizer and a linear rampup followed by a cosine schedule as the learning scheduler. Further, we find that tuning momentum value as well as the weight decay factor was important in observing training convergence. Additional information of hyperparameters is detailed in [Table 4.1](#).

We monitor task performance via a reconstruction task and a classification of the object class metadata from the play data. These tasks are trained simultaneously during SSL pretraining through auxiliary losses. The representations are fed into the respective decoders after a stop gradient operation to prevent the online probing tasks from affecting the SSL prediction task loss. For reconstruction we use a small 4-layer transformer as the decoder, while for multi-class classification we only use a linear layer.

Architecture	ViT-Tiny (adapted)
Embedding dim	192
EMA decay	[0.994, 1.0]
LR	1e-4
Batch size	64

Table 4.1: Training hyperparameters for **Sparsh-skin**. All models run for 500 epochs with optimizer AdamW, a weight decay cosine schedule from 0.04 to 0.4, and a learning rate warmup of 30 epochs.

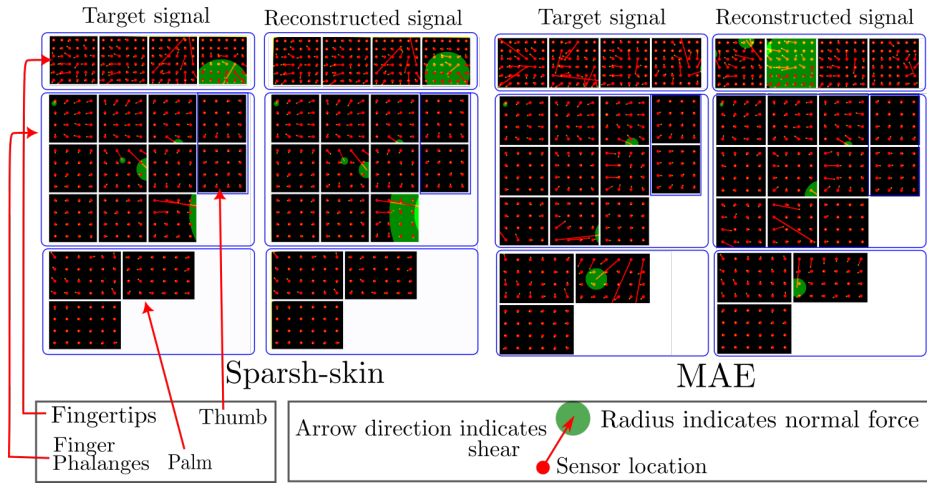


Figure 4.3: Visualization of reconstructions from online probe. Red dots indicate the canonical sensor positions on the robot hand. Normal force applied on the sensor directly correlates with the radius of green blobs over a sensor. Further, the offset of the green blob center, highlighted by red vectors from the canonical sensor position to the offset position, indicates the shear force applied on the sensors. When compared to MAE, **Sparsh-skin** reconstructs Xela signals effectively. Specifically, note that the normal forces and directions are better preserved by **Sparsh-skin**. Here, we visualize a single frame from a 0.1s tactile window.

Fig. 4.3 (left) displays a visualization of the ground-truth raw Xela data, where green represent z-axis displacement and arrows indicate xy-axis displacement of sensors. The reconstructed Xela signal is shown in **Fig. 4.3** (right), and demonstrates qualitative accuracy in recreating sensor displacements. We find that **Sparsh-skin** trained using the MAE recon-

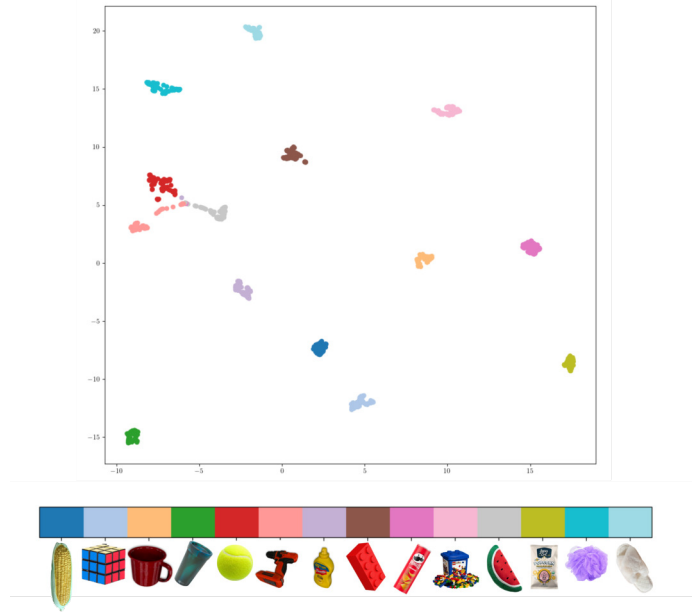


Figure 4.4: We apply UMAP dimensionality reduction on features obtained from 2048 random (0.1s window) samples from the test set of the pretraining dataset. Note that Xela sequences from each object are mapped to distinct, non-overlapping clusters. The **Sparsh-skin** representations capture information that allows to identify objects since they all exhibit different tactile properties.

struction objective (identical tokenization) scheme, is significantly inferior at reconstruction compared to **Sparsh-skin** trained via self-distillation.

In terms of object classification performance, we achieve approximately 95% accuracy across 14 classes, while both BYOL (treating tactile signal as images) [74] and MAE (using tactile and proprioception history) [80] are limited to $\sim 81\%$ accuracy. Additionally, Fig. 4.4 presents a UMAP [131] visualization of the representations, where sequences from each object are mapped to distinct, non-overlapping clusters.

4.4 Evaluation of Sparsh-skin for Contact Perception

We assess the ability of **Sparsh-skin** to comprehend tactile properties, enhance perception, and enable policy learning for manipulation through four downstream tasks spanning tasks studied in the tactile sensing literature: namely (1) Force estimation, (2) Joystick state

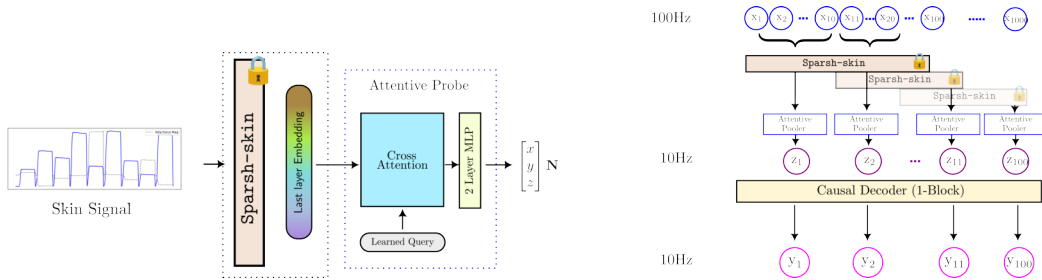
estimation, (3) Pose estimation, and (4) Policy learning via the plug insertion task.

Please refer to Table 4.2 for details on labeled data curation for evaluation tasks.

Task	Dataset	Size	Collector	Label
Force estimation	Normal load (indenter: sphere, flat)	50k datapoints	Robot	3-axis force
Pose estimation	Object sliding	108 trajectories	Human	Object pose $\mathbf{SE}(2)$
Joystick state estimation	Joystick motion	817 trajectories	Human	Normalized roll, pitch, yaw
Plug insertion	Demonstrations	100 trajectories	Human	Absolute EE pose

Table 4.2: Datasets for evaluating **Sparsh-skin** representations on downstream tasks.

4.4.1 Evaluation Protocol



(a) Attentive pooling + small 2-layer MLP for regression tasks (b) Decoder with a 1-layer transformer block for sequence to sequence prediction tasks

Figure 4.5: We use two types of decoders for (a) instantaneous, and (b) temporal tasks. Both decoders contain the attentive pooler which uses a learned query token to cross attend to sensor features to output a *single token full-hand* representation.

Downstream task decoders. The tasks we consider are of two types: a) requiring instantaneous prediction, and b) requiring temporal reasoning over tactile data. For tasks such as force estimation that require an instantaneous estimate, we use attentive pooling (see Fig. 4.5a). For tasks such as pose estimation and joystick state estimation, that require

sequence reasoning, tactile observations are transformed into tokens at the output frequency through a cascaded application of the backbone network. This is followed by attentive pooling as illustrated in [Fig. 4.5b](#).

Model comparisons. For each of the downstream tasks, we explore multiple variants of the **Sparsh-skin** encoder, along with additional baselines:

1. BYOL*, our reproduction of the BYOL [69] approach to tactile representation learning following [74] using our collected play data and tactile data formatted as images, since the setup used in [74] does not contain palm sensing and uses an older variant of the tactile sensor.
2. End-to-end, training the entire encoder-decoder network with same capacity as **Sparsh-skin** using only labeled task data
3. **Sparsh-skin** (frozen), pretrained representation that uses tactile and hand configuration history.
4. **Sparsh-skin** (finetuned), where the encoder network is finetuned with task-specific data.
5. **Sparsh-skin** (MAE), pretrained representation that uses tactile and hand configuration history trained using MAE supervision instead of self-distillation.

For tasks **(1 - 3)**, we measure performance using the average root mean squared error (RMSE). Additionally, we evaluate each method for sample efficiency by reducing the downstream labeled data accessible during training. Then, for **(4)** plug insertion, we measure success rate (SR) across trials where we select the best model from **(1 - 3)** for comparison against the end-to-end baseline.

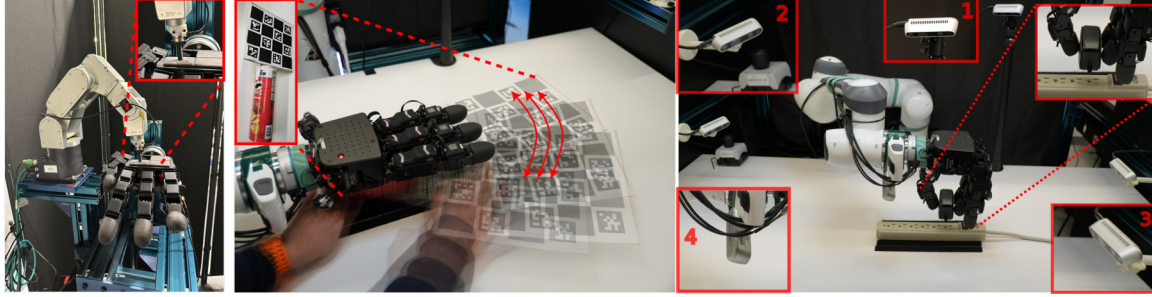


Figure 4.6: Hardware setup used for downstream tasks: **(Left)** shows the setup for force estimation. We use 3D printed probes attached to a F/T sensor to indent onto the Xela sensors. **(Middle)** shows the setup for pose estimation. We track an object mounted with an ArUco marker to obtain ground truth pose estimates while randomly moving it under the robot hand. **(Right)** shows the setup for plug insertion policy task. We collect tactile measurements and camera observations from three third-person view cameras and a wrist camera view.

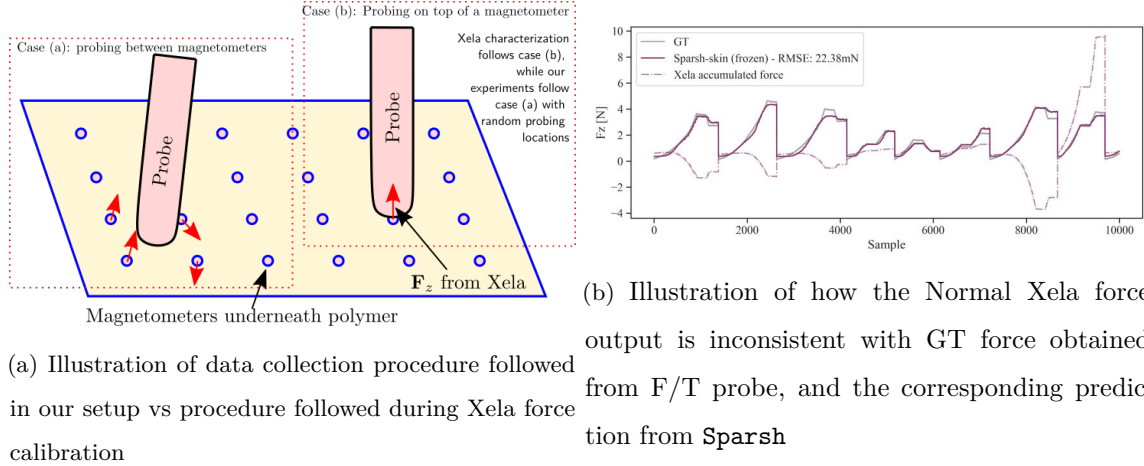


Figure 4.7: Illustration of data collection protocol followed for force estimation with Xela sensors

4.4.2 Force Estimation

This task constitutes regressing tactile signals to 3-axis normal and shear forces applied on the robot hand’s palm. Our data collection setup involves a robot arm with a force-torque (F/T) probe indenter (ATI Nano 17) applying varying normal forces (0.25-5.0N) to the Allegro hand palm, while collecting force-labeled datasets using 3D printed hemispherical

and flat indenter shapes, as shown in Fig. 4.6 (left). As illustrated Fig. 4.7(a), we randomly sample the F/T probe position over the sensor pad, resulting in positions both on top of the magnetometers and in between them, unlike the case of sensor characterization which only tests for positions on top of the magnetometers. This results in cases where the probe may slide and present slightly uneven force outputs. The dataset consists of 3-axis force measurements, end-effector poses, and Xela signals with their corresponding timestamps for synchronization. After resampling to the Xela frequency (~ 100 Hz), our training data set contains 50k datapoints.

The decoder is trained using min-max normalized force measurements, supervised with L1 loss, and optimized with Adam until convergence. We evaluate performance and sample-efficiency using the average root mean squared error (RMSE) by testing **Sparsh-skin** performance under varying amounts of labeled downstream task data.

Sparsh-skin features are pooled via attentive pooling to obtain a full-hand representation $z_{hand} \in \mathbb{R}^d$. The force decoder consist of shallow 2-layer MLP with 3 outputs regressing to normalized force for each axis. Based on results shown in Fig. 4.8(a), we find that while the end-to-end model is particularly worse at predicting forces throughout the spectrum, in low data regimes – 3.3% to 10% of the labeled data in this case – it is interesting to note that **Sparsh-skin** (finetuned) and **Sparsh-skin** (frozen) do not see any significant loss in performance. To this end, we test the models with even smaller number of downstream task data samples to find that **Sparsh-skin** (finetuned) is able to predict forces at a reasonable accuracy (350 mN in z) even with only ~ 100 samples. Additionally, we find **Sparsh-skin** (MAE) is worse at predicting forces highlighting that MAE may not be suitable for noisy magnetic flux signals. Furthermore, BYOL* is competitive albeit marginally inferior with respect to **Sparsh-skin** (frozen) as this task tests for instantaneous force decoding.

We note that we indent the tactile sensor pads at both, positions on top of the sensor as well as positions in between magnetometer locations, while choosing these positions randomly. This results in cases where the probe may slide and present slightly uneven force outputs. Specifically, in Fig. 4.7(b) we note that **Sparsh-skin** predicts the correct normal forces, while accumulation (mean) of normal forces from the magnetometers over the sensor pad results in inconsistent force outputs compared to ground truth. Please refer

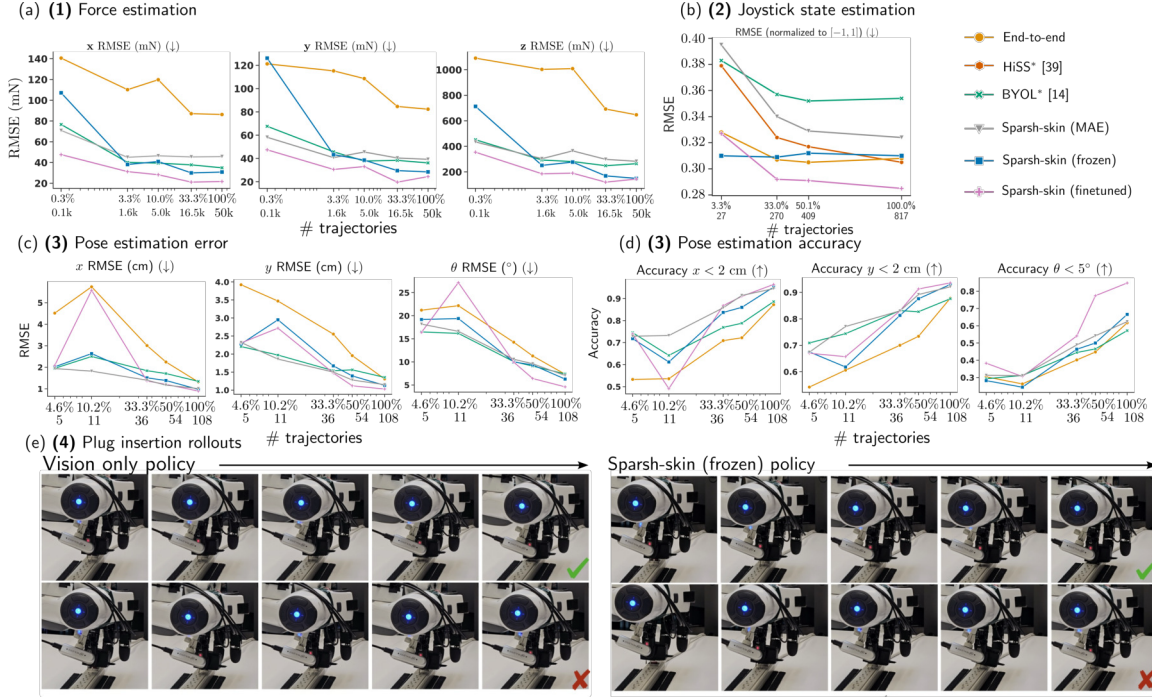


Figure 4.8: Summary of results comparing Sparsh-skin on all tasks. (a) Force estimation (RMSE $\downarrow$): BYOL pre-training is less accurate at predicting normal forces. (b) Joystick state estimation ($\downarrow$): Sparsh-skin outperforms end-to-end overall and is competitive with HiSS* even when it is given access to only 3.3% of dataset. (c) Pose estimation error ($\downarrow$) and (d) Pose estimation accuracy ($\uparrow$): Sparsh-skin (finetuned) has a $\sim 10\%$ improvement over end-to-end for translation and $\sim 20\%$ improvement for rotation. (e) Snapshots of plug insertion policy rollouts (success and failure). Vision-only policy succeeds primarily when the starting position is directly above the socket, while Sparsh-skin (frozen) achieves 75% success rate, with failures mainly due to loss of grip when sliding to locate the socket.

to Appendix B.1 for additional illustrations of force estimation accuracy from Sparsh-skin representations.

4.4.3 Joystick State Estimation

We adapt this task from [14], as a study of *full-hand* object state estimation. The task is a sequential problem of predicting the joystick states (roll, pitch and yaw) given a short

tactile history. Furthermore, the challenges are amplified in the presence of drifting Xela signal and high-dimensional observations from all the sensing pads.

This dataset contains 1024 trajectories of the Xela sensorized Allegro hand interacting with a joystick of approximately 25 to 40 seconds each. We resample the synchronous joystick states and Xela tactile signal data to 10Hz and 100Hz, respectively. The joystick states in this dataset are recorded as $(x, y, z) \in [-1, 1]^3$, corresponding to the normalized roll, pitch, and yaw of the joystick, and the Xela sensor observations are recorded with the baseline already subtracted. We use the decoder architecture for sequence prediction as in Fig. 4.5b, where the output frequency of predictions is 10Hz.

In addition to the comparison of **Sparsh-skin** with an end-to-end approach, we also compute the RMSE results from the best reported model in HiSS [14] (denoted as HiSS* in Fig. 4.8(b)), which uses a hierarchical sequence-to-sequence model leveraging the Mamba state-space model [70] as the baseline from the open-source code. We test the sample efficiency of all models by restricting access to a small number of labeled data samples for training the decoders.

Sparsh-skin matches baseline (HiSS [14]) performance using full data, despite challenges from jittery teleoperation such as inconsistent touch even with similar joystick maneuvers. Notably, **Sparsh-skin** (frozen) achieves similar performance even with only 3.3% of the data, demonstrating high sample efficiency. **Sparsh-skin** consistently shows lower prediction error across data budgets (Fig. 4.8(b)). Furthermore, **Sparsh-skin** (finetuned) drastically speeds up training, reaching comparable performance to an end-to-end approach in 12k optimization steps versus 220k (a 95% speedup) when using a 33% data budget, see Appendix B.2.

4.4.4 Pose Estimation

Knowledge of object pose changes is critical for visuo-tactile SLAM [169] and dexterous manipulation tasks such as in-hand rotation, in-hand translation, and reorientation [202, 149, 203, 199] where object pose is usually part of the privileged state information when training policies using simulated data. To transfer policies to the real world, a state estimator is

trained to match the latent embedding of the observations from privileged information and when the state is restricted to a history of available proprioception inputs [149, 148].

While a direct view into the instantaneous contact state of the object can be obtained by thresholding the magnetic flux change observed by the Xela sensors, this task tests the ability to track and accumulate slip under the sensors to predict object pose changes ($\mathbf{t}_i^R \triangleq (x, y, \theta) \in \mathbf{SE}(2)$) using the setup in Fig. 4.6 (middle). In particular, we evaluate whether from the skin representations we can estimate SE(2) transformations of the object relative to the sensor plane in the robot coordinate frame.

In our data collection protocol, the object is placed on a table and with the robot hand facing downward and pressing against the object. We collect 120 trajectories (~ 30 s each), by manually sliding/rotating an object in a range of $\sim (25\text{cm}, 25\text{cm}, 100^\circ)$ under the Allegro hand, tracking ground truth object pose using ArUco tags. These poses in the camera frame are subsampled and then transformed into the robot hand frame and then projected into $\mathbf{SE}(2)$. Furthermore, since we do not use a highly accurate method to compute the ground truth pose, we preprocess the relative ground truth poses to reduce noise by smoothing the pose trajectories using the Savitzky-Golay filter.

We use the sequence decoder (Fig. 4.5b) which processes 1-second windows of tactile data (100Hz) and object pose (10Hz). The pose ground truth is standardized using mean-variance normalization, and the downstream decoder is supervised with MSE loss. In addition to RMSE, we measure the performance for this task using pose accuracy, measured as the proportion of the data samples that the model predicts with translational error under 2cm and a rotational error under 5° . The dataset contains well distributed poses trajectories spanning a range of $\sim (25\text{cm}, 25\text{cm}, 100^\circ)$. To evaluate sample efficiency, we train with data from sequences ranging between 5 to 100 trajectories.

From Fig. 4.8(c)(d), all representations models pre-trained on play data achieve lower RMSE and higher pose prediction accuracy than the traditional end-to-end approach. In particular for **Sparsh-skin** (fine-tuned) we find a $\sim 10\%$ improvement over the end-to-end model with the full dataset for translation, and $\sim 20\%$ improvement for rotation. In low data regimes (33%) **Sparsh-skin** (MAE) outperforms other models since there is a direct correlation between translation and the displacement of the magnetometers on the Xela

sensors, while BYOL* shows similar performance to **Sparsh-skin** (frozen) which maintains $\sim 70\%$ accuracy. This is explained by the correlation of the displacement of the object and the displacement of the magnetometers on the Xela sensors. Additionally, we observe that allowing in-domain data to fine-tune the **Sparsh-skin** representations is advantageous, especially for better tracking rotation of the object, which is harder, as it involves torsion. Additional analysis of the performance of pose estimation using **Sparsh-skin** representations can be found in Appendix B.3.

4.5 Policy Learning: plug insertion

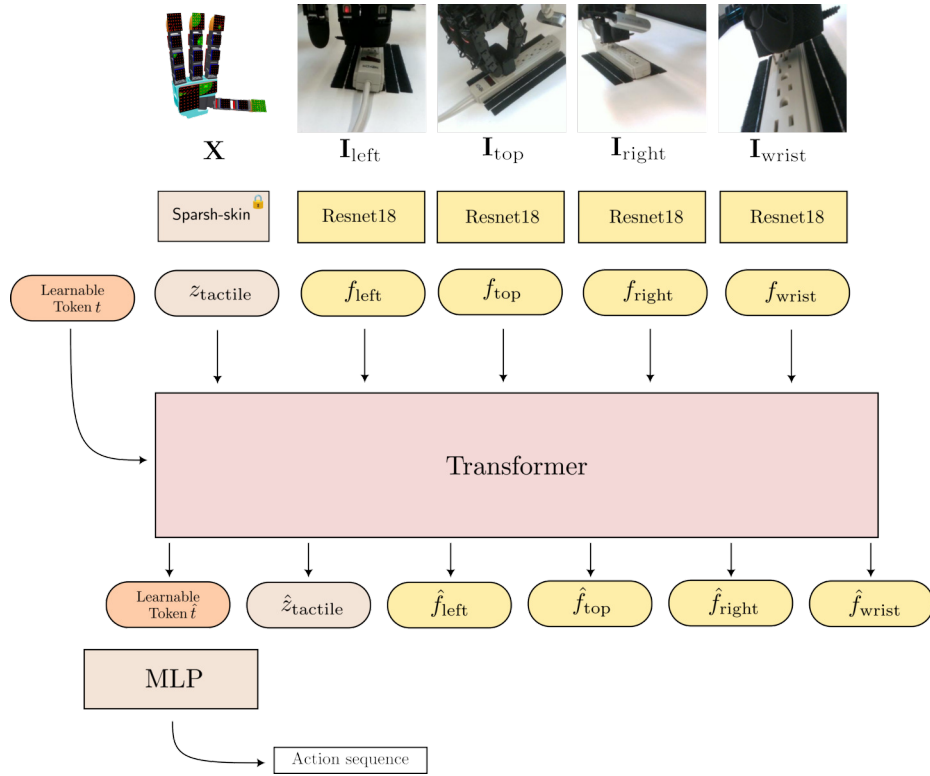


Figure 4.9: Illustration of the policy architecture: We use a transformer to fuse information from visual and tactile modalities, through the use of a learnable action token, which is then used to subsequently predict action sequences.

We adapt the insertion task [40, 192, 144] as it is fundamentally tactile requiring touch

feedback to observe the alignment state of the plug. The task involves inserting a pre-grasped plug into a fixed socket using a 7-DOF Franka arm and Allegro hand (Fig. 4.6-right) unlike [144] which used parallel jaw grippers. We collected 100 demonstrations via kinesthetic teleoperation, recording synchronized data: four camera views ($\mathbf{I}_t^{\text{left}}, \dots$), Allegro tactile readings (z_t), and robot joint states. The arm’s initial position was randomized within a $0.05m \times 0.05m \times 0.02m$ volume $\sim 10\text{cm}$ above the socket, while the socket position is fixed. The policy predicts sequences of absolute end-effector poses (3D position + axis-angle orientation) $\mathbf{a} \triangleq (\mathbf{T}_t, \mathbf{T}_{t+1}, \dots)$, conditioned on visual and tactile observations but not proprioception (joint states).

We train an imitation learning policy that outputs action sequences [219], with **Sparsh-skin** representations as input. We use a transformer decoder to predict action sequences given camera and tactile observations. Fig. 4.9 illustrates the architecture of the action decoder. Images are encoded using a Resnet18 CNN, which are trained from scratch to produce image features, while the tactile observations are processed through **Sparsh-skin**. Further, a learnable token (CLS / action) token is also concatenated with the observation tokens. After processing through the transformer, we extract the action token, which is then passed into a small 2-layer MLP to predict a sequence of actions. For this task, follow a *receding-horizon* control approach, where we choose a prediction action sequence length of 16, of which 8 actions are executed, given only the observations from the current timestep.

We evaluate average success rate over 20 trials with randomized start positions, comparing **Sparsh-skin** variants (V + **Sparsh-skin** (frozen), V + end-to-end, V + **Sparsh-skin** (finetuned)) against a vision-only (V) baseline to assess tactile contribution.

We find that policies conditioned on pretrained **Sparsh-skin** features outperform the end-to-end model, as shown in Table 4.3. In Fig. 4.8(e), we present snapshots of real-world policy deployments for both vision-only and visuo-tactile **Sparsh-skin** (frozen) policies in two scenarios: a successful and a failed rollout.

Without tactile modality (vision only), we find that the policy is able to get close to the socket but indefinitely continues to search for it and does not push the plug in, even when it is directly above the socket. Further, we find that this policy tends to keep pushing the plug to the left of the socket, which we hypothesize is due to perceptual aliasing, where the

Policy	Success Rate ($\uparrow$)
VisuoSkin [144]	0.66
Vision only (V)	0.20
End-to-end V+T	0.40
Sparsh-skin V+T (frozen)	0.75
Sparsh-skin V+T (finetuned)	0.70

Table 4.3: Policy learning (plug insertion): Success rate percentage reported over 20 trials, while ensuring identical initial conditions during each trial for the tested policy variants. VisuoSkin results are obtained from [144]

plug incorrectly appears to be right above the socket from the wrist camera. On the other hand, all model variants with access to the tactile modality observe respectable success rates. In qualitative inspection, we find that the policies using **Sparsh-skin** (V+T frozen) representations slides after making contact with the extension board, while **Sparsh-skin** (V+T end-to-end) and **Sparsh-skin** (V+T finetuned) tends to retry by lifting the plug, when mistakes occur. As noted earlier, in comparison with [144, 13] which trains end-to-end visuo-tactile policies, our setup uses a multifinger Allegro hand as the manipulator, where the plug is grasped using three fingers; nevertheless, we find that policies trained with tactile features from **Sparsh-skin** are competitive.

4.6 Related Work

Representation learning for vision-based tactile sensors has recently gained significant attention. Since the sensor outputs are images, techniques from computer vision [80, 181] have been extended to tactile sensors. This is motivated by a move beyond task-specific encoders to pretrained encoders that promise generalization, with prior work leveraging mask-autoencoders (MAE) [217, 21], contrastive learning [205, 197, 38, 65], and state-of-the-art methods like self-distillation and joint-embedding predictive architectures [87] to learn tactile representations.

Research on learning representations for magnetic-based sensors remains relatively underexplored. Since these sensors produce low-dimensional signals, the consensus view is that representation learning is likely unnecessary. However, as we highlight in our work, these signals are indeed high-dimensional due to the complexities of full hand sensing, dynamic tactile signals and hand poses, and magnetic sensor physical properties. This high dimensionality means they benefit from large-scale pretraining to compress information into semantically rich representations that enhance downstream task performance. Recently, HyperTaxel [117] applied contrastive learning to learn representations for the Xela sensor for the task of surface recognition but it did not show whether these representations capture contact dynamics. Similarly [74, 191] propose representation learning with self-supervised methods such as BYOL [69] and MAE [80]. While the idea of representation learning is promising, the choice of meaningful image augmentations without data corruption, is unclear for BYOL. Furthermore, by treating instantaneous tactile measurements as images, these methods discard temporal information and may therefore be suboptimal for tactile tasks that rely on contact dynamics.

4.7 Discussion

In this chapter we introduced **Sparsh-skin**, a tactile representation model trained via self supervision for magnetic skins on dexterous hands. Our model learns contact state priors from an unlabeled dataset of contact-rich teleoperated hand interactions with various household objects.

The proposed self-supervised learning (SSL) approach is based on the self-distillation paradigm, where we introduce Xela sensor tokenization and block masking. Through evaluations across tactile-centric estimation (force, pose, object state) and policy learning tasks, we demonstrate the efficacy of our supervision objective, tokenization, masking strategies and pretraining of the model over a large unlabeled dataset containing ~ 4 hours of atomic contact interactions with household objects. In experiments, when considering sample efficiency (training on 33% downstream data), we find that **Sparsh-skin** (frozen) outperforms the end-to-end model baseline by $\sim 56\%$, our adaptation of the BYOL approach to tactile representation learning by $\sim 28\%$ and the **Sparsh-skin** (MAE) baseline by $\sim 53\%$. We be-

lieve that **Sparsh-skin** represents a step toward foundation models for full-hand tactile representations that enables high-dexterity robotics tasks.

Limitations. We identify the following limitations for the model design and evaluation framework:

1. **Model design:** Although **Sparsh-skin** handles contact dynamics implicitly by learning representations over windows of tactile signal, the data corruption strategy is inherently spatial. Future work may consider explicitly handling temporal correlation learning via temporal prediction tasks.
2. **Pose estimation:** Our pose estimation task is not designed for applications where the hand is not static. While pose estimation in its current iteration is designed to track a 2D pose with a flat fixed hand configuration, real object interactions involve both 3D pose changes, and simultaneously changing hand configurations. There are also exciting avenues to study tasks such as grasp stability prediction [19] and slip prediction.
3. **Manipulation policy:** Although our experiments with the real physical system support the hypothesis that tactile information from magnetic skin sensors improves policy performance, we still need to assess their ability to generalize. Visuo-tactile policies can overfit to the specific tactile signatures of objects and environments used in data collection.

Chapter 5

MULTISENSORY TOUCH REPRESENTATIONS FOR ROBOT MANIPULATION

5.1 Introduction

Touch is a rich and multifaceted sense that plays a central role in human dexterity. Humans fluidly adapt their interactions to the physical properties of objects by integrating a wide spectrum of touch signals that include skin deformation, vibrations, motion, and pressure. This multisensory feedback enables us to distinguish between a plastic and paper cup, twirl a pen between fingers with ease, and manipulate tools under severe visual occlusion. Leveraging the multisensory nature of touch is desirable for robust, fine-grained robot manipulation.

Despite its importance, multisensory touch remains significantly underutilized in robotics. Most approaches rely on unimodal tactile sensing, such as GelSight-like sensors [208, 108, 114], due to standardized hardware availability. However, advances like Digit 360 [109] now enable capturing high-resolution images, vibrations, motion, and pressure in a compact form, making multisensory touch accessible. Prior work on using tactile modalities independently shows promise [205, 132], but a unified, scalable, and easily integrable method to take advantage of these modalities is still lacking. Representation learning offers a viable solution to integrating heterogeneous sensory inputs from sensors like Digit 360 by fusing complementary contact information from all modalities into a shared latent space. In addition, as has been shown for vision-based tactile sensors [195, 217, 87, 72] representation learning allows a downstream task training to be more data-efficient and robust to noise or irrelevant variations [140, 10, 83, 98].

This chapter introduces **Sparsh-X**, the first self-supervised backbone for multisensory touch representation learning across four key tactile modalities: image, audio, motion, and pressure. Trained on unlabeled data from diverse manipulation behaviors, such as sliding,

tapping, rotating, picking up, and dropping, **Sparsh-X** learns to fuse heterogeneous tactile signals into compact and expressive contact embeddings. Beyond **Sparsh-X**, this unlabeled dataset also facilitates research in representation learning and benchmarking. Through supervised tasks, we show that **Sparsh-X** captures a wide range of physical properties, including object-level characteristics (e.g., type and mass), static contact properties (e.g., force and material), and dynamic interaction cues (e.g., motion and impact). Representations that encode these physical properties at the fingertip level are especially valuable for dexterous manipulation, as they enable feedback of object and contact state directly in latent space.

Ultimately, tactile signals are valuable only when effectively used in policy learning. This remains a challenge, particularly in reinforcement learning due to the *sim-to-real* gap [121]. We demonstrate that **Sparsh-X** can be effectively applied in policy training via two examples: (1) imitation learning, and (2) tactile adaptation of policies trained in simulation with privileged access to contact information. Experiments across manipulation tasks, including insertion and in-hand rotation, demonstrate that integrating **Sparsh-X** leads to significantly improved real-world performance over end-to-end tactile image baselines. By unifying multi-sensory touch in a shared latent space, **Sparsh-X** takes a step toward foundation models for touch, enabling scalable and data-efficient learning for fine-grained robotic manipulation.

5.1.1 Contributions

To summarize, this chapter makes the following key contributions:

- [Section 5.2](#) introduces **Sparsh-X**, the first unified backbone for multisensory touch fusing image, audio, motion, and pressure signals into a general-purpose representation. We describe the architecture, fusion mechanism and data collection using Digit 360.
- [Section 5.3](#) presents evaluation protocol to corroborate **Sparsh-X** representations capturing physical properties of objects and contact interactions solely from SSL pretraining.

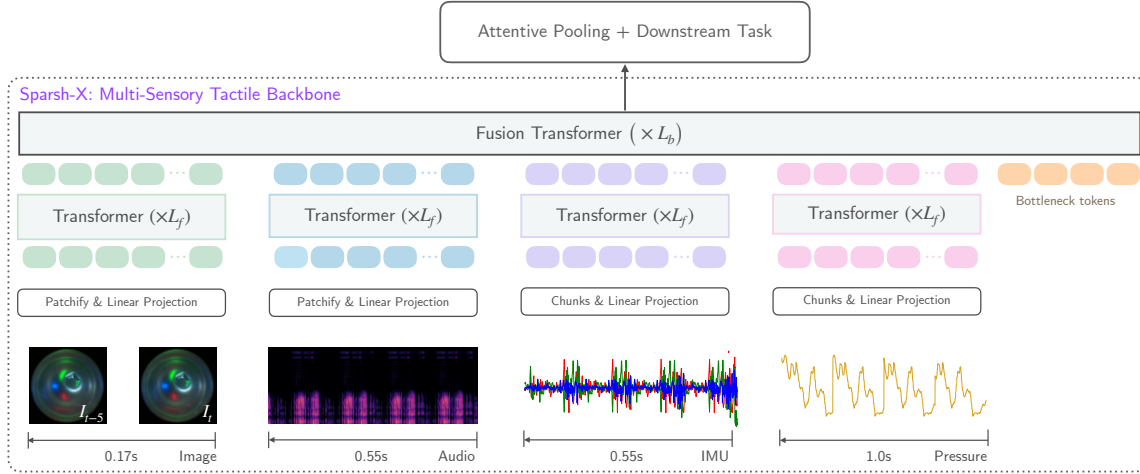


Figure 5.1: **Sparsh-X**, a multisensory touch transformer for general-purpose representations, integrates four tactile inputs: image, audio, accelerometer, and pressure. Each modality is processed independently in the first L_f layers, then fused using bottleneck tokens for cross-modal attention in the final L_b layers.

- Section 5.4 shows how real-world touch representations can be leveraged for fine-grained manipulation skills, via imitation learning and tactile adaptation.

5.2 Learning Multisensory Touch Representations

Fusing tactile modalities is crucial to discovering correlation between modes. For instance high-frequency audio and tactile image can both indicate making/breaking contact. However, traditional tactile sensing work has largely relied only on unimodal approaches, i.e., on tactile images of elastomer deformations [208, 108]. In cases, where additional modalities are considered [205, 116], they have been treated independently. In this section, we introduce **Sparsh-X**, a backbone for general multimodal touch representations for the Digit 360 [109] sensor. Our model integrates four tactile modalities: image, audio, accelerometer, and pressure. Through self-supervision on $\sim 1\text{M}$ unlabeled contact interactions, **Sparsh-X** compresses contact information into a unified multimodal representation.

5.2.1 Model Architecture

Sparsh-X is a transformer-based backbone [47] (see Fig. 5.1) where each input signal is first processed independently for L_f layers through self-attention. Thereafter, we allow cross-modal information flow via attention bottlenecks, as in [136]. Specifically, we concatenate B bottleneck fusion tokens to each modality’s embedding for the subsequent L_b blocks. After each cross-modal update, the fusion tokens are averaged across modalities to promote information sharing. Intuitively, the bottleneck tokens act as multimodal summarizers, distilling and exchanging information between tactile modalities within each transformer block. Following experimental insights from [136], we set the total number of transformer layers to $L = L_f + L_b = 12$, with $L_f = 8$ layers for unimodal processing and $L_b = 4$ fusion layers with $B = 4$ bottleneck tokens.

The inputs to **Sparsh-X** are image, audio, accelerometer, and pressure recorded by the Digit 360 sensor. Since all modalities have different sampling frequencies and data structures, we describe the steps for preprocessing and tokenization. *Tactile images* are sampled at 30fps [87] and passed to the model with a temporal stride of 5 concatenated along the channel dimension. We crop to zoom-in the fish-eye image and resize to $224 \times 224 \times 3$. Image patches (16×16) are then tokenized to embeddings of 768 dimensions through a linear projection layer. *Audio* comes from two contact microphones sampled at 48kHz. A 0.55s window of audio signal is converted into a log-mel spectrogram of 128 channels computed from a 5ms Hamming window with hop length 2.5ms. We concatenate the spectrograms from both microphones, resulting into an audio input of 224×256 which is further tokenized with a patch size of 16. *IMU* data from the 3-axis accelerometer is sampled at 400Hz and combined in a window of 0.55s. The *pressure* signal is sampled at 200Hz and combined in a window of 1.1s window. Both signals are tokenized resulting in 224×3 and 224×1 temporal signals.

5.2.2 SSL Training Pipeline

We train **Sparsh-X** using Self-Supervised Learning (SSL) which offers several benefits, including the ability to learn general representations, robustness to distractors, and independence from labeled data. Our SSL training dataset consists of ~ 1 M samples generated from

two primary sources (Fig. 5.2): an Allegro hand with Digit 360 sensors on the fingertips that performs random motions with objects such as dipping into a tray filled with various items; and a manual picker [29, 204, 160] with the same sensor adapted to the gripping mechanism, used to execute atomic manipulation actions such as picking up, sliding, tapping, placing, and dropping objects against diverse surfaces that vary in roughness, hardness, softness, friction, and texture properties. Please refer to Appendix C.1 for further details regarding the pretraining dataset.

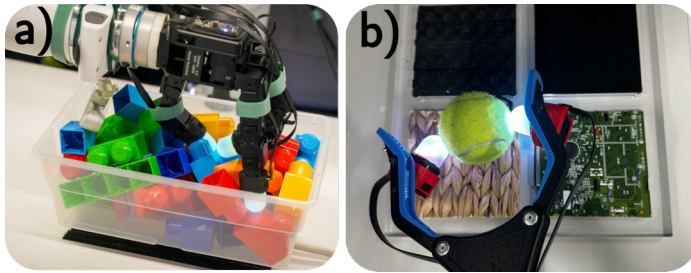


Figure 5.2: Pretraining data collection setup using (a) Digit 360 and Allegro hand (b) two-fingered manual picker.

We employ a teacher-student self-distillation approach [22, 140], where both branches consist of an encoder and a predictor head. After tokenizing each multisensory touch input, appending a register token, and adding sinusoidal positional embeddings, we apply masking to the student input tokens per modality, retaining 10-50% of the signal for local masks and 50-100% for global masks. We concatenate the register tokens from global and local masks and pass them through their corresponding prediction heads. As in [22], the prediction task involves clustering, where teacher tokens serve as pseudo-labels for the student network, with centroids that adapt over time as the model learns. We use cross-entropy between the softmax outputs of the teacher and student networks as optimization objective. We train **Sparsh-X** for 200 epochs on 16 A-100 GPUs, with 128 batch size and AdamW optimizer with linear rampup followed by a cosine schedule as the learning scheduler.

In the next section, we propose a set of downstream tasks for evaluating the capability and generalization of our representations on touch-centric tasks. Our study is driven by two central research questions, first, *what physical properties do our representations capture from*

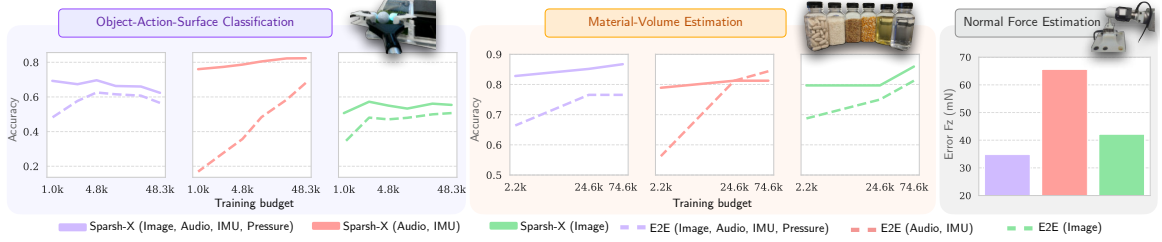


Figure 5.3: Performance of frozen **Sparsh-X** representations with different tactile inputs. The synergy of multiple modalities improves object-action-surface identification (*left*) and material-quantity estimation (*middle*), outperforming tactile image alone and showing data efficiency over E2E. Combined modalities also enhance normal force estimation (*right*), a task typically addressed with vision-based tactile sensing.

contact interactions?, and second *how can real-world touch representations be leveraged for manipulation policy learning?*.

5.3 Inferring Physical Properties with Sparsh-X

We design supervised tasks to evaluate **Sparsh-X** representations’ ability to capture physical properties for robotic manipulation. These tasks cover (1) object-level characteristics (e.g., type and quantity), (2) static contact properties (e.g., force and material), and (3) dynamic interaction cues (e.g., sliding, tapping). For each task, we train a task-specific attentive decoder [25] in a supervised manner, using as input **Sparsh-X** representations. Importantly, the encoder weights of **Sparsh-X** are frozen to isolate and assess the quality of the representations learned from self-supervised pretraining.

5.3.1 Object-Action-Surface Classification

We repurpose the SSL pretraining dataset collected with the manual picker. We annotate each data point with metadata specifying the object being held (golf ball, wood block, LEGO block), the action performed (tap, linear slide, circular slide), and the external surface in contact (grass, fabric, plastic, foamwork). From the 104 available sequences, 69 are used for training and the remaining 35 for testing the performance of the classifier. The dataset is balanced in number of samples per label.

We train the decoder on the subset of the pre-training dataset that uses the manual picker to jointly classify the object being grasped (golf ball, LEGO, or wood block), the action being performed (planar sliding, circular sliding, or tapping), and the extrinsic surface in contact with the object (plastic, fabric, grass, or formwork). This task probes whether the representations encode static contact properties like friction, stiffness, and roughness, as well as dynamic cues that distinguish between different motion patterns. To assess the benefits of multisensory touch, we ablate the input modalities to **Sparsh-X** and compare performance against traditional end-to-end (encoder-decoder) models trained from scratch. As shown in Fig. 5.3 (left), combining tactile modalities significantly boosts accuracy. For instance, pairing audio with IMU yields a 32% improvement, while using all modalities together provides a 13% gain compared to using tactile images alone. Pre-training further enhances performance when using all modalities, consistently outperforming E2E models with task-specific embeddings, with a particularly notable 10% margin under the lowest data regime. For completeness, confusion matrices plots are provided in Appendix C.3.

5.3.2 *Material-Quantity Estimation*

We evaluate the capacity of our model to distinguish materials (both solids and liquids) and to provide a coarse mass estimation through shaking motions [92, 130, 119, 32]. We train an attentive decoder to jointly classify material type and quantity by shaking 8oz bottles with a parallel gripper equipped with Digit 360 sensors (see Figure C.3). The dataset includes four solids (corn kernels, lentils, vitamin pills, rice), two liquids with distinct viscosity (water, oil), and three fill levels (full, half, quarter). More about data collection in Appendix C.2. As shown in Fig. 5.3 (middle), **Sparsh-X** representations from all modalities achieve the highest accuracy across all training data budgets, outperforming end-to-end models trained on tactile images alone by 20.5%. **Sparsh-X** representations outperform end-to-end models across all sensory inputs, demonstrating superior data efficiency and generalization. For completeness, confusion matrices plots are provided in Appendix C.3.

5.3.3 Normal Force Estimation

Following the protocol from **TacBench** ([Section 3.4.1](#) and [Appendix C.2](#)), we use a hemispherical probe to apply normal forces of up to 3.5N to the Digit 360 sensor by indenting perpendicularly into the elastomer surface. An attentive regression head is trained to estimate the applied normal force from frozen **Sparsh-X** representations across various tactile sensory inputs. Evaluation is performed on samples with randomized force magnitudes and indentation locations, using the same probe geometry. As shown in [Fig. 5.3](#) (right), combining all tactile modalities leads to improvement in force estimation accuracy, achieving an average error of 35mN, a 17% improvement over using only tactile image.

5.4 Sparsh-X for Policy Learning

We investigate whether manipulation policies can benefit from touch representations that encode, in latent space, information about both static and dynamic physical object properties, such as friction, mass, and forces experienced at the fingertips. We demonstrate the effectiveness of multisensory touch representations for learning and improving manipulation policies in two key settings: (a) real-world imitation learning and (b) real-world tactile adaptation of sim-trained policies. We evaluate our approach on two contact-rich tasks, plug insertion and in-hand rotation. Through these application examples, we highlight how general-purpose touch representations can boost policy performance and robustness.

5.4.1 Plug-Insertion via Imitation Learning

Insertion is a fundamental skill in robot manipulation and has long served as a benchmark task in the literature [\[40, 161, 173, 192\]](#). We evaluate the utility of multisensory touch representations in a plug-insertion task, where a robot equipped with an Allegro hand and Digit 360 sensors must insert a pre-grasped plug into a fixed socket ([Fig. 5.4](#)-left). Using kinesthetic teleoperation, we collect 100 demonstrations with randomized initial arm poses, recording joint states, wrist poses, camera images, and tactile data.

Our policy architecture is adapted from ACT [\[220\]](#). The observations to the policy include an embedding of the wrist camera image and **Sparsh-X** representations for thumb,

index, and middle finger sensors, processed through an attentive pooling layer [25]. We train a ResNet18 [82] in an end-to-end (E2E) fashion to learn the wrist image embeddings. We append learnable action tokens, which are processed by the transformer and subsequently decoded into a sequence of actions. In our setup, the model predicts a sequence of absolute robot end-effector poses i.e., $\mathbf{a} \triangleq (\mathbf{T}_t, \mathbf{T}_{t+1}, \dots, \mathbf{T}_{t+H})$ with a prediction horizon of $H = 8$. Policy block diagram and deployment information of the policy in the experimental setup is presented in Appendix C.4.

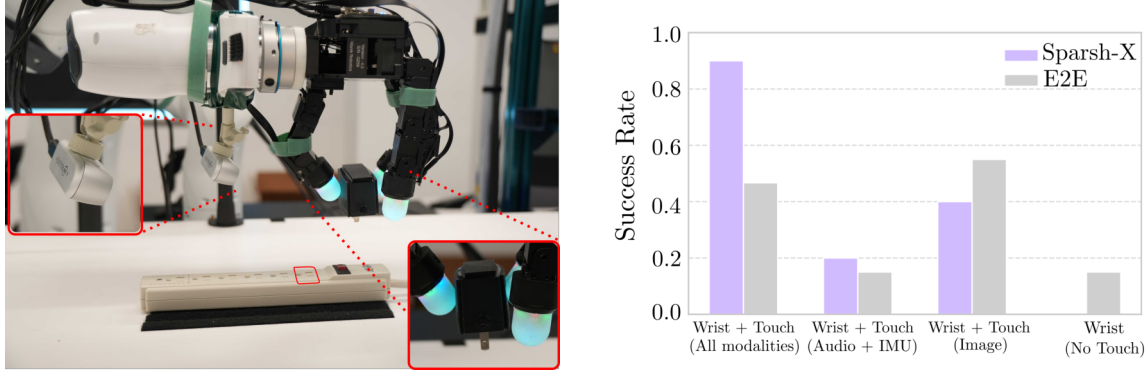


Figure 5.4: Plug-insertion task. (Left) Experimental setup. (Right) Success rate over 20 trials. Leveraging multimodal touch with Sparsh-X improves performance by 500% over external-vision-only and 63% over E2E tactile-vision-only policies.

Evaluation. To evaluate the contribution of multisensory touch, we ablate the policy by varying the combination of tactile sensory inputs used, always in conjunction with the wrist camera. We also include a vision-only baseline without touch. Each policy is evaluated in 20 trials, with randomized starting positions on the wrist.

As shown in Fig. 5.4(right), multisensory touch is key for achieving high performance in a tight-tolerance insertion task, with a 90% success rate. Pretraining plays a crucial role, yielding a 90% performance boost compared to training representations with all modalities from scratch jointly with the policy model on task-specific data. These results highlight the benefits of both multimodality and pretraining. Access to multiple sensing modalities enables better discrimination of subtle contact cues. For example, audio can signal initial contact or collisions, while tactile images and pressure provide information about normal

and shear forces that are critical for alignment and insertion.

Policies using tactile images outperform those relying solely on audio and motion cues. Interestingly, for tactile images, end-to-end training outperforms using frozen pretrained representations. The tactile image signal varies little between trials, allowing a specialized encoder to focus on subtle changes in contact patch location and size. It is worth noting, however, that these encoders are evaluated in-distribution, suggesting that pretrained representations may still benefit from increased data diversity and scale or fine-tuning. Notably, our results show a 63% improvement in performance when using **Sparsh-X** with all modalities compared to an end-to-end policy trained with tactile image alone. While pretrained tactile image encoders may benefit from broader data diversity, combining touch modalities helps mitigate such limitations, given their complementary nature. As a final remark, incorporating touch significantly improves performance on this task. In particular, the wrist-only policy often fails due to visual aliasing, where insufficient camera parallax results in plug pegs incorrectly appearing directly on top of socket openings, leading to failed insertions.

5.4.2 In-Hand Rotation with sim-to-real Tactile Adaption

A common strategy for learning dexterous manipulation policies is to first train a base policy using privileged information typically available only in simulation (e.g., object physical properties like mass and friction, or contact signals like location and forces) and later distill the information into another model using inputs only accessible in the real-world such as proprioception [105]. This raises a natural question: *when richer information becomes available in the real world, how can we bring policies trained in simulation closer to the privileged information setting?*

We explore sim-to-real tactile adaptation in the context of in-hand rotation. As our base policy, we use Hora [147], a proprioception-only policy for rotating objects along the z -axis. Hora is trained in simulation via rapid motor adaptation [105], leveraging privileged information such as object pose, shape, mass, friction, and other properties that can be perceived through touch at the fingertips. Since **Sparsh-X** captures physical properties, our goal is to do *tactile adaptation* on top of Hora, improving stability during rotation by

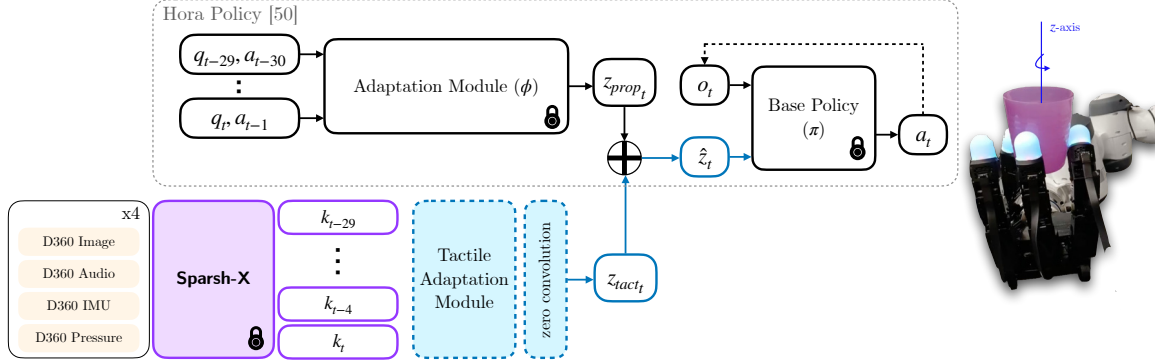


Figure 5.5: We introduce real-world tactile adaptation of sim-trained policies via ControlNet [215], where the zero-convolution layer enables gradual fine-tuning of the embedding $\hat{z}_t$ using Sparsh-X representations.

reducing slip.

We propose *tactile adaptation* via ControlNet [215], which allows the integration of new control modalities without retraining the base model. Applied to policy learning, ControlNet ensures that performance does not degrade below that of the original Hora policy. As shown in Fig. 5.5, we learn a tactile adaptation module that connects to the frozen base policy through a zero-initialized convolutional layer, allowing progressive integration of tactile information. Specifically, we feed to the tactile adaptation module Sparsh-X representations of all four fingertips over the past 1.5 seconds, aligning with the proprioceptive history window used by the base policy.

We rollout the baseline Hora policy (using the open-sourced policy checkpoint) to collect real-world sequences, capturing proprioceptive joint states, target actions, and tactile data related to the in-hand rotation of cup-like objects. For training, we select 50 successful trajectories in which the object remains stably rotating along the z -axis for at least 30 seconds. The tactile adaptation module is then optimized to minimize the L2 loss between the real-world joint angles and the target action output from Hora with the ControlNet. The policy block diagram and the implementation information of the policy on the experimental setup is presented in Appendix C.4.

Evaluation. We compare the baseline Hora policy against two tactile adaptation vari-

ants: Hora+ControlNet(**Sparsh-X**), using pre-trained **Sparsh-X** representations with different combinations of tactile modalities, and Hora+ControlNet(E2E), trained end-to-end. To isolate the impact of tactile feedback over improvements from rejection sampling, we also evaluate against Hora fine-tuned on real-world data and a proprioception-only imitation learning baseline. The primary goal is to improve stability during in-hand rotation by reducing slip, where the object either shifts into the palm or is completely dropped. We evaluate each policy based on vertical drift and time-to-fall, performing 10 trials per policy with a maximum episode duration of 60 seconds.

Our ControlNet approach with **Sparsh-X** representations, reduced vertical translation by 90% (see Fig. 5.6 (top)), using either all modalities or just tactile images. While Hora+ControlNet(E2E) also improved stability a bit, it slowed rotation. Critically, our method outperformed both finetuned Hora and the proprioception-only imitation learning baseline, demonstrating that the benefit stems from tactile feedback, not just good demonstrations. The imitation learning baseline was unreliable, frequently failing due to out-of-distribution states as reflected in the lowest time-to-fall metric.

We also evaluate model robustness to altering the physical properties of the object, specifically friction and mass as shown in Fig. 5.6 (middle). When friction is reduced, Hora+ControlNet(**Sparsh-X**) outperforms all other policy variants. By using a synergy of all tactile modes, it can maintain object stability without losing grasp. In contrast, Hora+ControlNet(image) struggles, possibly indicating that changes in the contact patch are too subtle to be captured by **Sparsh-X** from tactile images alone. This result underscores the complementary strengths of multisensory touch. When the object’s mass is increased, Hora+ControlNet(**Sparsh-X**) and Hora+ControlNet(image) allow the baseline policy to adapt its finger gaiting to compensate for the added weight. This adaptation is possible because the tactile properties of the object can be captured by **Sparsh-X** and transferred to the privilege information embedding in the latent space.

5.5 Related Work

Vision-based tactile sensors [208, 108, 114, 45] have been widely used in contact-rich tasks, including material and volume prediction [211, 71], shape inference [169, 60], localiza-

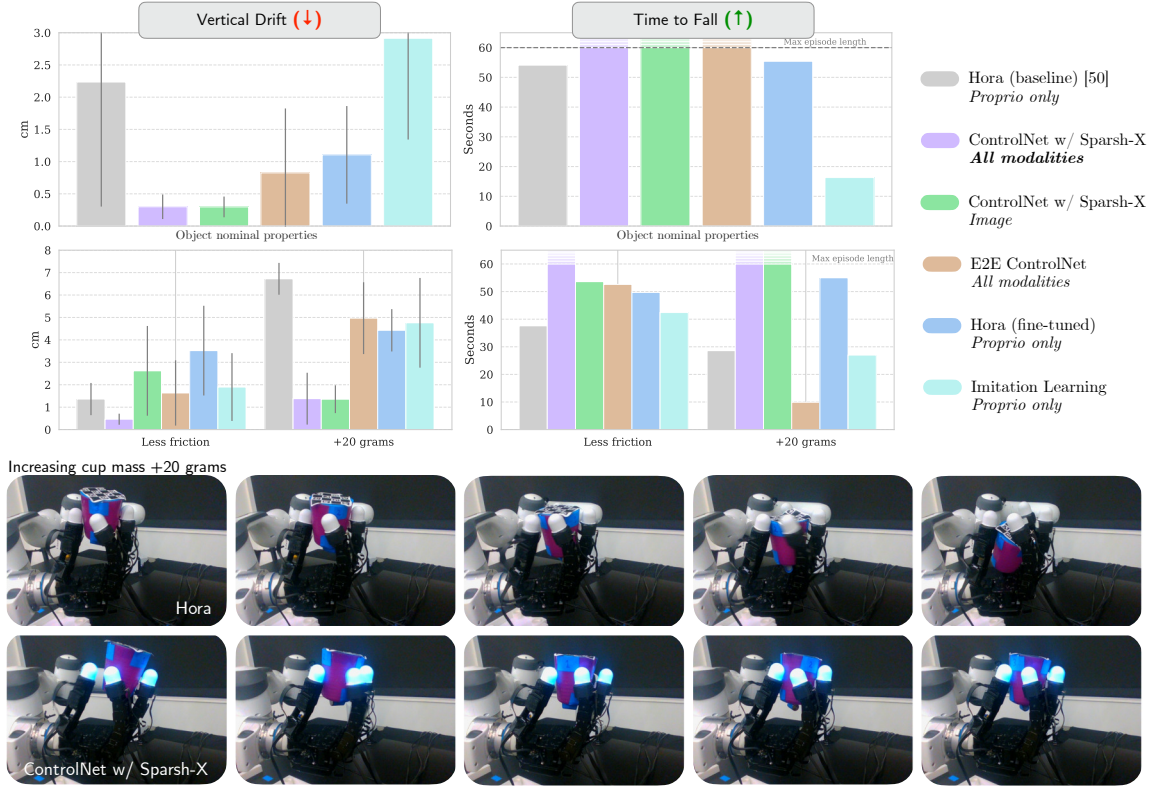


Figure 5.6: **Top.** For object nominal properties, tactile adaptation with **Sparsh-X** reduces vertical drift by 90% compared to Hora. Fine-tuning with successful rollouts does not yield same performance, highlighting the effectiveness of tactile adaptation. Metrics from 10 trials (60s episodes). **Middle.** Under dynamical changes, tactile adaptation shows superior stability than Hora variants. Metrics from 5 trials (60s episodes). **Bottom.** Snapshots of policy rollouts with and without multisensory touch input as object mass increases.

tion [171, 93], insertion [40], and contour-following [5, 196] among others. While tactile sensing has been primarily vision-based, other modalities such as audio have also been used independently for capturing object properties [58, 31] and dynamic behaviors [175], though audio alone may be insufficient for perceiving continuous interactions such as forces, deformations, and motion. Subsequently, prior work has also explored audio-visual learning [132, 123] as a natural extension. While audio and vision modalities augment the tactile state complementarily, additional modalities such as fingertip motion and accumulated pressure can provide additional information for detecting shear forces, recognizing object properties, and predicting object slip and pose changes.

Self-supervised learning (SSL) has been effective for developing tactile image representations [52, 72, 195, 87, 217], allowing better performance and data-efficiency in downstream tasks. Other approaches that use additional tactile modalities such as audio, often rely on using task-specific data to fine-tune pre-trained encoders (e.g. AST [68] and BYOL-A [139] for audio) or joint audio-visual encoders [134]. MULSA [116] introduced multimodal transformers integrating vision, tactile image, and audio from contact microphones by treating all signals as RGB images, but suffers from quadratic complexity of pairwise attention since it simply concatenates the tokens from all modalities. MimicTouch [205] proposed unimodal SSL for tactile images and audio separately, without explicit cross-modal fusion. In contrast, we propose **Sparsh-X**, a multisensory framework that fuses image, audio, motion, and pressure signals via bottleneck self-attention [136] into a shared latent space. This enables to capture contact properties for downstream tasks, while also reducing computational complexity compared to vanilla transformer-based concatenation strategies.

The Digit 360 sensor. Digit 360 [109] offers multisensory touch sensing in a compact fingertip form factor, making it well-suited for dexterous robotic hardware. Inside a dome-shaped elastomer, it integrates a hyper-fisheye camera, contact microphones, IMU, static pressure sensors, and more. While it marks a significant step toward standardizing touch sensing, its unique form factor introduces new challenges. For example, the soft hemispherical dome deforms and moves upon contact, complicating shear force estimation. Additionally, the use of a hyper-fisheye lens and directional lighting limits the applicability of

photometric surface reconstruction methods like Poisson integration [208]. Despite these limitations, learning touch representations through scale pretraining offers a promising path to overcome such challenges.

5.6 Discussion

We present **Sparsh-X**, a self-supervised backbone for general multisensory touch representations. Through both policy learning and supervised tactile experiments, we demonstrate that incorporating multisensory touch from Digit 360 and scale pretraining over $\sim 1M$ samples significantly enhances task performance compared to using end-to-end approaches with tactile images alone.

Our study is driven by two central research questions. First, *how can real-world touch representations be leveraged for manipulation policy learning?* **Sparsh-X** pretraining enables better and more robust policies. We explore two approaches: imitation learning (IL) and tactile adaptation of sim-trained policies. IL is naturally suited as demonstrations capture rich tactile signals from all modalities provided by Digit 360. For sim-policies, we propose tactile adaptation via ControlNet, enabling the propagation of tactile information previously accessible only in simulation as privileged information. We validate these approaches on two fundamental manipulation tasks: plug insertion and in-hand object rotation. Notably, **Sparsh-X** enhances policy performance by 63% over policies using tactile images alone and improves robustness by 90% by using touch to recover object state during manipulation.

Second, *what tactile properties do our representations capture?* We find that **Sparsh-X** representations effectively captures physical properties that allows to identify objects, actions, surfaces, estimate intrinsic properties and forces from multisensory touch signals. We evaluate **Sparsh-X** on a suite of supervised benchmark tasks common in the literature: object-action recognition, material-quantity estimation, and force prediction. We perform ablations over tactile input modalities and training data budgets to assess the impact of pretraining and multisensory fusion. Our results show that a synergy of touch from images, audio, IMU, and pressure, leads to higher accuracy across all tasks even in low-data regimes. Compared to training end-to-end with tactile images only, **Sparsh-X** achieves an average improvement of 48% across all tasks, demonstrating the benefits of both pretraining and

touch sensory fusion.

Limitations. While our study highlights the benefits of multisensory touch and self-supervised pretraining, some limitations remain. Each modality introduces its own challenges in capturing a broad and diverse set of contact interactions at scale. In our pretraining dataset, the tactile image modality from the Digit 360 sensor exhibits the lowest diversity in terms of number of different devices used with their own optical artifacts, potentially limiting its generalization in downstream performance. We believe that as the community increasingly adopts multisensory tactile sensors like the Digit 360, collaborative efforts can help build larger and more diverse datasets to support scalable pretraining. Additionally, our experiments focus exclusively on frozen **Sparsh-X** representations to understand the pure impact of pretraining on generalization. However, allowing fine-tuning with task-specific data could further improve performance and help compensate for modality-specific data limitations. Finally, our evaluation of force sensing is limited to normal force estimation under controlled contact conditions. Generalizing to varying contact geometries and multiple simultaneous contacts remains an open area for future work. Moreover, we do not consider shear force estimation in this study, as separating the effects of extrinsic forces from the internal deformation of the elastomer presents non-trivial modeling challenges.

Chapter 6

VISUO-TACTILE WORLD MODELS

6.1 Introduction

World models (WMs) have emerged as a leading paradigm in machine learning, offering robots the ability to understand the physical world and plan interactions in *imagination* [1, 157, 120]. In this work, we advance world models for robot manipulation by extending beyond purely visual imagination to incorporate modalities that directly ground contact interactions (Fig. 6.1). By complementing vision with touch, world models for robot manipulation gain access to local contact signals that anchor predictions in the physics of contact. Tactile sensing provides this crucial information, enabling the model to capture object permanence and force-driven motion, and to move beyond the ambiguities and aliasing of vision alone.

We introduce the first multi-task **Visuo-Tactile World Model** (VT-WM). Vision provides global context about the robot’s kinematics and the task scene, but it does not reveal the state of physical contact. Tactile sensing supplies this missing local signal, capturing how the hand and object actually interact. Together, these modalities enable the model to maintain object permanence even under heavy occlusion or visual aliasing. As shown in Fig. 6.1, VT-WM consistently represents the cube throughout the phases of a stacking task: in-hand during transport and placement, and back in the scene once released. Multimodality also disambiguates visually similar states with different outcomes. For example, from the camera view the robot hand may appear to rest on a cloth, yet only tactile feedback can reveal whether contact is sufficient for the cloth to move when wiping, or if it will remain in place. By grounding imagination in both global vision and local touch, VT-WM preserves object permanence and predicts object–robot interactions that respect the laws of motion.

We evaluate the gains of VT-WM over V-WM on a set of contact-rich manipulation tasks. Our first focus is how visuo-tactile training improves imagination by preserving ob-

ject permanence and adhering to physical motion laws during autoregressive rollouts. Tactile grounding helps prevent common hallucinations in vision-only models, such as objects disappearing under occlusion, teleporting, or moving without applied forces due to visual aliasing. We then assess how this grounding translates to planning. While V-WM and VT-WM perform similarly on reaching tasks that mainly test kinematic fidelity, zero-shot plans generated by VT-WM show a stronger ability to maintain contact in the real world. This capability proves crucial for manipulation actions such as pushing, wiping, and placing, where reliable hand-object interaction determines task success.

6.1.1 Contributions

To summarize, this chapter makes the following key contributions:

- [Section 6.2](#) described the architecture and training procedure of the multi-task visuo-tactile world model, integrating fingertip tactile sensing with vision to jointly model global context and local contact dynamics.
- [Section 6.3](#) shows that visuo-tactile grounding substantially improves imagination quality in terms of object permanence and compliance with physical laws.
- [Section 6.5](#) demonstrates that contact grounding in imagination enable more reliable zero-shot planning on real contact-rich tasks.

6.2 World Models that Understand Contact

Vision-only world models have shown promising capabilities in action steerability and spatial reasoning [8, 1], generating plausible rollouts with reasonable robot kinematics and high-quality visuals. These properties make them useful for planning free-space motions. However, simulating object interactions comes with some challenges. Object motion depends on forces invisible to exocentric cameras, and occlusions during grasping, pushing, or placing often cause artifacts such as teleportation, disappearance, or physically implausible dynamics.

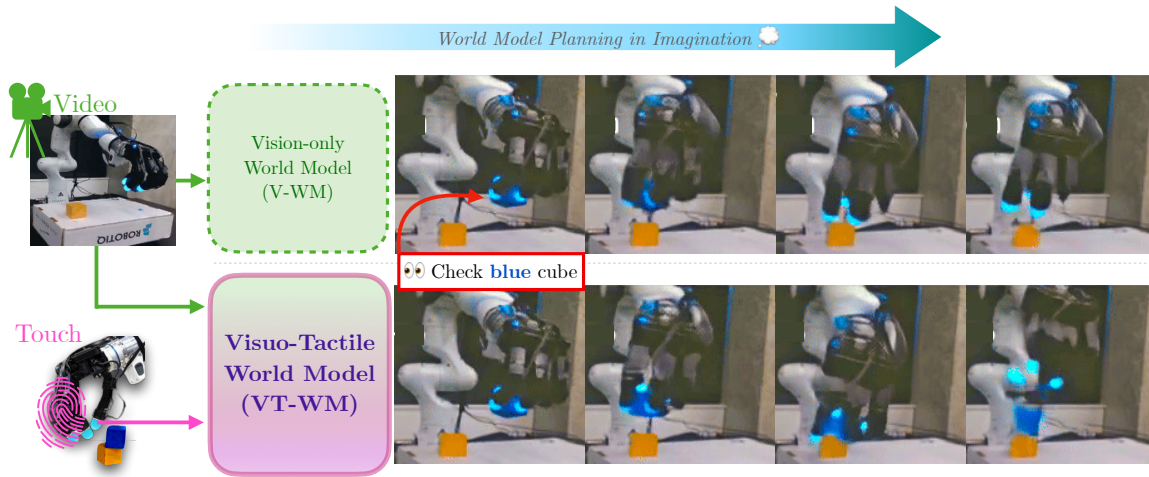


Figure 6.1: Visuo-Tactile World Model complements vision with touch, providing contact grounding of robot-object interactions. Notice that when using the WMs for planning a cube stacking task, the VT-WM has notion of object permanence of the **blue cube** when transporting, placing and releasing the object. The contact grounding provided by the vision-based tactile sensor helps to reduce hallucinations often present in V-WMs, enabling more reliable zero-shot planning in contact-rich manipulation tasks.

We introduce **Sparsh** (VT-WM), which uses tactile sensing to complement vision to overcome these limitations. Touch provides contact information during occlusion, grounding the model’s imagination in contact physics and producing more accurate rollouts for contact-rich manipulation tasks.

6.2.1 What vision doesn’t see: Sensing contact with touch

Touch provides essential local perception, enabling robots to distinguish properties like stiffness, friction, and roughness that are difficult to infer from vision alone. It also captures the dynamics of contact, which is crucial for manipulation tasks. For instance, when manipulating an object in-hand, touch provides context about forces, slip, and subtle pose changes.

Vision-based tactile sensors typically stream image data at 30-60 FPS, providing rich information about the contact area, including force, shape, and texture features. This tactile information is crucial for disambiguating contact states that from an exocentric camera may

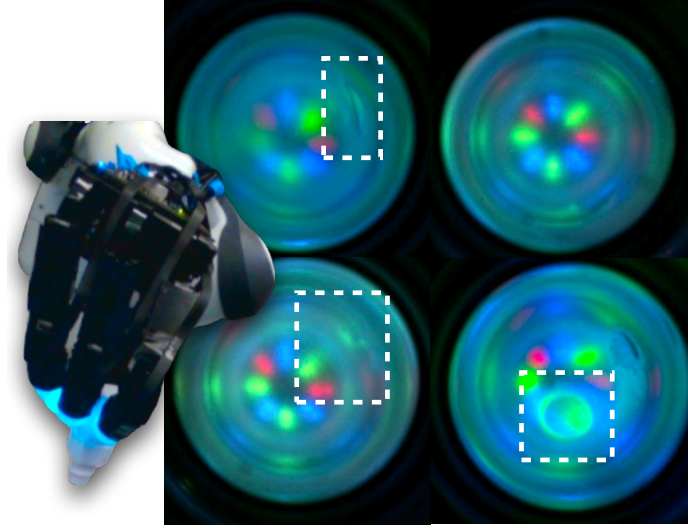


Figure 6.2: Tactile images from Digit 360 sensors. White boxes highlight contact while the hand holds a screw.

appear visually similar. For example, a robot hand’s grasp on a cup might look the same from a distance, but tactile sensing can differentiate between a no-contact state, a subtle touch, or a firm grasp. In this work, we use Digit 360 sensors as fingertips for an Allegro Hand mounted on a robot arm. A visualization of tactile images captured by a Digit 360 sensor is shown in [Fig. 6.2](#).

6.2.2 MultiTask Visuo-Tactile World Model

Model Architecture

Our visuo-tactile world model is designed to address a key challenge in multimodal robot world models: *how to combine exocentric vision with tactile sensing in order to generate consistent imagined futures*. As shown in [Fig. 6.3](#), the architecture consists of three main components: a *vision encoder*, a *tactile encoder*, and an autoregressive *predictor*.

The vision encoder extracts latent states s_k that capture the robot and its environment from exocentric video. The tactile encoder compresses high-frequency contact feedback into a compact state t_k that emphasizes salient physical interactions. These representations are fused with control actions and passed to the predictor, a forward dynamics model that

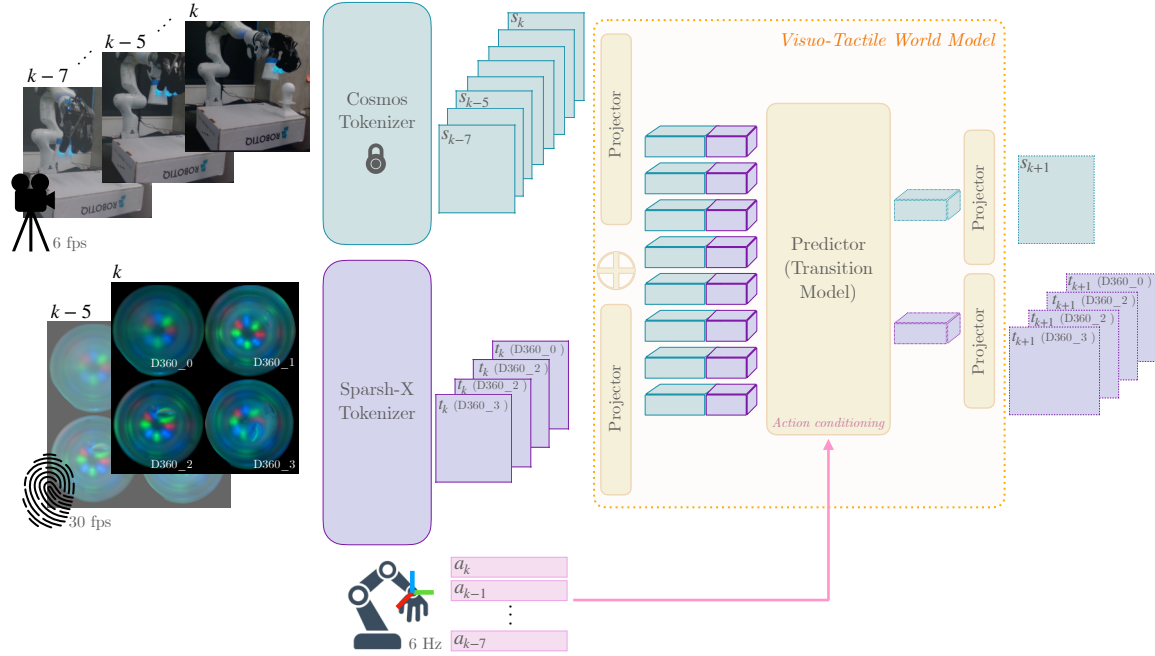


Figure 6.3: **Visuo-Tactile World Model.** Vision (s_k) and tactile (t_k) latents, obtained from Cosmos and Sparsh-X encoders, are processed by a transformer predictor given control actions a_k to generate next-step states (s_{k+1}, t_{k+1}) .

estimates the next-step states $(s_{k+1}, t_{k+1}) \sim P_\phi(s_k, t_k \mid a_k)$.

This formulation enables the model to *imagine multiple possible futures under control actions*. Unlike purely visual world models, our predictor leverages tactile signals to disambiguate perceptually identical visual states. For instance, two identical video frames of a robot hand around a cup can lead to different rollouts: if the tactile input indicates contact, the imagined sequence shows the cup being lifted; if not, the cup remains on the table. This tactile-informed disambiguation is significant for planning in contact-rich manipulation.

We frame the predictor as a supervised next-state estimation problem with ground-truth future latents as targets. Both modalities are encoded with pretrained networks: Cosmos tokenizer [1] for vision and Sparsh-X [88] for Digit 360 tactile sensors. The encoded contexts s_k and t_k are augmented with sinusoidal positional embeddings, and projected into a unified representation $\mathbb{R}^{(b,t,s,d)}$. Vision and tactile tokens are concatenated along the spatial dimension to form a unified input sequence. The predictor then processes these

multi-modal tokens through a 12-layer transformer that alternates between two types of attention mechanisms:

Spatio-Temporal Self-Attention The model processes vision-touch tokens through factorized attention that operates in two stages: **spatial** attention enables all tokens within a timestep to interact, while **temporal** attention tracks how each token evolves across past timesteps. This factorization efficiently captures both local dynamics and global context while avoiding the $O((THW)^2)$ complexity of full spatiotemporal attention. Action tokens undergo the same factorized attention process.

Action Conditioning via Cross-Attention After each self-attention block, vision-touch tokens cross-attend to action tokens to incorporate the robot’s control inputs into the predictions. This alternating pattern of self-attention and cross-attention allows the model to iteratively refine its latent states based on both sensory observations and executed actions.

All attention layers employ Rotary Position Embeddings (RoPE) [166] for relative position encoding. After the transformer, the representations are projected back to their original dimensions through modality-specific output heads, yielding predictions s_{k+1} and t_{k+1} .

Training visuo-tactile world model

During training, the vision input is a 1.5-second exocentric videoclip (9 frames at 6 fps, 320×192 resolution), encoded framewise with Cosmos. The tactile input consists of two frames per Digit 360 sensor (four sensors total), covering the most recent 0.16 seconds. This shorter horizon reflects the higher temporal frequency and local nature of contact information, which complements the slower, global context provided by vision. The action input includes changes in proprioceptive state (translation, quaternion rotation) and a binary hand state representing pre-set open/close configurations. We chunk action sequences from 30Hz into groups of 5, with the full chunk of delta-states provided to the predictor. This combination ensures that the predictor models both the external scene and the internal actuation history, a prerequisite for accurate visuo-tactile imagination. The model uses a maximum context length of 9 frames for both vision and touch modalities. Additional details about training

dataset and hyperparameters are provided in Appendix D.1.1 and Appendix D.1.2.

The model is trained with an objective that combines teacher forcing with autoregressive sampling to balance training stability with long-horizon coherence as proposed by [8].

Teacher Forcing Loss. The primary training signal comes from next-step prediction with ground-truth context. Given a sequence of T frames, we compute:

$$L_{teacher} = \sum_{k=1}^{T-1} \|\hat{s}_{k+1} - s_{k+1}\|_1 + \|\hat{t}_{k+1} - t_{k+1}\|_1 \quad (6.1)$$

where $\hat{s}_{k+1}$ and $\hat{t}_{k+1}$ are predicted from ground-truth states up to time k , and s_{k+1} and t_{k+1} are the encoded latents given ground truth observations at time step $k + 1$. This provides dense supervision and stable gradients but can lead to distribution shift during autoregressive rollouts.

Sampling Loss. To improve long-horizon generation, we additionally train on sampled trajectories. During training, we sample future states autoregressively for H steps (typically $H = 3 - 5$), then compute predictions conditioned on these sampled states:

$$L_{sampling} = \sum_{k=1}^H \|\hat{s}_{k+1}^{sampled} - s_{k+1}\|_1 + \|\hat{t}_{k+1}^{sampled} - t_{k+1}\|_1 \quad (6.2)$$

The sampled states are generated without gradients to prevent training instability. The final loss combines both objectives with equal weighting: $L = L_{teacher} + L_{sampling}$.

In next section, we go over our experimental framework which evaluates the advantages of visuo-tactile world model (VT-WM) for robot manipulation by addressing the key questions:

- *Contact Perception:* Do VT-WMs better capture object permanence and causal compliance than vision-only WMs, and generate futures consistent with action conditioning?
- *Zero-shot Planning:* Does improved contact perception lead to more reliable zero-shot plan transfer in open-loop execution?
- *Data Efficiency:* Given limited demonstrations, how does fine-tuning a multi-task world model for planning compare to behavioral cloning?

6.3 Contact Perception of Visuo-Tactile World Model in imagination

To evaluate the benefits of incorporating touch, we compare rollouts from a multi-task vision-only world model (V-WM) and our multi-task visuo-tactile world model (VT-WM), conditioned on the same actions and context. The action sequences are drawn from successful demonstrations on the real system, which enables a direct comparison between each model’s rollouts and the corresponding ground-truth videos. This setup allows us to assess how well the models capture motion dynamics and the physical plausibility of object interactions. We focus our evaluation on object permanence, causal compliance, and action controllability. These metrics are components of the World Consistency Score [152], a proposed benchmark in the state-of-the-art to evaluate a generative model’s ability to maintain coherent and physically plausible futures over time.

6.3.1 Object Permanence

This metric assesses a model’s ability to maintain a consistent representation of an object’s existence and state even when the object is temporarily occluded. As shown in Fig. 6.5, we evaluate whether objects remain represented during heavy occlusion (e.g., during a grasp) and reappear in the correct state once revealed. In the cube-stacking task, VT-WM exhibits stronger object permanence than V-WM: as the blue cube is occluded in the hand during transport and placement, VT-WM preserves its representation, and upon release, the cube re-emerges in the imagined scene at the correct location above the yellow target cube. In Appendix D.2-Fig. D.3, we visualize the tactile predictions generated by VT-WM, demonstrating the model’s ability to maintain congruent visual and tactile representations of object contact.

For quantitative evaluation, we employ CoTracker [97] to track keypoints on the object, providing pixel-level visibility and trajectories. We then compare the normalized Fréchet distance between the ground-truth visual trajectory and the one imagined by V-WM and VT-WM under the ground-truth action conditioning. To ensure comparability, trajectories are expressed relative to the object’s initial image position and normalized by the length of the ground-truth trajectory. A lower Fréchet distance indicates that the imagined trajectory

more closely reflects the real motion and state of the object, thereby capturing the physical coherence required for object permanence.

Fig. 6.4 reports normalized Fréchet distances across five tasks. To assess statistical significance, we complement the Fréchet distance analysis with paired t -tests across tasks. VT-WM achieves consistently lower distances than V-WM, with statistically significant improvements in *place fruits* ($t = 4.38, p < 0.001$), *push fruits* ($t = 6.06, p < 10^{-6}$), and *cube stacking* ($t = 2.40, p < 0.05$). For *wipe with cloth* and *scribble with marker*, the differences follow the same downward trend but do not reach significance at the 5% level. Quantitatively, VT-WM reduces normalized Fréchet distance by 18–47% across all five tasks, corresponding to an average overall reduction of $\approx 33\%$. These results indicate that VT-WM produces more physically coherent rollouts, with statistically significant gains in tasks requiring reliable object permanence, such as object pushing and stacking.

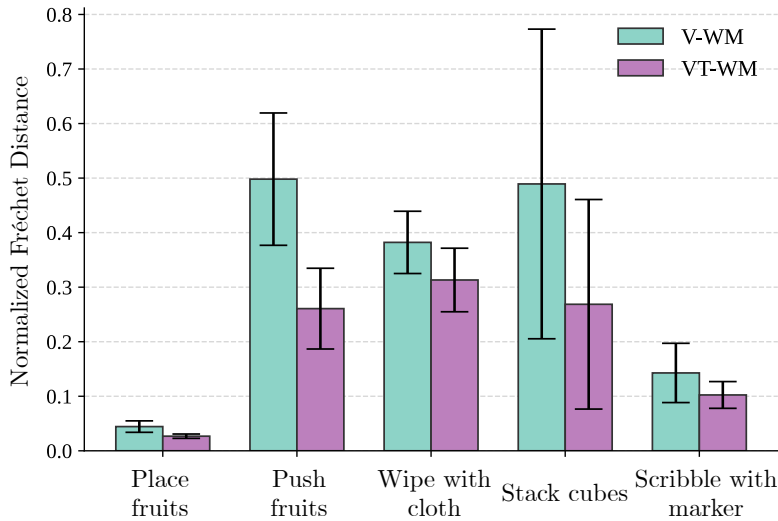


Figure 6.4: *Object permanence*. VT-WM achieves an average reduction of $\approx 33\%$ relative to V-WM (with 95% CI) of the normalized Fréchet distances for objects in motion.

6.4 Causal Compliance

This metric evaluates whether changes in object state occur as physically plausible consequences of the robot’s actions. A causally compliant model predicts that an object’s state

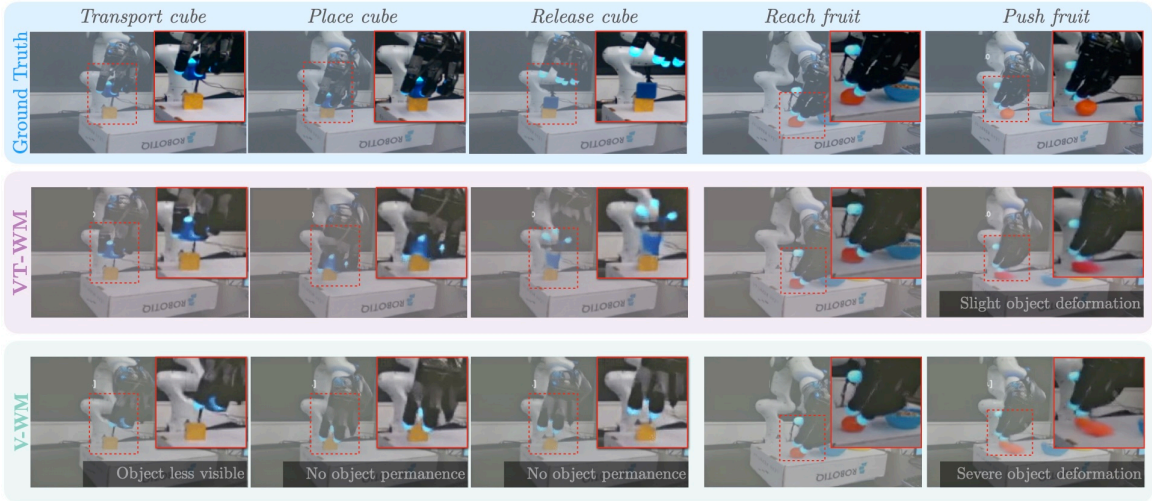


Figure 6.5: VT-WM preserves object permanence and consistent hand-object interactions during imagination, while V-WM often loses objects or produces severe deformations.

changes only when it is subject to external forces. Assessing causal compliance is essential for developing physics-informed world models that respect the principles of contact dynamics and avoid unrealistic motions or deformations.

For a quantitative measure of causal compliance, we use CoTracker to compute the trajectory error of keypoints on objects in the scene that are not subject to any external force and should therefore remain stationary. We again use the normalized Fréchet distance between ground-truth and imagined rollouts as our metric. A higher Fréchet distance indicates that the world model hallucinates changes in the position or deformation of these passive objects, thereby violating basic physical laws such as Newton’s first law of motion. As shown in Fig. 6.6, VT-WM consistently achieves lower distances than V-WM across most tasks. Paired t -tests confirm statistically significant improvements at the 5% level in *place fruits* ($t = 3.66, p < 0.001$), *push fruits* ($t = 2.28, p < 0.05$), and *wipe with cloth* ($t = 2.99, p < 0.01$), while differences in *cube stacking* ($t = 1.75, p = 0.09$), and *scribble with marker* ($t = -1.22, p = 0.23$) are not significant. Quantitatively, VT-WM achieves relative reductions of 43.6%, 16.4%, and 66.1% in *place fruits*, *push fruits*, and *wipe with cloth*, respectively, alongside a smaller improvement in *cube stacking* and a degradation in *scribble with marker*. Overall, VT-WM reduces hallucinated motion by an average of $\approx 29\%$

across tasks, reflecting stronger causal compliance in most scenarios.

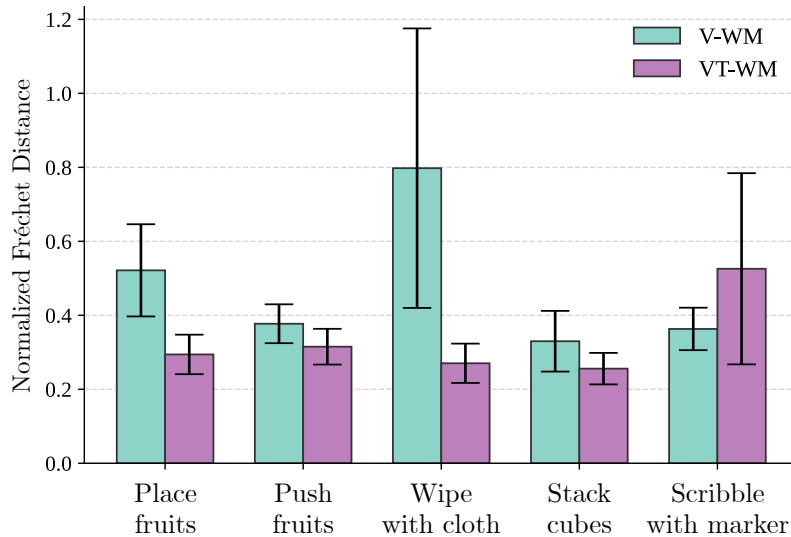


Figure 6.6: *Causal compliance* evaluates WMs adherence to the laws of motion. Normalized Fréchet distance for static objects (95% CI) shows that VT-WM outperforms V-WM, with an overall improvement of $\approx 29\%$.

In Fig. 6.7 we show snapshots of a trajectory where the robot performs a wiping motion just above a cloth, without making contact. In the ground-truth sequence (top row), keypoints on the cloth remain stationary. In contrast, the V-WM’s rollout (bottom row), conditioned on the real actions, shows significant displacement of keypoints and deformations of the cloth. This highlights the V-WM’s difficulty to distinguish between contact and non-contact states based on visual input alone. The VT-WM’s rollouts (middle row), however, exhibit fewer artifacts and less variation, demonstrating the advantage of tactile sensing in providing the world model with critical physical grounding.

In Appendix D.2 we showcase the action controllability of the world model. We compare ground-truth trajectories with VT-WM rollouts under the same action sequences and illustrate what the world model *imagines* in terms of contact.

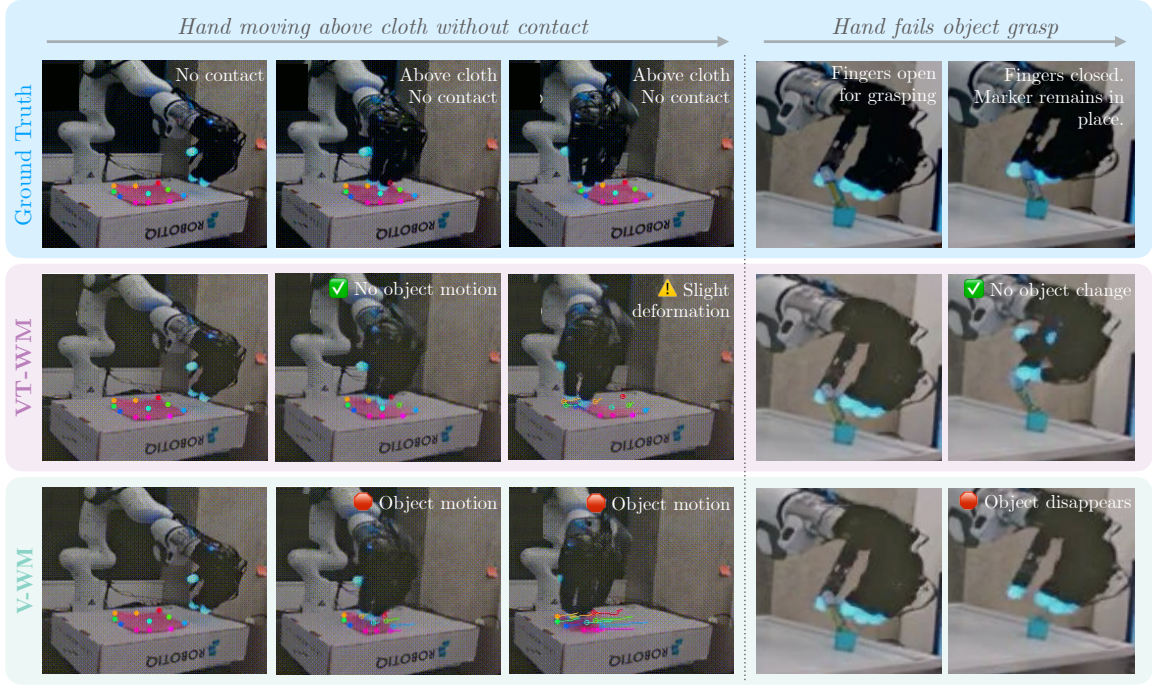


Figure 6.7: Comparison of rollouts, illustrating that VT-WM prevents spurious motion of objects not subject to forces, whereas V-WM often hallucinates unintended displacements.

6.5 Zero-Shot Planning with Visuo-Tactile World Model

6.5.1 Planning in Imagination

The action-conditioned nature of our predictor enables the use of the visuo-tactile world model as a simulator within a Cross-Entropy Method [156] (CEM). At each step, the planner samples a population of action sequences $\{a_{k:k+H}^i\}_{i=1}^N$ over a horizon H . For each sequence, the predictor autoregressively generates future latents $(s_{k+1:k+H}, t_{k+1:k+H})$. A cost function, defined by energy minimization with respect to a goal image, assigns a score to each trajectory. In practice, this cost can be as simple as an ℓ_2 distance between the final predicted visual latent s_{k+H} and the latent of the goal image s_{goal} . CEM then selects the top-performing fraction of sequences, updates the sampling distribution toward them, and iterates until convergence. The best sequence is then executed on the real robot in an open-loop manner. Algorithm 1 provides an algorithmic description for planning via CEM and the world model dynamics.

We do not provide the tactile modality as a goal signal, thus the planning objective remains purely vision-based. The role of tactile in the VT-WM is to enhance the reliability of the learned world model and, therefore, to improve planning indirectly. First, tactile feedback during training enables the world model to capture contact physics that are difficult to infer from vision alone. Second, when generating rollouts, tactile context in the initial state helps disambiguate visually identical observations (e.g., distinguishing whether the robot is already in contact with an object). This yields more physically consistent imagined futures and more accurate cost evaluations, which we hypothesize translate into higher-quality plans.

6.5.2 Zero-shot planning transfer to real-world

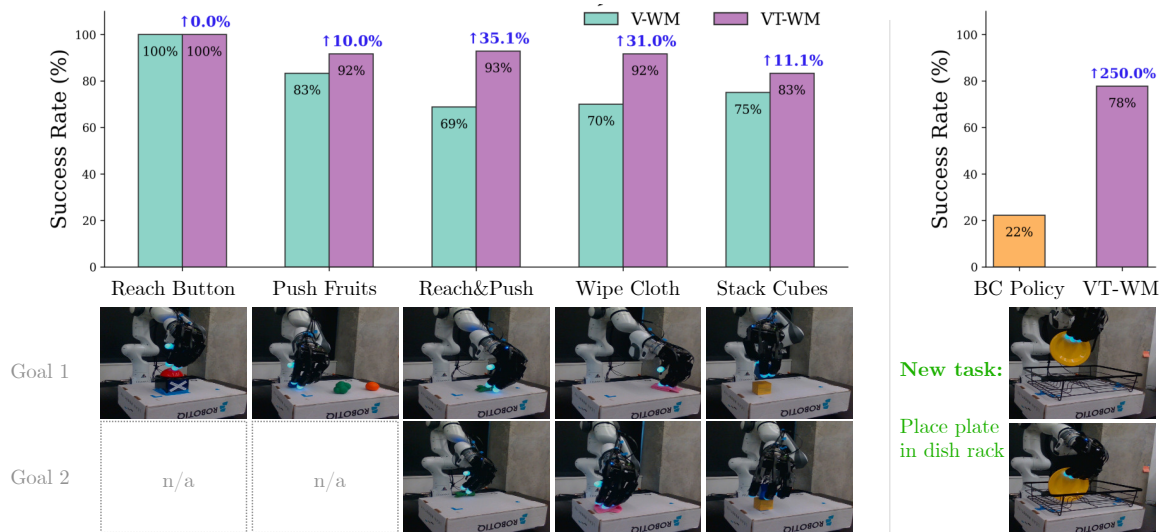


Figure 6.8: *Left:* Success rate of plans via CEM with VT-WM and V-WM on real robot. For all tasks the VT-WM achieves equal or better performance (blue labels), empirically demonstrating the better planning capability with contact grounding via tactile sensing. *Right:* success rate on new task highlights the data efficiency of VT-WM compared to classical behavioral cloning (BC) policies.

Does the improved contact perception of VT-WM translate into superior planning performance on a real robot? We hypothesize that VT-WM produces more effective plans for contact-rich tasks. For example, in a cube-stacking task, a physically grounded model

should avoid opening its hand while transporting a cube. Similarly, for pushing, the model must recognize that contact is a prerequisite for object motion.

To evaluate this, we employ the Cross-Entropy Method (CEM) [156, 39] to solve a goal-conditioned energy minimization problem. The objective is to plan an optimal action sequence over a fixed horizon H , where the cost is the distance in latent space between the final predicted visual state s_{k+H} and the goal image latent s^{goal} . The search space for CEM is $\mathbb{R}^7$, consisting of 3D translation and 3D orientation of the wrist pose, plus a binary variable for the hand’s open/closed configuration.

We evaluate open-loop zero-shot transfer of the generated plans on the real robot. To initialize planning consistently, the initial RGB and tactile embeddings are passed as context to the world model, indicating whether optimization should begin from an in-contact or no-contact state. Both VT-WM and V-WM are tested on five tasks of increasing difficulty: *reach button*, *push fruits*, *reach & push*, *wipe cloth*, and *stack cubes*. The first two are single-goal tasks, while the latter three involve multiple subgoals.

Fig. 6.8(left) reports success rates, averaged over five trials per task from distinct initial conditions. The results confirm the superior planning capability of VT-WM across all tasks, supporting our hypothesis that a contact-aware model generates more effective plans. On simple tasks such as *reach button*, both models achieve 100% success, consistent with prior visual world models for robot manipulation such as V-JEPA-2AC [8]. However, the benefits of tactile input become increasingly evident in contact-rich tasks: VT-WM improves success rates by 10% on *push fruits*, 35% on *reach & push*, 31% on *wipe cloth*, and 11% on *stack cubes*. These gains are most pronounced in multi-step tasks involving sustained contact, where vision alone is insufficient to inform about the object state during planning. In Appendix D.3.2, we describe the experimental setup for each task and present a qualitative evaluation of the planned trajectories and their corresponding real-world executions.

6.5.3 CEM Planning via WM vs Behavioral Cloning

For a new task with a limited number of successful demonstrations, *how does fine-tuning a multi-task world model for planning compare to training a task-specific behavioral cloning*

(BC) policy? We hypothesize that VT-WM can extract task structure even in low-data regimes, since it already encodes contact dynamics from prior tasks. For example, insertion may involve a new object, but concepts such as alignment and adjusting contact with a receptacle can be reused. In contrast, a BC policy must learn both spatial and contact reasoning from scratch.

We collect 20 demonstrations of the task “place plate in the dish rack,” which requires transporting the plate and inserting it between the racks. We augment our multi-task dataset (see Appendix D.1.1) with the new sequences and continue training VT-WM, while also training a task-specific BC policy ACT [220] that outputs action chunks over a fixed horizon. For VT-WM, we use CEM planning and zero-shot transfer to the real robot. The task is divided into two subgoals: alignment and insertion (see Fig. 6.8). The BC policy is deployed in closed loop, where at each timestep it receives the latest RGB and tactile inputs and executes the first action of the predicted chunk.

We evaluate both methods in nine real-robot trials, randomizing the initial plate-in-grasp pose. As shown in Fig. 6.8(right), VT-WM planning achieves a 77% success rate, compared to 22% with BC. These results highlight the data efficiency of multi-task world models and their advantage over task-specific policies. Moreover, failure modes differ, for instance, VT-WM mostly places the plate beside the rack, whereas 57% of BC failures involve the robot never reaching the rack at all.

6.6 Related Work

In this chapter we aim to train general (multi-task) world models that can leverage both tactile and visual observations. Specifically we train latent-state world models, which first project observations into latent representations and then train an action-conditioned dynamics model in that latent space.

Foundational encoders for vision and touch: Unlike the established hardware platforms for computer vision, the field of robotic tactile sensing lacks a single standardized sensor. Nevertheless, vision-based tactile sensors have emerged as one of the most prominent solutions. Devices like GelSight [207], Digit [107], and the more recent Digit 360 [110]

capture tactile information by imaging the deformation of a soft elastomer surface.

Similar to the development of general-purpose visual encoders like CLIP [151], DINO [22, 140], I-JEPA [6], and Cosmos Tokenizer [1], recent efforts have focused on creating foundational models for vision-based tactile sensors through self-supervised learning. Models such as SITR [73], T3 [217], UniT [195], Sparsh [87] (see Chapter 3), and Sparsh-X [88] (see Chapter 5) learn robust, low-dimensional tactile representations without requiring explicit labels. Benchmarks like TacBench [87] evaluate the quality of these embeddings, demonstrating their ability to compress information about contact dynamics, including force fields, slip states, and pose changes, as well as static properties like texture and material. In this work, we use Digit 360 sensors as fingertips for an Allegro Hand mounted on a Franka Panda arm and we use the Sparsh-X model [88] to obtain tactile embeddings, and the Cosmos encoder [1] to obtain RGB embeddings.

Action-Conditioned World Models for Real World Robotics: There has been an influx of training general purpose action-conditioned video-generation models [90, 157, 200, 15, 1, 8]. However, these lines of work focus on show-casing the generation capabilities, and only have limited (if any) results for using these models to control robots.

The majority of previous work on world models applied to real world tasks focuses on visual dynamics models [2, 17, 37, 135, 53, 51, 50, 201]. Visual dynamics models are either trained directly in pixel-space [53, 51, 50, 201, 4], or in a learned latent space [187, 2, 75, 76, 137, 190, 177, 91, 112], or in more structured representation spaces such as keypoint representations [129, 37] or tracked 3D states [135]. In our work, we train action-conditioned world models in latent states extracted from both RGB and tactile observations.

While several works have investigated learning dynamics model on touch observations [172, 176, 3] there is little work on training world models with vision and touch [214]. Furthermore these dynamics models are task-specific - while in our work we aim to train general purpose multi-modal world models that can be used for visual MPC on multiple tasks.

6.7 Discussion

Visuo-Tactile World Model (VT-WM) leverages tactile sensing to complement vision, enabling world models for robot manipulation to be grounded in contact. While vision-only world models (V-WMs) have shown promise in spatial reasoning and capturing robot kinematics, they are prone to hallucinate object interactions, with failures such as object disappearance, teleportation, or unrealistic deformations. By integrating fingertip tactile sensing with exocentric vision, VT-WM grounds imagination in the physics of contact, producing more accurate rollouts and capturing core concepts such as object permanence under occlusion and compliance with physical laws.

We studied the gains of multimodality in VT-WM over V-WM through three key questions. First, *does adding touch improve a world model’s understanding of contact?* Comparing autoregressive rollouts under identical action sequences, we found that VT-WM preserved object permanence, even when objects were occluded by the robot hand, and maintained resting states for objects not subject to external forces. Quantitatively, VT-WM reduced normalized Fréchet distance by 33% on average relative to V-WM, better reflecting true object dynamics.

Second, *does contact grounding improve planning?* Using CEM-based imagination for goal-conditioned planning, we zero-shot transferred plans to a real robot under randomized initial conditions. While both models performed similarly on free-space tasks such as reaching, VT-WM achieved up to 35% higher success rates in contact-rich tasks requiring precise hand-object interaction, including pushing, wiping, and stacking.

Third, *is VT-WM data efficient when targeting a new task?* To compare with task-specific behavioral cloning (BC), we fine-tuned VT-WM on just 20 demonstrations of a plate-insertion task. VT-WM reached a 77% success rate, over $3\times$ higher than BC, by reusing priors from previously learned contact-rich tasks such as alignment and insertion. This highlights its ability to efficiently adapt to new tasks with limited data.

Limitations. While our results demonstrate clear benefits of visuo-tactile world model, following limitations point to promising directions for future research. First, our tactile

modality is limited to vision-based tactile sensing, specifically the Digit 360 sensor. However, the VT-WM framework applies to other tactile modalities as well, provided that an appropriate pretrained tactile encoder is available. Second, evaluation of contact perception uses unseen robot trajectories but only within tasks from the training distribution, leaving open the question of how well the model generalizes to entirely novel manipulation tasks or object characteristics. Third, our planning experiments randomize initial robot states but are limited to the same scene and objects, without testing generalization to objects with different visual or physical properties such as size, shape, or color. Planning with world models via CEM remains computationally expensive, as it requires generating many autoregressive rollouts per particle. This leads to open-loop execution in trajectory chunks, unlike classical policies that operate in closed-loop at higher control frequencies. Finally, our comparison against a single task behavior cloning (BC) policy does not fully rule out the possibility that a multi-task BC policy could also exhibit strong data efficiency for the new task.

Chapter 7

CONCLUSION

In this thesis, we demonstrated the importance and advantages of the role of touch in robot perception for manipulation. Touch is the channel towards dexterous robots by proving the hand-object observability that frequently vision systems cannot capture. We argued that one of the major bottlenecks in touch for robotics has been the reliance on task-specific, hand-engineered features that fail to generalize across different tasks, sensors and objects. To this end, we developed a suite of self-supervised representation learning frameworks—spanning optical, magnetic, and multisensory modalities—that allow robots to perceive, ground, and eventually *imagine* the physical world through the lens of contact.

We began this journey by addressing the problem of tracking general object-environment contacts through touch perception in a neural representation. In [Chapter 2](#), we introduced Neural Contact Fields (NCF), a method that moves beyond assuming simple contact types by tracking the probability and geometry of extrinsic contacts on an object’s surface. By proving that latent spatial descriptors help to generalize contact tracking across novel and complex object shapes, we established that a robot’s understanding of “where” it is touching is a prerequisite for “how” it should move. This was validated by our findings that explicit contact estimation significantly boosts the success rates and speed of high-precision policy tasks like the insertion of household objects.

Moving from the geometric interpretation of touch, we then sought to create a more universal *language for touch* that could scale across diverse sensor types. In [Chapter 3](#), we presented **Sparsh**, a general-purpose backbone for vision-based tactile sensors. By shifting from pixel-space to latent-space supervision, we demonstrated the versatility of the learned tactile representations, as well as its robustness to raw signal artifacts. This work showed that a single pre-trained model could solve a vast array of downstream tasks, from slip detection to imitation learning, with remarkably little labeled data. We extended this phi-

losophy of sensor-agnostic representation in [Chapter 4](#) to magnetic skins on dexterous hands. Through **Sparsh-skin**, we proved that the principles of self-distillation and tokenized masking are effective even for non-optical morphologies, providing a scalable pathway toward foundation models for full-hand tactile perception.

As manipulation tasks grow in complexity, a single modality often becomes insufficient for resolving the ambiguities of the physical interaction. In [Chapter 5](#), we moved toward a holistic multisensory approach with **Sparsh-X**. By fusing high-fidelity fish-eye tactile images with audio, IMU, and pressure signals, we enabled robot hands to recover from manipulation failures that would be invisible to vision or tactile images alone. This synergy allowed for a significant improvement in policy robustness for tasks like insertion and in-hand rotation. Our policy experiments with **Sparsh-X** show tactile adaptation as a viable path for robots to maintain a good estimate of an object’s state and recover from failures during contact-rich interactions prone to finger occlusions.

Finally, in [Chapter 6](#), we synthesized these tactile perception capabilities to enhance the fidelity of world models for robot manipulation. We introduced Visuo-Tactile World Models (VT-WM), a framework wherein tactile representations complement exocentric vision to ground a robot’s internal imagination in the physical realities of contact. This work demonstrates that while vision provides global context and semantic scene understanding, touch serves as a critical supervisor that enforces physical laws during interaction (e.g. object permanence and reaction to forces). By grounding world models in touch, we enable the generation of future rollouts that are significantly more faithful to the physical world. This increased fidelity translates directly into better zero-shot planning, effectively bridging the gap between low-level multisensory sensing and high-level cognitive reasoning. Collectively, the contributions of this thesis provide a roadmap for evolving robot manipulation into integrated visuo-tactile agents capable of interacting with the world with true tactile intelligence.

To conclude, I would like to outline two exciting directions for future research.

Reflexive and Grounded Manipulation. A compelling frontier for future research lies in the formal integration of touch sensing into a dual-process framework for robot decision

making, analogous to the System 1 (fast) and System 2 (slow) pathways of human thought. This bifurcated approach would allow for the integration of Tactile System 1, a fast-acting reflexive loop that provides the kind of dynamic, fluid manipulation necessary for handling unpredictable objects in real-time. Just as humans subconsciously adjust their grip force upon detecting a micro-slip, robots require a light-weight model that can bypass heavy deliberative reasoning models to provide object stability. By embedding touch into these low-latency responses (e.g. reacting to sudden changes in center of mass, shear forces, or contact instability) we move towards a system of robot *reflexes* that ensures in-hand object stability without the need for high-level computation.

Conversely, touch must be more deeply integrated into the slower, reasoning-heavy System 2 to serve as a critical veridical anchor for high-level planning. While vision and language provide the global context and semantic strategy in WMs and VLAs, touch provides the ground-truth observability of the object’s hidden state, such as confirming the internal contents of a container or the friction of a novel surface. Integrating touch into System 2 reasoning will enable robots to perform long-horizon tasks with a sense of physical common sense, allowing them to pause, reassess, and adapt when the felt reality contradicts their visual or logical expectations.

Physical Grounding at Scale. A second critical direction involves the evolution of generative world models from visually compelling simulators to physically consistent engines for robot planning. Nevertheless, these models often lack the contact grounding necessary to respect the laws of physics, which could lead to unrealistic plans for manipulating objects in the physical world. Without a tactile anchor, generative models frequently hallucinate non-physical behaviors, such as objects deforming unnaturally, robot effectors penetrating rigid surfaces, or items teleporting across the scene. In the robotics context, these hallucinations are not merely visual glitches; they represent a fundamental flaw that downstream planners can inadvertently exploit, leading to policies that fail when transfer to the real world.

This dissertation has provided initial evidence that touch can force a model to respect the laws of motion, for instance, ensuring that an object at rest remains at rest unless a force is applied. However, significant research is required to scale these visuo-tactile

world models. The challenge lies in training models that move beyond video generation to capture true interaction dynamics across diverse robot embodiments, scenes and task intents. Investigating how to scale data collection efforts and training will be essential for enabling faithful offline policy evaluation. By stress-testing a robot’s dexterity in a grounded WM’s imagination, we can develop a level of trust in offline results that has previously been impossible, ensuring that the strategies learned in simulation hold true in the complexities of physical reality.

BIBLIOGRAPHY

- [1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qingsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.
- [2] Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016.
- [3] Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Cheston Tan, Yunzhu Li, and Jiajun Wu. Robopack: Learning tactile-informed dynamics models for dense packing. *arXiv preprint arXiv:2407.01418*, 2024.
- [4] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.
- [5] Kirsty Aquilina, David AW Barton, and Nathan F Lepora. Tactile control for object tracking and dynamic contour following. *Robotics and Autonomous Systems*, 178:104710, 2024.
- [6] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. [Self-supervised learning from images with a joint-embedding predictive architecture](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

- [7] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629, June 2023.
- [8] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.
- [9] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023.
- [10] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-JEPA: Latent video prediction for visual representation learning. 2023.
- [11] Maria Bauza, Antonia Bronars, and Alberto Rodriguez. Tac2Pose: Tactile object pose estimation from the first touch. *The International Journal of Robotics Research*, 42(13):1185–1209, 2023.
- [12] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. [Reskin: versatile, replaceable, lasting tactile skins](#). In *5th Annual Conference on Robot Learning*, 2021.
- [13] Raunaq Bhirangi, Venkatesh Pattabiraman, Enes Erciyes, Yifeng Cao, Tess Hellebrekers, and Lerrel Pinto. [Anyskin: Plug-and-play skin sensing for robotic touch](#). *arXiv preprint arXiv:2409.08276*, 2024.
- [14] Raunaq Bhirangi, Chenyu Wang, Venkatesh Pattabiraman, Carmel Majidi, Abhinav Gupta, Tess Hellebrekers, and Lerrel Pinto. [Hierarchical state space models for continuous sequence-to-sequence modeling](#). In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org, 2025.
- [15] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024.

- [16] David Córdova Bulens, Nathan F. Lepora, Stephen J. Redmond, and Benjamin Ward-Cherrier. Incipient slip detection with a biomimetic skin morphology. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8972–8978, 2023.
- [17] Arunkumar Byravan, Felix Leeb, Franziska Meier, and Dieter Fox. Se3-pose-nets: Structured deep dynamics models for visuomotor planning and control. *arXiv preprint arXiv:1710.00489*, 2017.
- [18] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. <https://github.com/huggingface/lerobot>, 2024.
- [19] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H. Adelson, and Sergey Levine. [More Than a Feeling: Learning to Grasp and Regrasp Using Vision and Touch](#). *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.
- [20] Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes? In *Conference on Robot Learning (CoRL)*, 2017.
- [21] Guanqun Cao, Jiaqi Jiang, Danushka Bollegala, and Shan Luo. Learn from incomplete tactile data: Tactile representation learning with masked autoencoders. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10800–10805. IEEE, 2023.
- [22] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [23] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.
- [24] Weihang Chen, Jing Xu, Fanbo Xiang, Xiaodi Yuan, Hao Su, and Rui Chen. General-purpose sim2real protocol for learning contact-rich manipulation with marker-based visuotactile sensors. *IEEE Transactions on Robotics*, 2024.
- [25] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for

- self-supervised representation learning. *International Journal of Computer Vision*, 132(1):208–223, 2024.
- [26] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.
 - [27] Yizhou Chen, Mark Van der Merwe, Andrea Sipos, and Nima Fazeli. Visuo-tactile transformers for manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 2026–2040. PMLR, 14–18 Dec 2023.
 - [28] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
 - [29] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
 - [30] Alex Church, John Lloyd, Raia Hadsell, and Nathan F Lepora. Deep reinforcement learning for tactile robotics: Learning to type on a braille keyboard. *IEEE Robotics and Automation Letters*, 5(4):6145–6152, 2020.
 - [31] Samuel Clarke, Negin Heravi, Mark Rau, Ruohan Gao, Jiajun Wu, Doug James, and Jeannette Bohg. Diffimpact: Differentiable rendering and identification of impact sounds. In *Conference on Robot Learning*, pages 662–673. PMLR, 2022.
 - [32] Samuel Clarke, Travers Rhodes, Christopher G. Atkeson, and Oliver Kroemer. Learning audio feedback for estimating amount and flow of granular material. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 529–550. PMLR, 29–31 Oct 2018.
 - [33] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016–2021.
 - [34] Nikhil Chavan Daffe, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S. Srinivasa, Michael Erdmann, Matthew T. Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrgge. Extrinsic dexterity: In-hand manipulation with external forces. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585, 2014.

- [35] Michael Danielczuk, Arsalan Mousavian, Clemens Eppner, and Dieter Fox. Object rearrangement using learned implicit collision functions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6010–6017. IEEE, 2021.
- [36] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024.
- [37] Neha Das, Sarah Bechtle, Todor Davchev, Dinesh Jayaraman, Akshara Rai, and Franziska Meier. Model-based inverse reinforcement learning from visual demonstration. In *Conference on Robot Learning (CoRL)*, 2020.
- [38] Vedant Dave, Fotios Lygerakis, and Elmar Rueckert. Multimodal visual-tactile representation learning through self-supervised contrastive pre-training, 2024.
- [39] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.
- [40] Siyuan Dong, Devesh K Jha, Diego Romeres, Sangwoon Kim, Daniel Nikovski, and Alberto Rodriguez. [Tactile-rl for insertion: Generalization to objects of unknown geometry](#). In *International Conference on Robotics and Automation (ICRA)*, 2021.
- [41] Siyuan Dong, Devesh K. Jha, Diego Romeres, Sangwoon Kim, Daniel Nikovski, and Alberto Rodriguez. Tactile-rl for insertion: Generalization to objects of unknown geometry. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6437–6443, 2021.
- [42] Siyuan Dong and Alberto Rodriguez. [Tactile-based insertion for dense box-packing](#). In *International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [43] Siyuan Dong, Wenzhen Yuan, and Edward H Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 137–144. IEEE, 2017.
- [44] Elliott Donlon, Siyuan Dong, Melody Liu, Jianhua Li, Edward Adelson, and Alberto Rodriguez. GelSlim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934, 2018.
- [45] Elliott Donlon, Siyuan Dong, Melody Liu, Jianhua Li, Edward Adelson, and Alberto Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934, 2018.

- [46] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [47] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [48] Yiming Dou, Fengyu Yang, Yi Liu, Antonio Loquercio, and Andrew Owens. Tactile-augmented radiance fields. *arXiv preprint arXiv:2405.04534*, 2024.
- [49] Yipai Du, Guanlan Zhang, Yazhan Zhang, and Michael Yu Wang. High-resolution 3-dimensional contact deformation tracking for fingervision sensor with dense random color pattern. *IEEE Robotics and Automation Letters*, 6(2):2147–2154, 2021.
- [50] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.
- [51] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *CoRL*, 12(16):23, 2017.
- [52] Ruoxuan Feng, Jiangyu Hu, Wenke Xia, Tianci Gao, Ao Shen, Yuhao Sun, Bin Fang, and Di Hu. Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. *arXiv preprint arXiv:2502.12191*, 2025.
- [53] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in Neural Information Processing Systems*, 29, 2016.
- [54] Jeremy A. Fishel and Gerald E. Loeb. [Sensing tactile microvibrations with the BioTac — Comparison with human sensitivity](#). In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1122–1127, 2012.
- [55] 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. In *Forty-first International Conference on Machine Learning*, 2024.

- [56] Letian Fu, Huang Huang, Lars Berscheid, Hui Li, Ken Goldberg, and Sachin Chitta. Safe self-supervised learning in real of visuo-tactile feedback policies for industrial insertion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10380–10386. IEEE, 2023.
- [57] Letian Fu, Huang Huang, Lars Berscheid, Hui Li, Ken Goldberg, and Sachin Chitta. [Safely Learning Visuo-Tactile Feedback Policies in Real For Industrial Insertion](#). In *International Conference on Robotics and Automation (ICRA)*, 2023.
- [58] Dhiraj Gandhi, Abhinav Gupta, and Lerrel Pinto. Swoosh! rattle! thump!—actions that sound. *arXiv preprint arXiv:2007.01851*, 2020.
- [59] Junli Gao, Zhaoji Huang, Zhaonian Tang, Haitao Song, and Wenyu Liang. [Visuo-Tactile-Based Slip Detection Using A Multi-Scale Temporal Convolution Network](#), 2023.
- [60] Ruihan Gao, Kangle Deng, Gengshan Yang, Wenzhen Yuan, and Jun-Yan Zhu. Tactile dreamfusion: Exploiting tactile sensing for 3d generation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.
- [61] 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. In *CoRL*, 2021.
- [62] 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.
- [63] 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.
- [64] 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*, pages 10598–10608, 2022.
- [65] Abraham George, Selam Gano, Pranav Katragadda, and Amir Barati Farimani. [VI-TaL Pretraining: Visuo-Tactile Pretraining for Tactile and Non-Tactile Manipulation Policies](#), 2024.
- [66] Oliver Gibbons, Alessandro Albini, and Perla Maiolino. A tactile feedback insertion strategy for peg-in-hole tasks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10415–10421. IEEE, 2023.

- [67] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. October 2019.
- [68] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.
- [69] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- [70] Albert Gu and Tri Dao. [Mamba: Linear-Time Sequence Modeling with Selective State Spaces](#). *arXiv preprint arXiv:2312.00752*, 2023.
- [71] Xiaofeng Guo, Hung-Jui Huang, and Wenzhen Yuan. Estimating properties of solid particles inside container using touch sensing. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8985–8992, 2023.
- [72] Harsh Gupta, Yuchen Mo, Shengmiao Jin, and Wenzhen Yuan. Sensor-invariant tactile representation. *arXiv preprint arXiv:2502.19638*, 2025.
- [73] Harsh Gupta, Yuchen Mo, Shengmiao Jin, and Wenzhen Yuan. Sensor-invariant tactile representation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [74] Irmak Guzey, Ben Evans, Soumith Chintala, and Lerrel Pinto. Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play. In *7th Annual Conference on Robot Learning*, 2023.
- [75] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.
- [76] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.
- [77] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. [Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system](#). In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.
- [78] Johanna Hansen, Francois Hogan, Dmitriy Rivkin, David Meger, Michael Jenkin, and Gregory Dudek. Visuotactile-rl: Learning multimodal manipulation policies with deep reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8298–8304. IEEE, 2022.

- [79] Nicklas Hansen, Zhecheng Yuan, Yanjie Ze, Tongzhou Mu, Aravind Rajeswaran, Hao Su, Huazhe Xu, and Xiaolong Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. *arXiv preprint arXiv:2212.05749*, 2022.
- [80] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [81] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [83] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019.
- [84] Carolina Higuera, Siyuan Dong, Byron Boots, and Mustafa Mukadam. Neural contact fields: Tracking extrinsic contact with tactile sensing. In *International Conference on Robotics and Automation (ICRA)*, 2023.
- [85] Carolina Higuera, Joseph Ortiz, Haozhi Qi, Luis Pineda, Byron Boots, and Mustafa Mukadam. Perceiving extrinsic contacts from touch improves learning insertion policies, 2023.
- [86] Carolina Higuera, Joseph Ortiz, Haozhi Qi, Luis Pineda, Byron Boots, and Mustafa Mukadam. Perceiving extrinsic contacts from touch improves learning insertion policies. *arXiv preprint arXiv:2309.16652*, 2023.
- [87] 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](#). 2024.
- [88] Carolina Higuera, Akash Sharma, Taosha Fan, Chaithanya Krishna Bodduluri, Byron Boots, Michael Kaess, Mike Lambeta, Tingfan Wu, Zixi Liu, Francois Robert Hogan, and Mustafa Mukadam. Tactile beyond pixels: Multisensory touch representations for robot manipulation. In *9th Annual Conference on Robot Learning*, 2025.
- [89] Francois R Hogan, Jean-François Tremblay, Bobak H Baghi, Michael Jenkin, Kaleem Siddiqi, and Gregory Dudek. Finger-STs: Combined proximity and tactile sensing

- for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(4):10865–10872, 2022.
- [90] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.
 - [91] Edward S Hu, Kwangjun Ahn, Qinghua Liu, Haoran Xu, Manan Tomar, Ada Langford, Dinesh Jayaraman, Alex Lamb, and John Langford. Learning to achieve goals with belief state transformers. *arXiv preprint arXiv:2410.23506*, 2024.
 - [92] Hung-Jui Huang, Xiaofeng Guo, and Wenzhen Yuan. Understanding Dynamic Tactile Sensing for Liquid Property Estimation. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.
 - [93] Hung-Jui Huang, Michael Kaess, and Wenzhen Yuan. [NormalFlow: Fast, Robust, and Accurate Contact-based Object 6DoF Pose Tracking with Vision-based Tactile Sensors](#). *IEEE Robotics and Automation Letters*, 2024.
 - [94] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.
 - [95] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. *arXiv preprint arXiv:2006.04902*, 2020.
 - [96] Shubham Kanitkar, Helen Jiang, and Wenzhen Yuan. PoseIt: a visual-tactile dataset of holding poses for grasp stability analysis. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 71–78. IEEE, 2022.
 - [97] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. 2024.
 - [98] Haresh Karnan, Elvin Yang, Daniel Farkash, Garrett Warnell, Joydeep Biswas, and Peter Stone. Sterling: Self-supervised terrain representation learning from unconstrained robot experience. *arXiv preprint arXiv:2309.15302*, 2023.
 - [99] 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. In *Robotics: Science and Systems*, 2023.

- [100] Sangwoon Kim, Antonia Bronars, Parag Patre, and Alberto Rodriguez. Texterity–tactile extrinsic dexterity: Simultaneous tactile estimation and control for extrinsic dexterity. *arXiv preprint arXiv:2403.00049*, 2024.
- [101] Sangwoon Kim and Alberto Rodriguez. [Active Extrinsic Contact Sensing: Application to General Peg-in-Hole Insertion](#). In *International Conference on Robotics and Automation (ICRA)*, 2022.
- [102] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.
- [103] Diederik P Kingma and Max Welling. [Auto-encoding variational bayes](#). In *International Conference on Learning Representations (ICLR)*, 2014.
- [104] Raj Kolamuri, Zilin Si, Yufan Zhang, Arpit Agarwal, and Wenzhen Yuan. Improving grasp stability with rotation measurement from tactile sensing. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6809–6816. IEEE, 2021.
- [105] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.
- [106] 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](#). *Robotics and Automation Letters (RA-L)*, 2020.
- [107] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. 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.
- [108] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. 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.
- [109] Mike Lambeta, Tingfan Wu, Ali Sengul, Victoria Rose Most, Nolan Black, Kevin Sawyer, Romeo Mercado, Haozhi Qi, Alexander Sohn, Byron Taylor, Norb Tydingco, Gregg Kammerer, Dave Stroud, Jake Khatha, Kurt Jenkins, Kyle Most, Neal Stein, Ricardo Chavira, Thomas Craven-Bartle, Eric Sanchez, Yitian Ding, Jitendra Malik, and Roberto Calandra. Digitizing touch with an artificial multimodal fingertip, 2024.

- [110] Mike Lambeta, Tingfan Wu, Ali Sengul, Victoria Rose Most, Nolan Black, Kevin Sawyer, Romeo Mercado, Haozhi Qi, Alexander Sohn, Byron Taylor, Norb Tydingco, Gregg Kammerer, Dave Stroud, Jake Khatha, Kurt Jenkins, Kyle Most, Neal Stein, Ricardo Chavira, Thomas Craven-Bartle, Eric Sanchez, Yitian Ding, Jitendra Malik, and Roberto Calandra. Digitizing touch with an artificial multimodal fingertip, 2024.
- [111] Mike Lambeta, Huazhe Xu, Jingwei Xu, Po-Wei Chou, Shaoxiong Wang, Trevor Darrell, and Roberto Calandra. PyTouch: A machine learning library for touch processing. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13208–13214. IEEE, 2021.
- [112] Patrick Lancaster, Nicklas Hansen, Aravind Rajeswaran, and Vikash Kumar. Modem-v2: Visuo-motor world models for real-world robot manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7530–7537, 2024.
- [113] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1), 2022.
- [114] Nathan F Lepora. Soft biomimetic optical tactile sensing with the tactip: A review. *IEEE Sensors Journal*, 21(19):21131–21143, 2021.
- [115] Nathan F Lepora, Yijiong Lin, Ben Money-Coomes, and John Lloyd. Digitac: A digit-tactip hybrid tactile sensor for comparing low-cost high-resolution robot touch. *IEEE Robotics and Automation Letters*, 7(4):9382–9388, 2022.
- [116] Hao Li, Yizhi Zhang, Junzhe Zhu, Shaoxiong Wang, Michelle A Lee, Huazhe Xu, Edward Adelson, Li Fei-Fei, Ruohan Gao, and Jiajun Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1368–1378. PMLR, 14–18 Dec 2023.
- [117] Hongyu Li, Snehal Dikhale, Jinda Cui, Soshi Iba, and Nawid Jamali. [HyperTaxel: Hyper-Resolution for Taxel-Based Tactile Signals Through Contrastive Learning](#). In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7499–7506, 2024.
- [118] Boyuan Liang, Wenyu Liang, and Yan Wu. [Tactile-Guided Dynamic Object Planar Manipulation](#). In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3203–3209, 2022.
- [119] Hongzhuo Liang, Shuang Li, Xiaojian Ma, Norman Hendrich, Timo Gerkmann, Fuchun Sun, and Jianwei Zhang. Making sense of audio vibration for liquid height estimation in robotic pouring. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5333–5339, 2019.

- [120] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.
- [121] Yijiong Lin, John Lloyd, Alex Church, and Nathan F Lepora. Tactile gym 2.0: Sim-to-real deep reinforcement learning for comparing low-cost high-resolution robot touch. *IEEE Robotics and Automation Letters*, 7(4):10754–10761, 2022.
- [122] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020.
- [123] Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppaswamy, Benjamin Burchfiel, and Shuran Song. Maniway: Learning robot manipulation from in-the-wild audio-visual data. In *8th Annual Conference on Robot Learning*, 2024.
- [124] John Lloyd and Nathan F. Lepora. Goal-driven robotic pushing using tactile and proprioceptive feedback. *IEEE Transactions on Robotics*, 38(2):1201–1212, 2022.
- [125] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.
- [126] Zhenyu Lu, Zihan Liu, Xingyu Zhang, Yan Liang, Yuming Dong, and Tianyu Yang. 3d force identification and prediction using deep learning based on a gelsight-structured sensor. *Sensors and Actuators A: Physical*, 367:115036, 2024.
- [127] Daolin Ma, Siyuan Dong, and Alberto Rodriguez. [Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements](#). In *International Conference on Robotics and Automation (ICRA)*, 2021.
- [128] Daolin Ma, Elliott Donlon, Siyuan Dong, and Alberto Rodriguez. Dense tactile force estimation using gelSlim and inverse FEM. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5418–5424. IEEE, 2019.
- [129] Lucas Manuelli, Yunzhu Li, Pete Florence, and Russ Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. *arXiv preprint arXiv:2009.05085*, 2020.
- [130] Carolyn Matl, Yashraj Narang, Ruzena Bajcsy, Fabio Ramos, and Dieter Fox. Inferring the material properties of granular media for robotic tasks. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2770–2777, 2020.
- [131] Leland McInnes, John Healy, and James Melville. [Umap: Uniform manifold approximation and projection for dimension reduction](#). *arXiv preprint arXiv:1802.03426*, 2018.

- [132] Jared Mejia, Victoria Dean, Tess Hellebrekers, and Abhinav Gupta. Hearing touch: Audio-visual pretraining for contact-rich manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6912–6919. IEEE, 2024.
- [133] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [134] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12486, 2021.
- [135] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on robot learning*, pages 1101–1112. PMLR, 2020.
- [136] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14200–14213. Curran Associates, Inc., 2021.
- [137] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.
- [138] Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, et al. [Factory: Fast contact for robotic assembly](#). In *Robotics: Science and Systems (RSS)*, 2022.
- [139] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Byol for audio: Exploring pre-trained general-purpose audio representations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:137–151, 2023.
- [140] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [141] Kei Ota, Devesh K Jha, Krishna Murthy Jatavallabhula, Asako Kanezaki, and Joshua B Tenenbaum. Tactile estimation of extrinsic contact patch for stable placement. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13876–13882. IEEE, 2024.

- [142] Utku Ozbulak, Hyun Jung Lee, Beril Boga, Esla Timothy Anzaku, Ho min Park, Arnout Van Messem, Wesley De Neve, and Joris Vankerschaver. Know your self-supervised learning: A survey on image-based generative and discriminative training. *Transactions on Machine Learning Research*, 2023. Survey Certification.
- [143] Daniel Palenicek, Theo Gruner, Tim Schneider, Alina Böhm, Janis Lenz, Inga Pfennig, Eric Krämer, and Jan Peters. Learning tactile insertion in the real world. *arXiv preprint arXiv:2405.00383*, 2024.
- [144] Venkatesh Pattabiraman, Yifeng Cao, Siddhant Haldar, Lerrel Pinto, and Raunaq Bhirangi. [Learning Precise, Contact-Rich Manipulation through Uncalibrated Tactile Skins](#). *arXiv preprint arXiv:2410.17246*, 2024.
- [145] PaXini. [ITPU multi-dimensional tactile sensing unit](#), 2025.
- [146] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. , title= Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Computer Vision and Oattern Recognition (CVPR)*, 2017.
- [147] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi~Ma, and Jitendra Malik. In-Hand Object Rotation via Rapid Motor Adaptation. In *Conference on Robot Learning (CoRL)*, 2022.
- [148] Haozhi Qi, Brent Yi, Mike Lambeta, Yi~Ma, Roberto Calandra, and Jitendra Malik. [From Simple to Complex Skills: The Case of In-Hand Object Reorientation](#). *arXiv preprint arXiv:2501.05439*, 2025.
- [149] 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 (CoRL)*, 2023.
- [150] Alec Radford, Jong~Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et~al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [151] Alec Radford, Jong~Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et~al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [152] Akshat Rakheja, Aarsh Ashdhir, Aryan Bhattacharjee, and Vanshika Sharma. World consistency score: A unified metric for video generation quality. *arXiv preprint arXiv:2508.00144*, 2025.

- [153] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.
- [154] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.
- [155] Branden Romero, Hao-Shu Fang, Pulkit Agrawal, and Edward Adelson. [EyeSight Hand: Design of a Fully-Actuated Dexterous Robot Hand with Integrated Vision-Based Tactile Sensors and Compliant Actuation](#). In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1853–1860, 2024.
- [156] Reuven~Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.
- [157] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.
- [158] Gerrit Schoettler, Ashvin Nair, Jianlan Luo, Shikhar Bahl, Juan~Aparicio Ojea, Eugen Solowjow, and Sergey Levine. [Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards](#). In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [159] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. [Proximal policy optimization algorithms](#). *arXiv:1707.06347*, 2017.
- [160] Nur Muhammad~Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.
- [161] Akash Sharma, Carolina Higuera, Chaithanya~Krishna Bodduluri, Zixi Liu, Taosha Fan, Tess Hellebrekers, Mike Lambeta, Byron Boots, Michael Kaess, Tingfan Wu, Francois~Robert Hogan, and Mustafa Mukadam. Self-supervised perception for tactile skin covered dexterous hands, 2025.
- [162] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.
- [163] Zilin Si, Zirui Zhu, Arpit Agarwal, Stuart Anderson, and Wenzhen Yuan. Grasp stability prediction with sim-to-real transfer from tactile sensing. In *2022 IEEE/RSJ*

- International Conference on Intelligent Robots and Systems (IROS)*, pages 7809–7816, 2022.
- [164] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua~B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. [Neural descriptor fields: Se \(3\)-equivariant object representations for manipulation](#). In *International Conference on Robotics and Automation (ICRA)*, 2022.
 - [165] Andrea Sipos and Nima Fazeli. [Simultaneous contact location and object pose estimation using proprioception and tactile feedback](#). In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.
 - [166] Jianlin Su, Yu~Lu, Shengfeng Pan, Ahmed Murtadha, Bo~Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.
 - [167] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.
 - [168] 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.
 - [169] 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](#). *Science Robotics*, page adl0628, 2024.
 - [170] Sudharshan Suresh, Zilin Si, Stuart Anderson, Michael Kaess, and Mustafa Mukadam. MidasTouch: Monte-Carlo inference over distributions across sliding touch. In *Conference on Robot Learning (CoRL)*, 2022.
 - [171] Sudharshan Suresh, Zilin Si, Stuart Anderson, Michael Kaess, and Mustafa Mukadam. Midastouch: Monte-carlo inference over distributions across sliding touch. In *Conference on Robot Learning*, pages 319–331. PMLR, 2023.
 - [172] Giovanni Sutanto, Nathan Ratliff, Balakumar Sundaralingam, Yevgen Chebotar, Zhe Su, Ankur Handa, and Dieter Fox. Learning latent space dynamics for tactile servoing. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3622–3628. IEEE, 2019.
 - [173] Bingjie Tang, Michael~A Lin, Iretoiayo Akinola, Ankur Handa, Gaurav~S Sukhatme, Fabio Ramos, Dieter Fox, and Yashraj Narang. Industreal: Transferring contact-rich assembly tasks from simulation to reality. *arXiv preprint arXiv:2305.17110*, 2023.

- [174] Ian~H. Taylor, Siyuan Dong, and Alberto Rodriguez. GelSlim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10781–10787, 2022.
- [175] Abitha Thankaraj and Lerrel Pinto. That sounds right: Auditory self-supervision for dynamic robot manipulation. In *Conference on Robot Learning*, pages 1036–1049. PMLR, 2023.
- [176] Stephen Tian, Frederik Ebert, Dinesh Jayaraman, Mayur Mudigonda, Chelsea Finn, Roberto Calandra, and Sergey Levine. Manipulation by feel: Touch-based control with deep predictive models. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 818–824. IEEE, 2019.
- [177] Manan Tomar, Philippe Hansen-Estruch, Philip Bachman, Alex Lamb, John Langford, Matthew~E Taylor, and Sergey Levine. Video occupancy models. *arXiv preprint arXiv:2407.09533*, 2024.
- [178] Tito~Pradhono Tomo, Massimo Regoli, Alexander Schmitz, Lorenzo Natale, Harris Kristanto, Sophon Somlor, Lorenzo Jamone, Giorgio Metta, and Shigeki Sugano. A new silicone structure for uskin—a soft, distributed, digital 3-axis skin sensor and its integration on the humanoid robot icub. *IEEE Robotics and Automation Letters*, 3(3):2584–2591, 2018.
- [179] Tito~Pradhono Tomo, Alexander Schmitz, Wai~Keat Wong, Harris Kristanto, Sophon Somlor, Jinsun Hwang, Lorenzo Jamone, and Shigeki Sugano. [Covering a Robot Fingertip With uSkin: A Soft Electronic Skin With Distributed 3-Axis Force Sensitive Elements for Robot Hands](#). *IEEE Robotics and Automation Letters*, 3(1):124–131, 2018.
- [180] Tito~Pradhono Tomo, Wai~Keat Wong, Alexander Schmitz, Harris Kristanto, Alexandre Sarazin, Lorenzo Jamone, Sophon Somlor, and Shigeki Sugano. [A modular, distributed, soft, 3-axis sensor system for robot hands](#). In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 454–460. IEEE, 2016.
- [181] Aaron Van Den~Oord, Oriol Vinyals, et~al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.
- [182] Mark Van~der Merwe, Youngsun Wi, Dmitry Berenson, and Nima Fazeli. [Integrated Object Deformation and Contact Patch Estimation from Visuo-Tactile Feedback](#). *arXiv:2305.14470*, 2023.
- [183] Filipe Veiga, Herke Van~Hoof, Jan Peters, and Tucker Hermans. [Stabilizing novel objects by learning to predict tactile slip](#). In *International Conference on Intelligent Robots and Systems (IROS)*, 2015.

- [184] 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](#). *Robotics and Automation Letters (RA-L)*, 2022.
- [185] 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.
- [186] Shaoxiong Wang, Yu~She, Branden Romero, and Edward~H Adelson. [GelSight Wedge: Measuring High-Resolution 3D Contact Geometry with a Compact Robot Finger](#). In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.
- [187] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in Neural Information Processing Systems*, 28, 2015.
- [188] Junhang Wei, Shaowei Cui, Jingyi Hu, Peng Hao, Shuo Wang, and Zheng Lou. [Multi-modal Unknown Surface Material Classification and Its Application to Physical Reasoning](#). *IEEE Transactions on Industrial Informatics*, 18(7):4406–4416, 2022.
- [189] Youngsun Wi, Pete Florence, Andy Zeng, and Nima Fazeli. [Virdo: Visio-tactile implicit representations of deformable objects](#). In *International Conference on Robotics and Automation (ICRA)*, 2022.
- [190] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.
- [191] Tianhao Wu, Jinzhou Li, Jiyao Zhang, Mingdong Wu, and Hao Dong. [Canonical Representation and Force-Based Pretraining of 3D Tactile for Dexterous Visuo-Tactile Policy Learning](#), 2024.
- [192] Yansong Wu, Zongxie Chen, Fan Wu, Lingyun Chen, Liding Zhang, Zhenshan Bing, Abdalla Swikir, Sami Haddadin, and Alois Knoll. Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation. *arXiv preprint arXiv:2409.11047*, 2024.
- [193] Huazhe Xu, Yuping Luo, Shaoxiong Wang, Trevor Darrell, and Roberto Calandra. Towards learning to play piano with dexterous hands and touch. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10410–10416. IEEE, 2022.
- [194] Zhengtong Xu, Raghava Uppuluri, Xinwei Zhang, Cael Fitch, Philip~Glen Crandall, Wan Shou, Dongyi Wang, and Yu~She. Unit: Unified tactile representation for robot learning. *arXiv preprint arXiv:2408.06481*, 2024.

- [195] Zhengtong Xu, Raghava Uppuluri, Xinwei Zhang, Cael Fitch, Philip~Glen Crandall, Wan Shou, Dongyi Wang, and Yu~She. UniT: Data efficient tactile representation with generalization to unseen objects, 2025.
- [196] Han Xue, Jieji Ren, Wendi Chen, Gu~Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. *arXiv preprint arXiv:2503.02881*, 2025.
- [197] Fengyu Yang, Chao Feng, Ziyang Chen, Hyounseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, et~al. Binding touch to everything: Learning unified multimodal tactile representations. *arXiv preprint arXiv:2401.18084*, 2024.
- [198] 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.
- [199] Max Yang, Chenghua Lu, Alex Church, Yijiong Lin, Chris Ford, Haoran Li, Efi Psomopoulou, David~AW Barton, and Nathan~F Lepora. Anyrotate: Gravity-invariant in-hand object rotation with sim-to-real touch. *arXiv preprint arXiv:2405.07391*, 2024.
- [200] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024.
- [201] Lin Yen-Chen, Maria Bauza, and Phillip Isola. Experience-embedded visual foresight. In *Conference on Robot Learning*, pages 1015–1024. PMLR, 2020.
- [202] Jessica Yin, Haozhi Qi, Jitendra Malik, James Pikul, Mark Yim, and Tess Hellebrekers. [Learning in-hand translation using tactile skin with shear and normal force sensing](#). *arXiv preprint arXiv:2407.07885*, 2024.
- [203] 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.
- [204] Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 1992–2005. PMLR, 16–18 Nov 2021.
- [205] Kelin Yu, Yunhai Han, Qixian Wang, Vaibhav Saxena, Danfei Xu, and Ye~Zhao. [MimicTouch: Leveraging Multi-modal Human Tactile Demonstrations for Contact-rich Manipulation](#). In *8th Annual Conference on Robot Learning*, 2024.

- [206] 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.
- [207] Wenzhen Yuan, Siyuan Dong, and Edward Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors: Special Issue on Tactile Sensors and Sensing*, 17(12):2762 – 2782, November 2017.
- [208] Wenzhen Yuan, Siyuan Dong, and Edward~H. Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017.
- [209] Wenzhen Yuan, Rui Li, Mandayam~A. Srinivasan, and Edward~H. Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 304–311, 2015.
- [210] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward~H. Adelson. Active clothing material perception using tactile sensing and deep learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, 2017.
- [211] 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, 2018.
- [212] Ben Zandonati, Ruohan Wang, Ruihan Gao, and Yan Wu. Investigating vision foundational models for tactile representation learning. *arXiv preprint arXiv:2305.00596*, 2023.
- [213] Andrea Zangrandi, Marco D’Alonzo, Christian Cipriani, and Giovanni Di~Pino. Neurophysiology of slip sensation and grip reaction: insights for hand prosthesis control of slippage. *Journal of neurophysiology*, 126(2):477–492, 2021.
- [214] Fan Zhang and Yiannis Demiris. Visual-tactile learning of garment unfolding for robot-assisted dressing. *IEEE Robotics and Automation Letters*, 8(9):5512–5519, 2023.
- [215] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, October 2023.
- [216] Xiang Zhang, Changhao Wang, Lingfeng Sun, Zheng Wu, Xinghao Zhu, and Masayoshi Tomizuka. Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning. In *Conference on Robot Learning*, pages 1621–1639. PMLR, 2023.

- [217] Jialiang Zhao, Yuxiang Ma, Lirui Wang, and Edward~H Adelson. Transferable tactile transformers for representation learning across diverse sensors and tasks. *arXiv preprint arXiv:2406.13640*, 2024.
- [218] Tony~Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
- [219] Tony~Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.
- [220] Tony~Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
- [221] Wenxuan Zhou and David Held. Learning to grasp the ungraspable with emergent extrinsic dexterity. In *ICRA 2022 Workshop: Reinforcement Learning for Contact-Rich Manipulation*, 2022.

Appendix A

SPARSH SUPPLEMENTARY MATERIAL

A.1 TacBench evaluations via fine-tuning

Fine-tuning the **Sparsh** encoders is another method of assessing the quality of pre-trained representations. Fine-tuning can potentially enhance performance in downstream tasks when the pre-trained model lacks task-relevant information.

We evaluated both the full and partial fine-tuning of **Sparsh** on TacBench. In full fine-tuning, all encoder parameters are updated through task supervision. In partial fine-tuning, we update only the last transformer block of the encoder. Figure A.1 shows the fine-tuning results in the benchmark with varying amounts of labeled data. Notably, models pre-trained in latent space (DINO, I-JEPA, V-JEPA) perform better in downstream tasks when fully fine-tuned, especially in regression tasks like force and pose estimation. For example, Figure A.1(a) illustrates that errors in force estimation are significantly lower with full fine-tuning, even with only 33% and 10% labeled data. Full fine-tuning also enhances performance in classification tasks such as slip detection, grasp stability, and textile classification, as shown in Fig. A.1(b,c). Adding in-domain data to the encoder reduces performance gaps in the benchmark between **Sparsh** (DINO), **Sparsh** (IJEPA), and **Sparsh** (VJEPA). However, this method is less effective for the **Sparsh** (MAE) model, which is trained in pixel space. We hypothesize that MAE weights are potentially more brittle when compared to other SSL models which enjoy a wider basin of minima due to weight updates via exponential moving average. In contrast, partial fine-tuning offers minor improvements, aligning closely with the performance of frozen models.

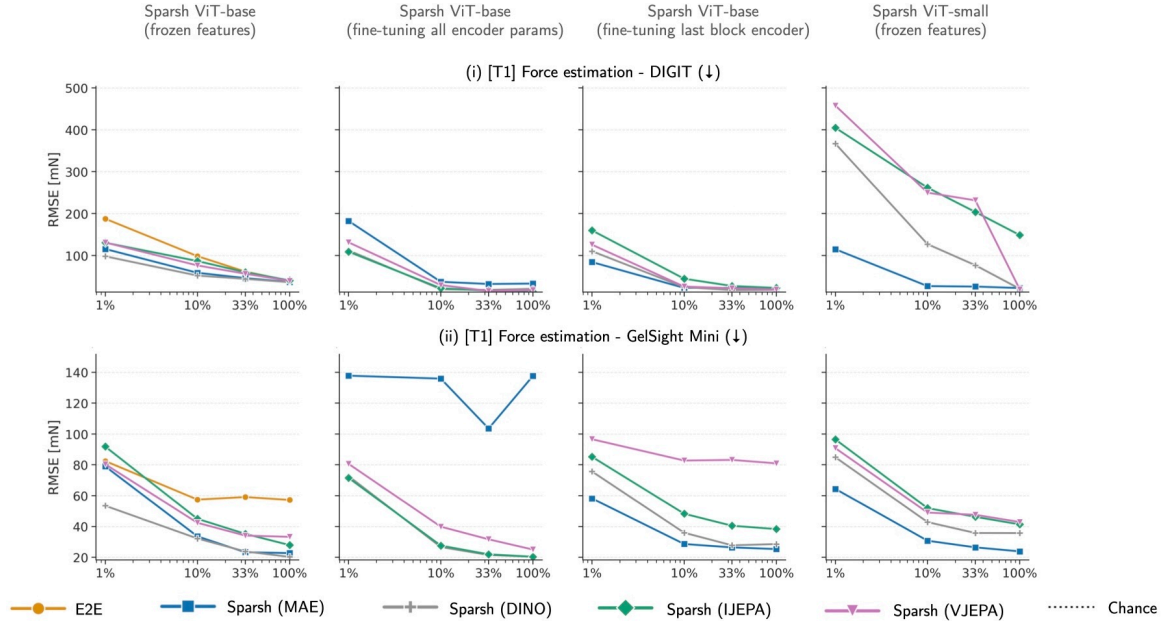
A.1.1 Sparsh ViT-small and performance

We train **Sparsh** for all SSL approaches decreasing the model capacity by using a transformer ViT-small. This let us study the effect of the dimensionality of the touch representations on downstream tasks, from 768 with ViTbase to 384 with ViTsmall.

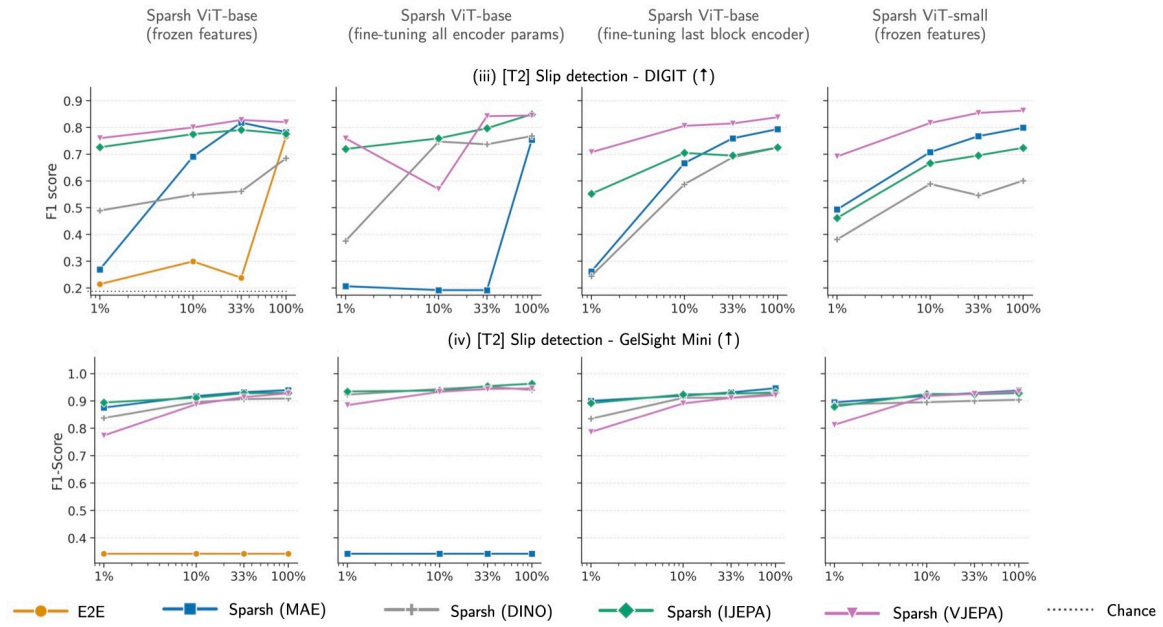
We evaluate **Sparsh**-vitsmall across TacBench. In Fig. A.1 we report the performance of each task for different budgets of labeled data following the attentive probing protocol. Reducing the dimensionality of the representations do plays an important role for some tasks. Regression-like tasks such as [T1] Force estimation (see Fig. A.1a) exhibit a decrease in performance when reducing the capacity of the encoders, specially when the downstream tasks needs to be trained under a limited number of labeled data.

For instance, **Sparsh** (DINO) increases the force estimation error by 74% for DIGIT and 50.3% for GelSight Mini when using representations from **Sparsh-vitsmall** and training the downstream tasks with 33% of labeled data. The decrease in performance is also observe in regression-by-classification tasks, as in [T3] Pose estimation. With **Sparsh-vitsmall** all models perform very similar but losing 20% accuracy even when training the downstream tasks with the full labeled dataset. Nevertheless the performance is still better than an E2E model with a vitbase encoder.

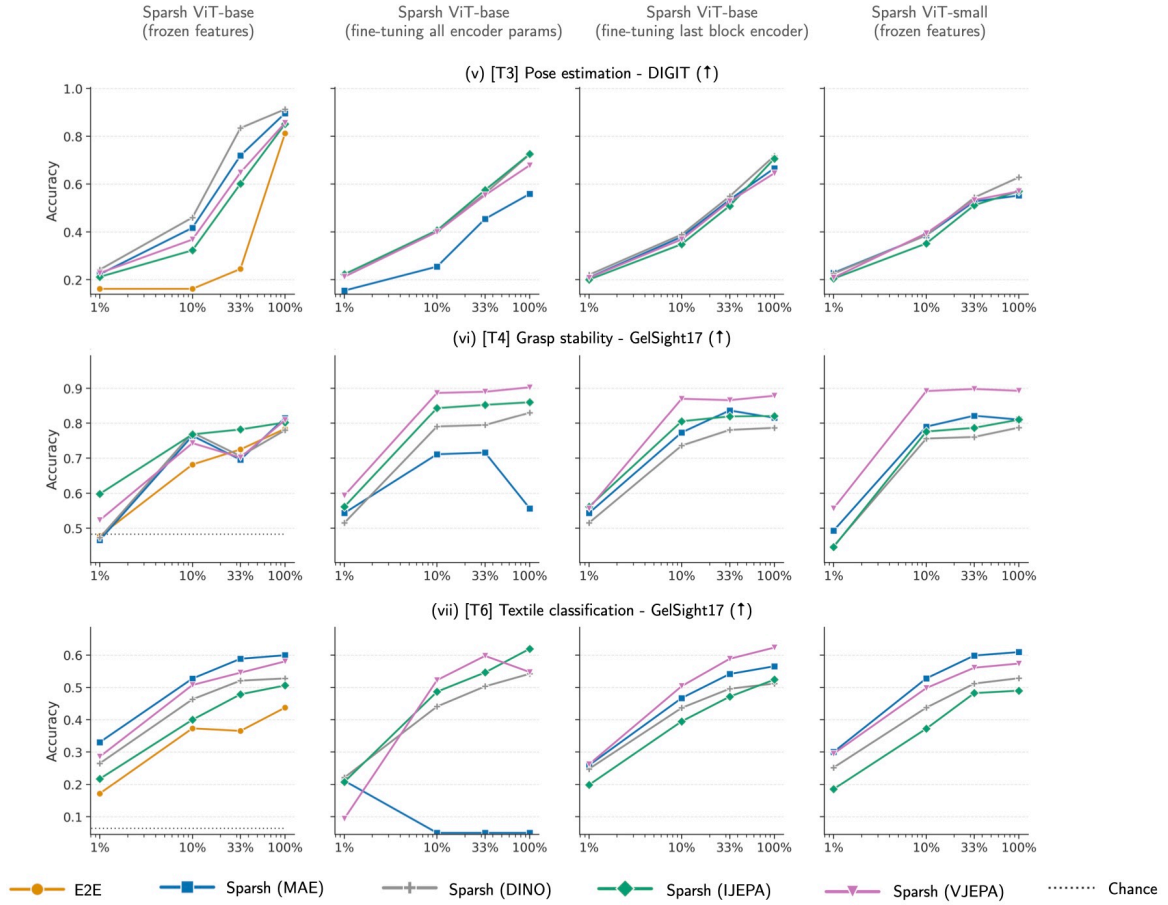
In general for classification tasks in the benchmark like [T2] Slip detection, [T4] Grasp stability and [T5] Textile recognition, there is no major effect of reducing the capacity of the encoder. The drop in performance is only significant when training the downstream task with the lowest amount of training data, 1% in our experiments.



(a) Additional evaluations for [T1] force estimation.



(b) Additional evaluations for [T2] slip detection.



(c) Additional evaluations for the perception tasks, [T3] pose estimation (top), [T4] grasp stability (middle) and [T6] textile classification (bottom).

Figure A.1: Additional evaluations of Sparsh representations on TacBench. We compare frozen Sparsh ViT-base (most left), Sparsh fully and partially fine-tuned (middle) and finally (most right) Sparsh ViT-small to gauge the effect of reducing the dimensionality of the representations.

Appendix B

SPARSH-SKIN SUPPLEMENTARY MATERIAL

B.1 Force Estimation

In [Fig. B.1](#), we present the correlation metrics between ground truth and predicted forces on test data for decoders trained with a 33% data budget. The results show that end-to-end training leads to overfitting, resulting in poor generalization to unseen strokes and essentially random normal force predictions. In contrast, using **Sparsh-skin** (frozen) representations yields better fitting, which can be further improved by adapting these representations to in-domain data.

In [Fig. B.2](#), we present a comparison between ground-truth testing strokes (normal loading sequences) and their reconstructed counterparts, obtained by passing Xela data through the frozen force decoder to recreate the sequences. The forces estimated via **Sparsh-skin** (frozen) are able to capture increasing/decreasing changes in the normal loading, as opposed to the end-to-end model. Shear from skin representations is not as accurate as normal force prediction, but the trend of the tangential forces matches the ground truth.

B.2 Joystick State Estimation

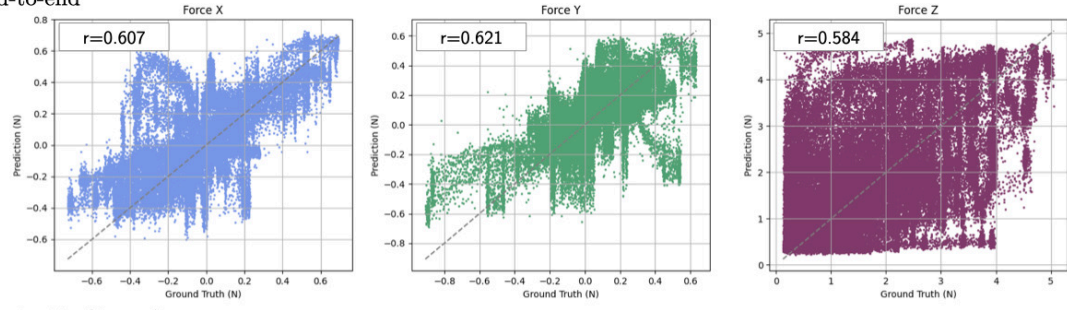
For this task, we highlight that when we train decoders using pretrained representations as the input, the convergence rate of the validation RMSE is significantly higher ([Fig. B.3](#)) than training the decoder using raw observations through uninitialized models. Specifically observe that **Sparsh-skin** (fine-tuned) is able to reach performance on par with end-to-end pretrained model within 12.9k optimization steps.

B.3 Pose Estimation

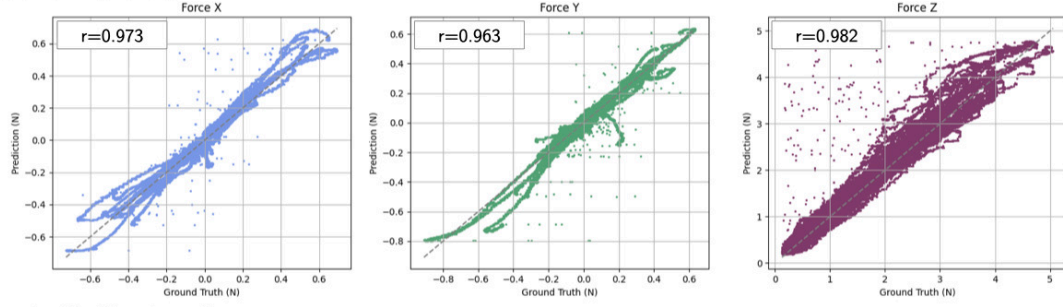
In this task, we aim to predict the object pose over 1-second trajectories. Xela observations at 100Hz are converted into tactile representation tokens at the output frequency using **Sparsh-skin** in a cascaded manner. Following attentive pooling, a single-layer transformer block is applied to reason about the 1-second context window of full-hand tactile features.

[Fig. B.4](#) compares ground-truth test pose sequences with their reconstructed counterparts, obtained from task models trained on 100% and 33% of the available data. The

End-to-end



Sparsh-skin (frozen)



Sparsh-skin (fine-tuned)

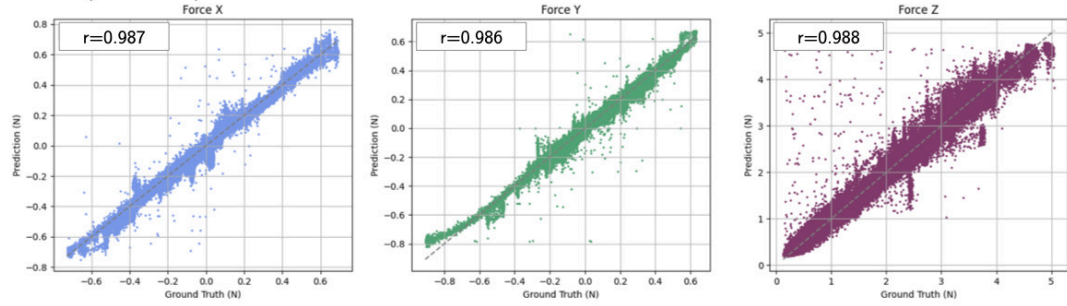


Figure B.1: Correlation between ground truth and predicted forces on unseen normal loading with a hemispherical indenter on Xela sensors.

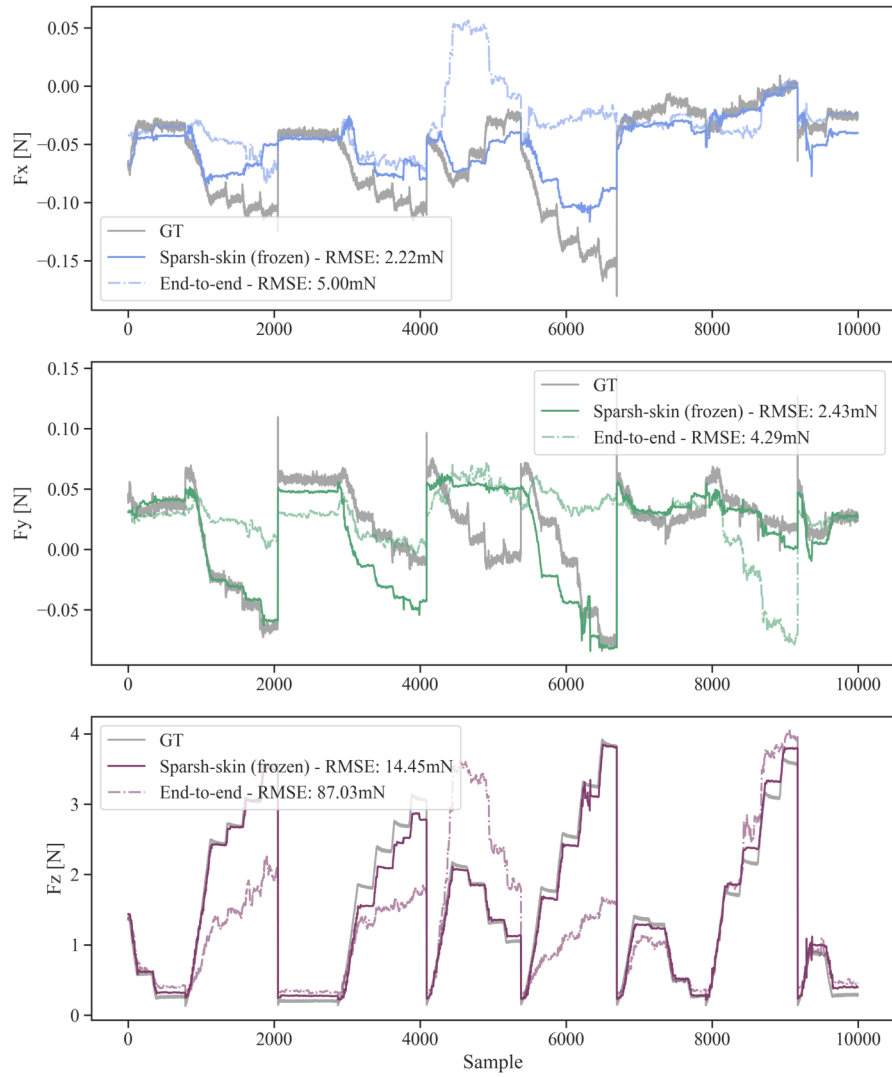


Figure B.2: Ground truth tangential and normal force from test strokes with flat indenter (gray) and force sequence reconstruction from Sparsh-skin (frozen) and end-to-end model.

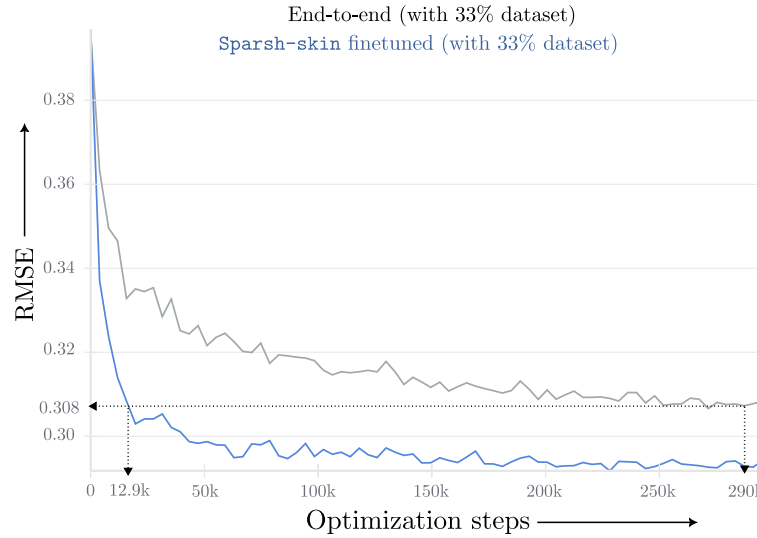


Figure B.3: Validation RMSE convergence rates between **Sparsh-skin** fine-tuned and end-to-end. We find that **Sparsh-skin** allows the model to generalize and learn the patterns required to infer joystick states significantly faster during training.

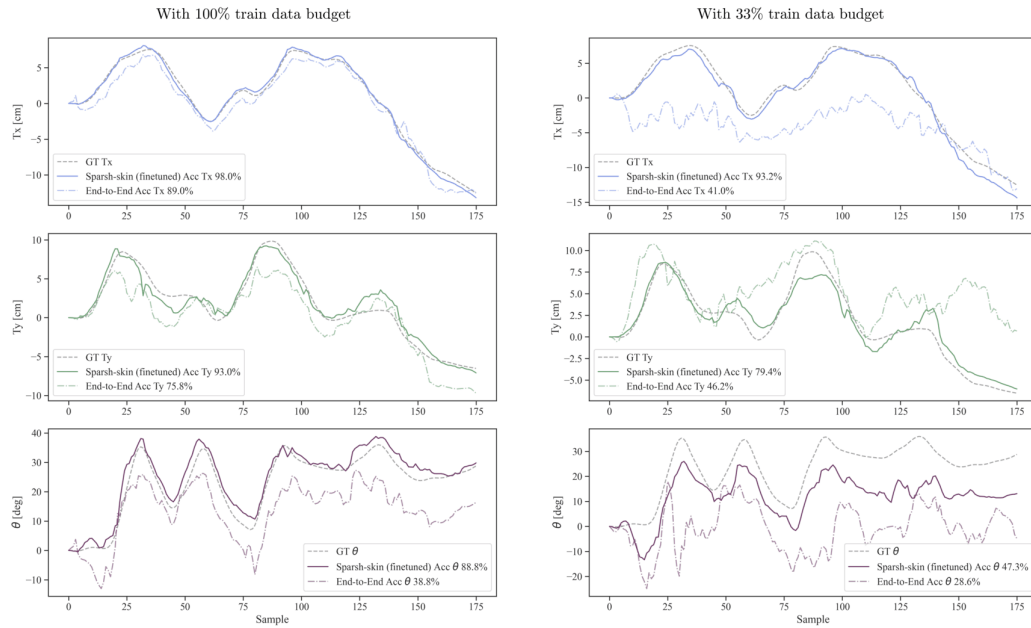


Figure B.4: Ground truth pose sequence for object in test set and reconstructed trajectory via end-to-end and **Sparsh-skin** (finetuned) representations. (left) Task decoders trained with 100% of train data budget, corresponding to 108 sequences. (right) Task decoders trained with 33% of train sequences.

results show that fine-tuning **Sparsh-skin** on the full dataset yields higher accuracy in estimating object pose changes over time compared to traditional end-to-end approaches. Moreover, even with a drastic reduction in labeled samples (to 33%), the model still achieves relatively good performance, particularly in tracking translation changes. Furthermore, for this task, we also visualize that this task requires *full-hand* sensing. For instance, in Fig. B.5, we observe that when we use **Sparsh-skin** by removing palm sensing on the Xela hand results in $\geq 10\%$ drop in pose tracking performance.

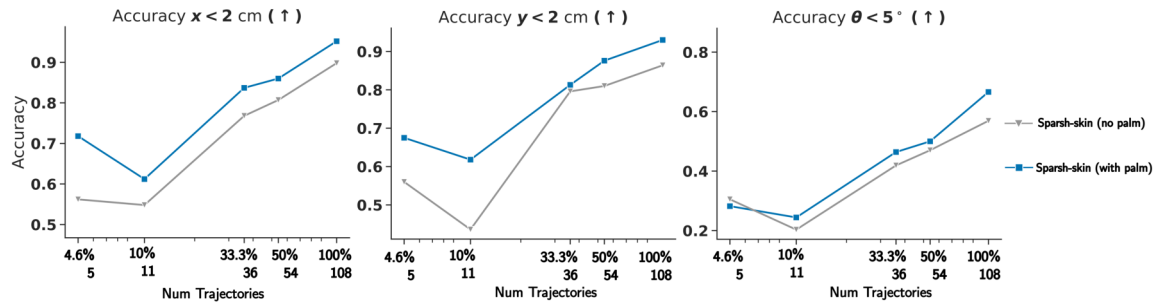


Figure B.5: Comparison of pose estimation accuracy of **Sparsh-skin** with and without palm sensing.

Appendix C

SPARSH-X SUPPLEMENTARY MATERIAL

C.1 Dataset for SSL Pretraining

Our self-supervised learning (SSL) dataset is sourced from two platforms: an Allegro hand equipped with Digit 360 sensors mounted on each fingertip, and a custom mobile picker tool with sensors integrated into its gripping mechanism. Since SSL representation learning does not require labeled data, we collect tactile data by having the Allegro hand interact rummaging freely with a tray filled with LEGO blocks and marbles. This setup enables the capture of rich, multi-contact interactions with objects that feature distinctive geometries, such as spherical shapes and sharp edges.

The mobile picker is used to gather tactile data from everyday manipulation actions, including tapping and sliding, across surfaces with varying friction and stiffness. This allows us to record both intrinsic and extrinsic contact interactions. We collected eight sequences with the Allegro hand, each lasting approximately 8.5 minutes, recording data from all four fingers. From the mobile picker, we gathered 104 sequences with an average duration of 3 minutes, logging data from both sensors on the gripper. For the subset of the dataset collected with the mobile picker, we provide annotations indicating the object in grasp, the action performed, and the surface in contact, to support downstream evaluation tasks. In total (see Fig. C.1), our dataset spans 18.6 hours of tactile data collected from six different Digit 360 sensors.

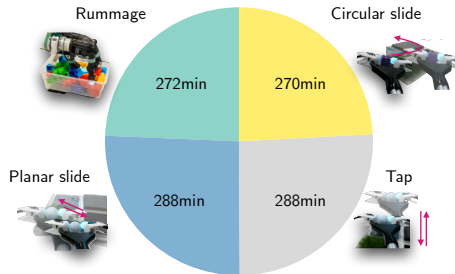


Figure C.1: Distribution of recorded Digit 360 data by platform. The dataset includes 18.6 hours of data collected using two platforms: the Allegro hand (4.5 hours) and the mobile picker tool (14.1 hours).

Sparsh-X processes temporal windows of data from each tactile modality. A visualization of the input data is shown in Fig. C.2.

Images. We input pairs of tactile images sampled with a temporal stride of 5, concatenated along the channel dimension. These images are captured using a hyperfisheye lens in Digit 360, allowing us to view the entire dome-shaped elastomer surface. Unlike planar GelSight-like sensors, these images include reflections from the surrounding LED light sources, visible near the center of the dome. While these reflections act as useful markers, encoding meaningful information about gel deformation upon contact, they also pose challenges for standard preprocessing techniques such as background subtraction or lighting augmentations, which risk corrupting the contact signal.

Audio. Digit 360 sensor is equipped with two contact microphones that capture vibrations, sampled at 48kHz. This signal is especially informative for detecting changes in contact state, such as making and breaking contact. **Sparsh-X** processes 0.5sec windows of audio data from each microphone in the frequency domain. After standardization and conversion to log-mel spectrograms, the audio is treated as a single-channel image input to the model.

IMU and Pressure. We extract 0.5 second and 1 second windows of data from the 3-axis accelerometer and the static pressure sensor embedded in the Digit 360. Each window is standardized using the mean and standard deviation computed per sequence to ensure consistency across variations in sensor signal amplitude.

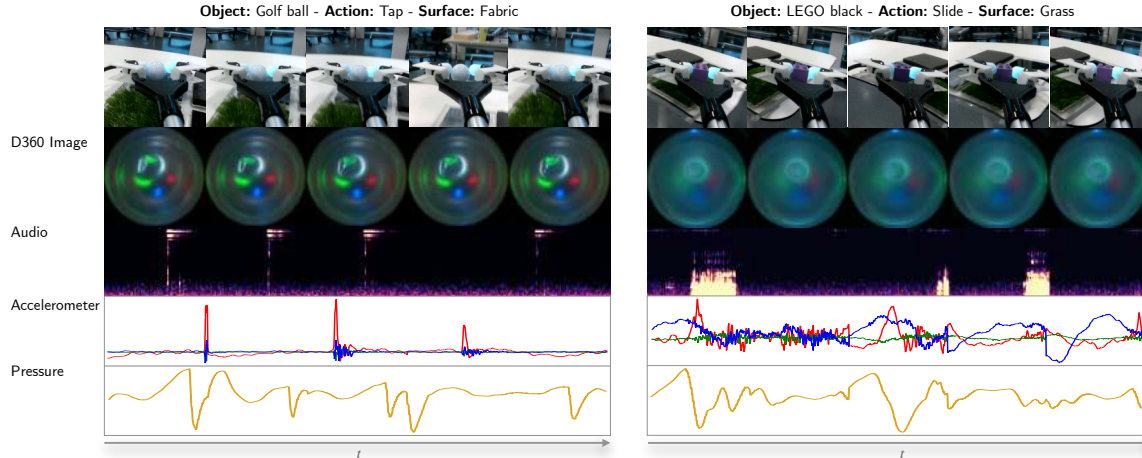


Figure C.2: Visualization of each of the tactile input modalities to **Sparsh-X**.

C.2 Datasets for Downstream Tasks

Material-Quantity Estimation. We design a 3D-printed gripper attachment to mount the Digit 360 sensors onto the Franka arm. The data collection protocol involves shaking six different 8oz bottles containing various materials (lentils, rice, corn kernels, vitamin pills, water, and oil) at different fill levels (full, half, quarter). An

illustration of this setup is provided in Fig. C.3. For shaking the bottles to create variation in the tactile signal, the Franka’s gripper is rotated left and right in randomized motion patterns, including variation in the initial angle. We collect 20 trajectories for each material-quantity combination, using 15 sequences for training and reserving the remaining 5 for evaluating the classifier.

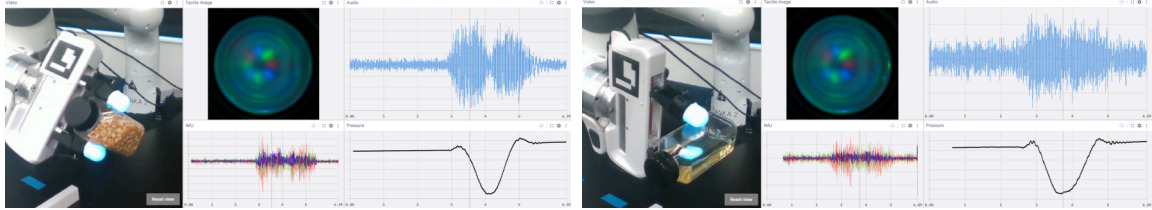


Figure C.3: Visualization of the experimental setup and tactile sensory inputs for the material-quantity classification dataset. The setup involves shaking bottles filled with different materials and quantities using the Franka’s gripper equipped with Digit 360 sensors.

Normal Force Estimation. We fix a hemispherical probe to a force/torque sensor and mount a Digit 360 sensor on the Meca arm, which is used to indent the elastomer surface perpendicularly, applying controlled normal forces of up to 3.5N. Fig. C.4 illustrates the experimental setup and the distribution of the collected data. We observe that the pressure modality correlates strongly with both the magnitude of the applied normal force and the location of the resulting deformation on the elastomer. The audio modality captures discrete events, such as the initial contact between the probe and the sensor surface.

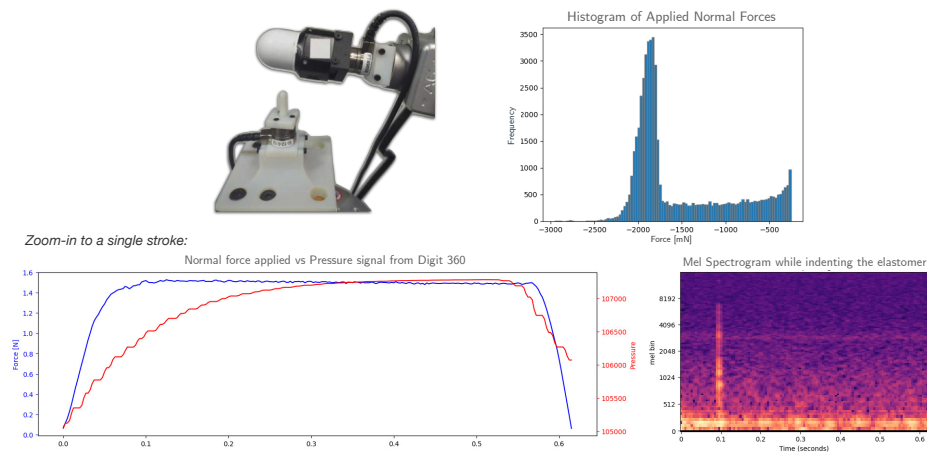


Figure C.4: **Top:** Experimental setup and data distribution for the normal force regression experiment. **Bottom:** Zoom-in on a single indentation stroke. Note that the pressure signal from the Digit 360 sensor correlates well with the ground-truth normal force measured by the force/torque sensor beneath the hemispherical probe. The mel spectrogram also reveals the moment of contact between the probe and the elastomer.

C.3 Benchmarking Sparsh-X for physical properties comprehension

Object-Action-Surface Classification. This task evaluates whether Sparsh-X can capture tactile cues that enable the identification of objects through both intrinsic and extrinsic contact interactions. The goal is to jointly classify the object being grasped, the action performed, and the surface in contact. The selected objects and surfaces span a range of properties, including texture, hardness, and friction.

We use representations from Sparsh-X to train a downstream classifier on the dataset described in Appendix C.2. Fig. C.5 shows confusion matrices on the test set for two classifiers: one trained using frozen Sparsh-X representations with all tactile modalities as input, and another trained end-to-end (E2E) using only tactile images. Note that the classification task involves 36 classes, representing all combinations of object, action, and surface. The results shown in the figure correspond to training with 50% of the labeled training set.

The Sparsh-X-based classifier shows stronger diagonal alignment, indicating more accurate predictions across the 36 joint object-action-surface classes. In contrast, the E2E model suffers from greater confusion among similar classes, particularly those with overlapping surface or action components (e.g., misclassifying "Tap-Foamwork" as "Tap-Fabric" or "Slide-Plastic"). These results highlight the benefit of multimodal tactile representations: incorporating audio, motion (IMU), and pressure modalities helps disambiguate fine-grained contact dynamics that are challenging to capture with images alone.

Material-Quantity Estimation. This task further evaluates Sparsh-X's ability to comprehend physical properties. Specifically, we focus on distinguishing materials based on their granularity and viscosity (e.g., solids and liquids), as well as estimating mass through coarse volume classification. We train a classifier to predict one of 18 joint classes, each representing a unique combination of material type and quantity level. The classifier is trained either using frozen Sparsh-X representations or end-to-end (E2E) from tactile images alone.

Fig. C.6 shows the confusion matrices for the material-quantity classification task when trained with 33% of labeled data, comparing an end-to-end (E2E) classifier trained solely on tactile images with a classifier trained on frozen Sparsh-X representations. The E2E model achieves 68.8% accuracy, while the Sparsh-X-based classifier reaches 87.5%, highlighting the benefit of multimodal tactile representations. In the E2E setting, we observe frequent confusion between different fill levels of the same material and between visually similar liquids such as oil and water. In contrast, the Sparsh-X-based classifier exhibits strong diagonal alignment, suggesting accurate identification across the 18 material-quantity classes.

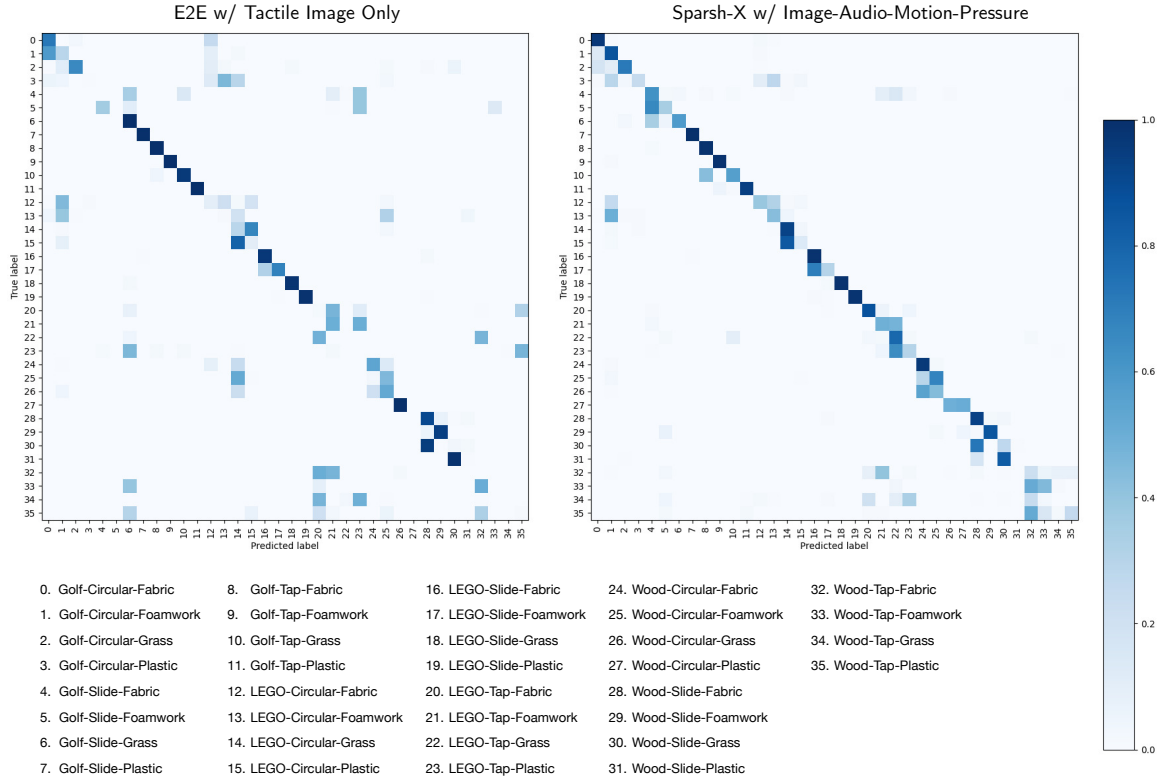


Figure C.5: Confusion matrix for object-action-surface classification. We compare an end-to-end classifier trained solely on tactile images with a classifier trained on frozen **Sparsh-X** representations, under a 50% training data budget.

C.4 *Sparsh-X* and Policy Learning

C.4.1 Real-World *Sparsh-X* and Policy Deployment

For real-world deployment of **Sparsh-X**, we use ROS2. We maintain circular buffers of 5 seconds for each tactile modality per Digit 360 fingertip. For synchronization, we use the timestamp of the image modality as the reference, selecting the closest-in-time sample from the other modalities (audio, motion from accelerometer, and pressure). Once inputs are processed, **Sparsh-X** can run inference at 50Hz on a GPU RTX 4090. However, the construction of log-mel spectrograms remains the main computational bottleneck for real-time processing. When processing all four Digit 360 sensors on the Allegro hand, end-to-end inference with **Sparsh-X** runs at approximately 20Hz. Fig. C.7 shows the deployment pipeline for policy experiments.

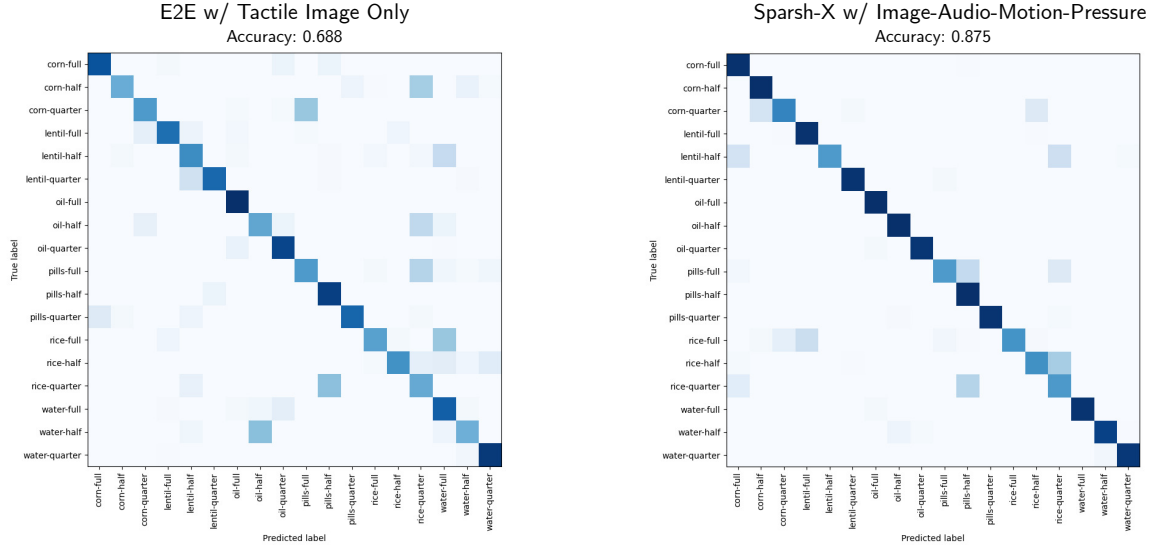


Figure C.6: Confusion matrix for material-quantity estimation. We compare an end-to-end classifier trained solely on tactile images with a classifier trained on frozen **Sparsh-X** representations, under a 33% training data budget.

C.4.2 Plug-Insertion via Imitation Learning

The robot, equipped with an Allegro hand and sensorized with Digit 360 fingertips, is tasked with inserting a pre-grasped plug into the first socket of an extension power strip. In our experimental setup, the socket position remains fixed, while the starting position of the robot arm is randomized within a 3D cuboid of $(5, 5, 2)$ cm around the nominal starting pose.

The model inputs include an embedding of the wrist camera image and **Sparsh-X** representations for thumb, index, and middle finger sensors, processed through an attentive pooling layer [25]. We train a ResNet18 [82] in an end-to-end (E2E) fashion to learn the wrist image embeddings. We append learnable action tokens, which are processed by the transformer and subsequently decoded into a sequence of actions. In our setup, the model predicts a sequence of absolute robot end-effector poses i.e., $\mathbf{a} \triangleq (\mathbf{T}_t, \mathbf{T}_{t+1}, \dots, \mathbf{T}_{t+H})$ with a prediction horizon of $H = 8$. An illustration of the plug insertion architecture is shown in Figure C.8.

C.4.3 In-Hand rotation with sim-to-real tactile adaption

Hora [147] is a two-stage policy that rotates objects along the z -axis. The first stage trains the policy using privileged information, which includes the object’s state or pose, local shape, mass, friction, and other physical properties that can be perceived by the fingertips. The second stage trains an adaptation module to approximate the

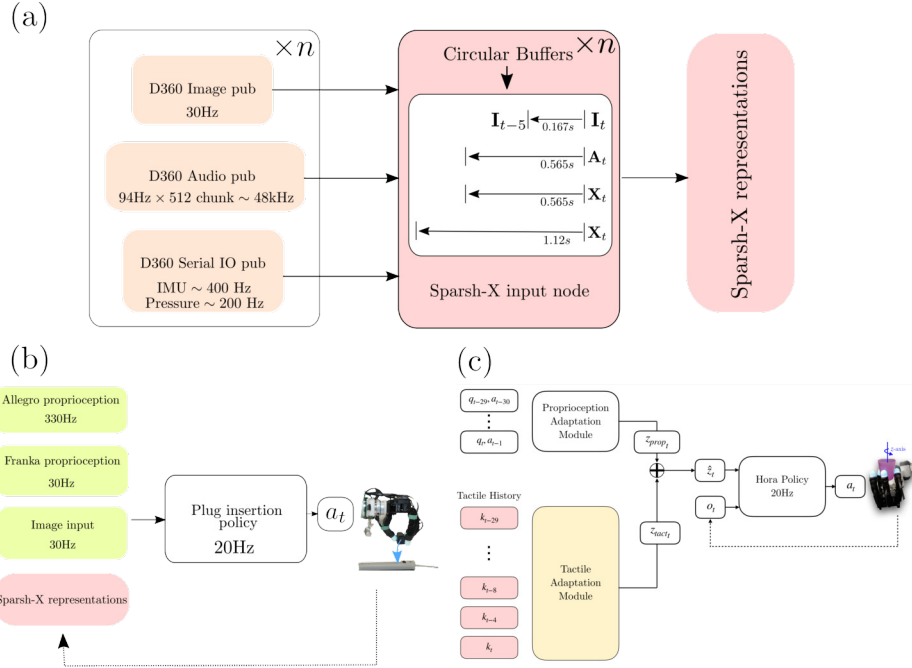


Figure C.7: Real-world policy deployment architecture: We use ROS2 middleware for policy deployment, and PyTorch for deep learning modules. In addition to the proprioceptive states of the robot and optional third-person vision modality, downstream policies take as input **Sparsh-X** representations from upto 4 fingertips of the Allegro hand. (a) illustrates how the inputs are constructed for **Sparsh-X**, (b) illustrates policy deployment for the plug insertion policy, and (c) illustrates the policy deployment for the in-hand rotation (Hora) policy.

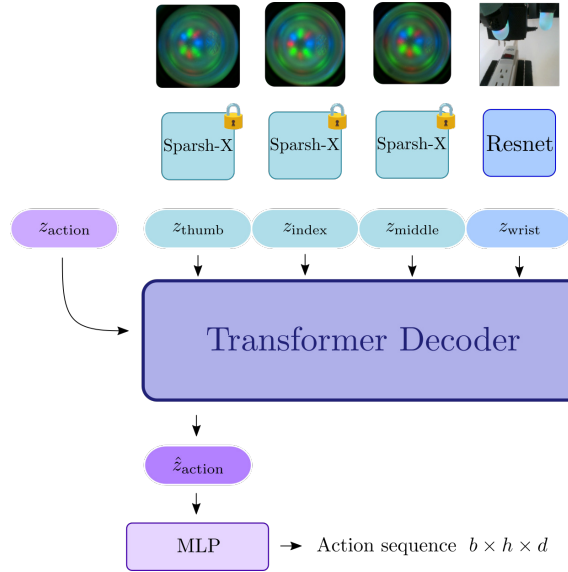


Figure C.8: Architecture overview of the plug insertion policy. A transformer decoder is trained to generate action sequences based on **Sparsh-X** representations from three fingertips, a wrist camera image capturing the current robot state, and a learnable latent action code. All embeddings are concatenated and processed by a lightweight MLP to decode the next end-effector action.

latent space of the privileged information from the discrepancy between observed proprioception history and commanded actions, which implicitly informs about contact.

Although the approximation of the privileged vector from proprioceptions transfers to the real setup, it operates with incomplete information about the object’s state. With multisensory touch sensing at the fingertip level, privileged information such as changes in object pose, slip, and friction are now accessible in real-world scenarios, albeit not directly. We can leverage **Sparsh-X** representations to fine-tune the real-world approximation of the privileged information embedding. The goal is to do *tactile adaptation* on top of the baseline policy to enhance stability during object rotation.

We pass to the tactile adaptation module frozen **Sparsh-X** representations for each of the four fingers in the Allegro-hand, with a temporal stride of 0.19s, equating to 8 touch representations per finger over a 1.5s window, which matches the proprioception state history consumed by the baseline Hora. Features for each finger are pooled using attentive pooling to create a global representation, which is then concatenated along the temporal dimension, resulting in a $(t \times n) \times 768$ input embeddings. The tactile adaptation model to be trained is a shallow MLP followed by the zero-convolution layer.

Our dataset consists of successful rollouts of the Hora policy, where the object keeps rotating without touching the palm for at least 30 seconds. The data is serialized into the lerobot dataset format [18], sampled at a control frequency of 20Hz. For training the tactile adaptation module, our objective is to minimize the L2 loss between the real-world hand joint angles and the target action given by the frozen Hora policy under the tactile-informed privileged embedding.

Appendix D

VISUO-TACTILE WORLD MODEL SUPPLEMENTARY MATERIAL

D.1 Training Visuo-Tactile World Model

D.1.1 Training dataset

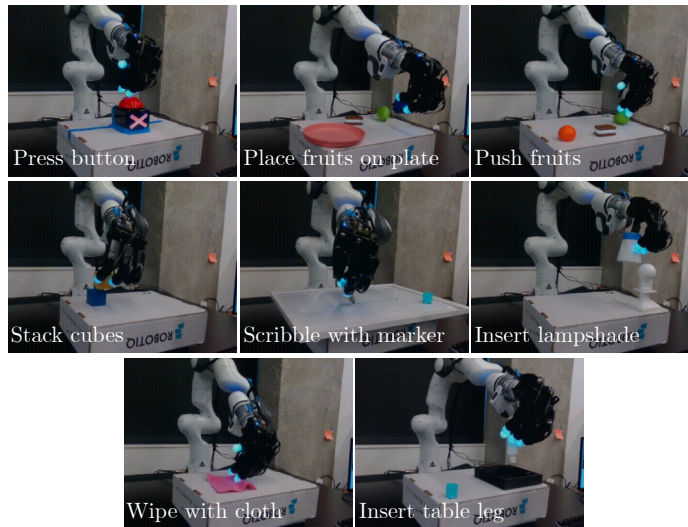


Figure D.1: Multitask Vision-Tactile Dataset. Trajectories for training the world model collected via teleoperation, including both successful and failure sequences.

To train our multi-task visuo-tactile world model, we collect a dataset of teleoperated robot arm trajectories performing fundamental contact-rich manipulation actions, such as pick and place, pushing, and insertion. Our hardware setup consists of a table-top Franka Panda arm with an Allegro Hand as the end-effector and a Digit 360 sensor mounted on each fingertip. An exocentric view from a camera captures the global context of the robot’s interaction with objects on the table.

Through teleoperation, we collect a diverse set of trajectories, without discriminating between successes and failures, for eight distinct contact-rich tasks (see Fig. D.1): pick and place on a plate, reach and press a button, push, wipe with a cloth, lampshade insertion, table leg insertion, cube stacking, and scribbling with a marker. For each task, we recorded successful and failure demonstrations. Each sequence contains

multimodal data streams: proprioceptive information (wrist pose, joint positions), exocentric video from the camera, and video from each Digit 360 fingertip sensor. All data streams were synchronized using timestamps and downsampled to 6 FPS for training the world model. Our training dataset for V-WM and VT-WM consists of 124 demonstrations totaling 112k datapoints, with each demonstration averaging 40 seconds. For validation, we use 26 demonstrations spanning all tasks, comprising 17k datapoints.

D.1.2 Training parameters

The model is optimized using AdamW [125] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.95$ and a weight decay of 0.01. We use a learning rate scheduler with linear warmup for the first 10,000 gradient updates to a peak learning rate of $3e-4$, followed by cosine decay to $3e-7$ over a total of 80,000 updates. We use an effective batch size of 64 distributed over 32 A100 GPUs. We found that fine-tuning the Sparsh-X encoder was beneficial to account for sensor-specific variations, such as those arising from manufacturing tolerances and elastomer wear, while the Cosmos Tokenizer [1] was kept frozen during training. Our visuo-tactile world model has a total of 173M parameters, of which 96M were trained.

D.2 Contact perception with Visuo-Tactile World Model

We corroborate the world model’s capacity to generate future states that are a reliable and predictable consequence of the given action conditioning. We study action controllability qualitatively by visualizing rollouts under simple, disentangled action commands: moving the end-effector along the Cartesian axes ($\pm x$, $\pm y$, $\pm z$) and opening/closing the hand. Actions conditioning is given to the VT-WM as deltas in the robot’s proprioceptive state.

We observe in Fig. D.2 that the VT-WM produces coherent rollouts aligned with the commanded actions. Translations along each axis result in consistent directional displacements of the end-effector in imagination (notice the reference frame in the figure), while hand open/close commands lead to corresponding changes in finger configurations. Notably, these behaviors emerge from the learned dynamics rather than explicit supervision of axis-aligned motion, indicating that the model internalizes the action-conditioned structure of the robot’s kinematics. We compare ground-truth trajectories with VT-WM rollouts under the same action sequences and illustrate what the world model *imagines* in terms of contact.

To illustrate the predictive capability of the visuo-tactile world model, we evaluate rollouts conditioned on real robot action sequences. Specifically, we use held-out demonstrations from two tasks in our dataset: *press button* and *scribble with marker*. For each task, the VT-WM is queried autoregressively using the ground-truth sequence of

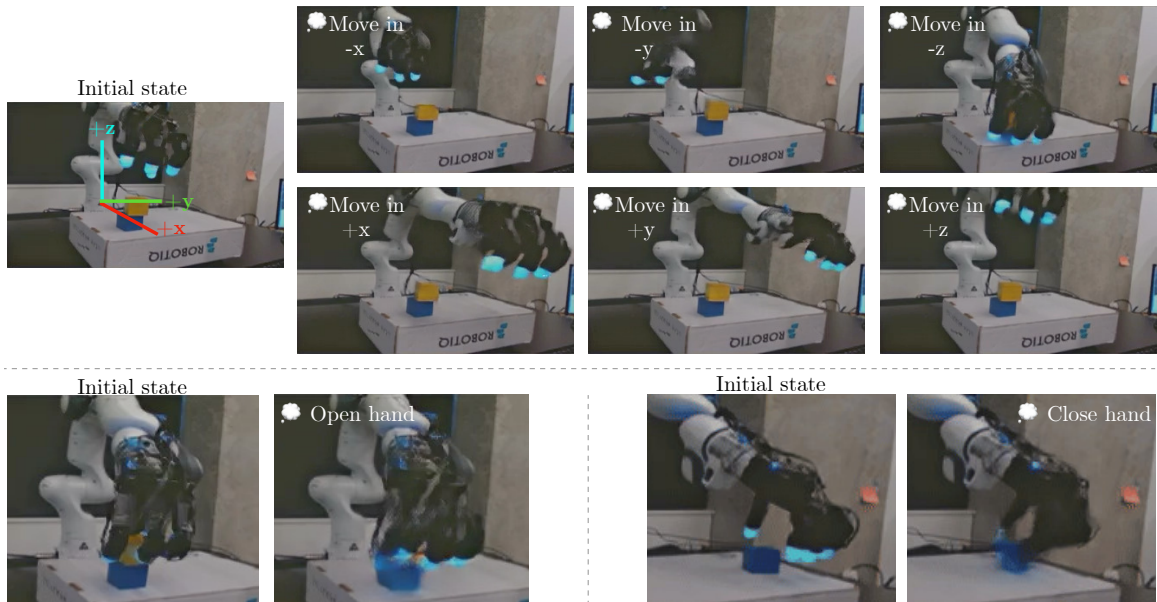


Figure D.2: Visuo-Tactile World Model generates rollouts aligned with commanded actions along reference axes ($\pm x$, $\pm y$, $\pm z$) and for hand open/close.

control deltas. Since the model produces latent representations of future visual and tactile observations, we employ pretrained decoders to reconstruct these latents for visualization. From Fig. D.3, across both tasks the predicted visual states closely resemble the final RGB images of the real trajectories. The predicted tactile states also capture the key interaction events: although slight differences appear in the precise location of per-finger contacts, the rollouts consistently indicate whether contact occurs and depict plausible patterns of hand-object interaction. This demonstrates that the VT-WM, when guided by real action sequences, generates in imagination physically meaningful futures across both visual and tactile modalities.

Fig. D.4 compares snapshots of a visuo-tactile world model rollout for the *insert table leg* and *cube stacking* tasks. We show the difference between the No-Contact and predicted tactile state during the trajectory. Tactile is very localized, although the predicted tactile states showcase contact unless the object is released. By producing consistent visuo-tactile rollouts under real control sequences, VT-WM demonstrates the ability to represent both global visual context and local contact dynamics in a unified predictive framework useful for planning.

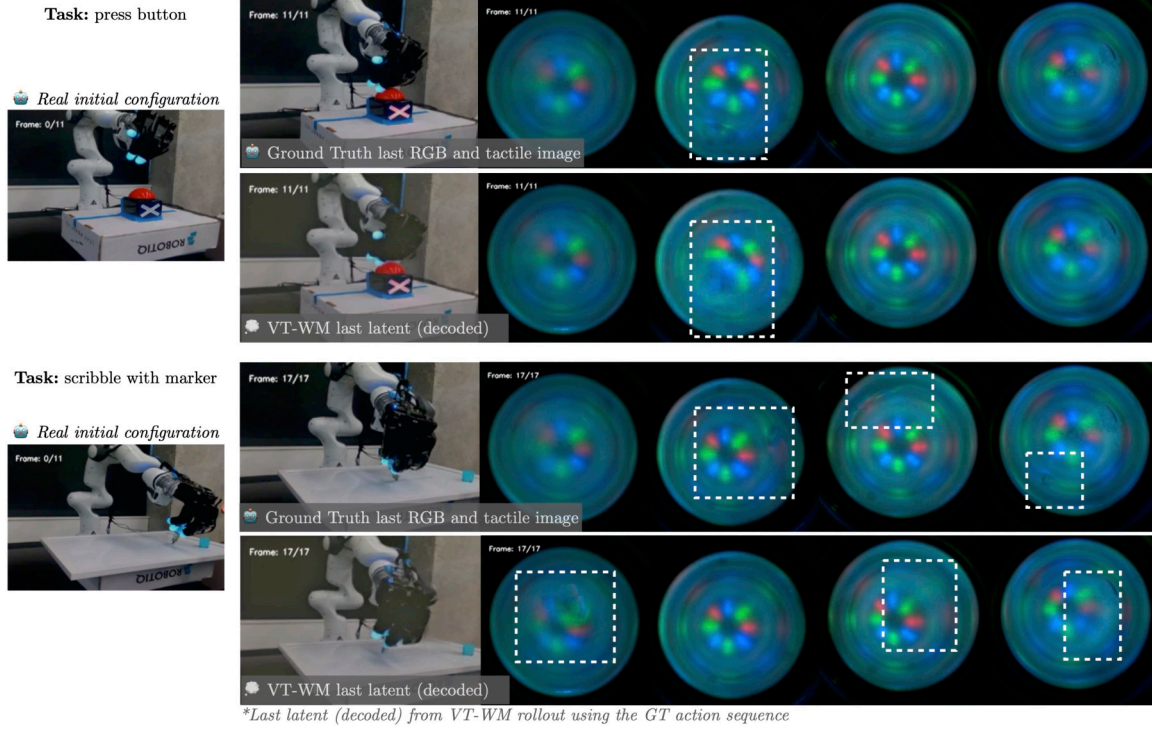


Figure D.3: Visuo-Tactile World Model rollouts conditioned on ground-truth action sequences. Predicted visual states closely match the final RGB observations, while predicted tactile states capture plausible contact events and finger–object interactions.

D.3 Zero-shot planning with World Models

D.3.1 CEM algorithm for planning with World Models

[Algorithm 1](#) describes planning in world model (WM) imagination using the Cross-Entropy Method (CEM). Given a goal image and the current multimodal context (vision and tactile), the algorithm first encodes these inputs into latent representations. CEM is then used to optimize a sequence of actions over a finite prediction horizon by iteratively sampling action sequences (particles), rolling them out in the world model, and evaluating their predicted visual outcomes against the goal latent state using an ℓ_2 distance. The top-performing action sequences are used to update the mean and variance of the action distribution, refining the search over multiple iterations. After convergence, the best action sequence is executed on the robot, and the process can be repeated over several trials with updated context.

Next, we describe the goal of each of the tasks we use to evaluate the planning capabilities of the world models and discuss their results.

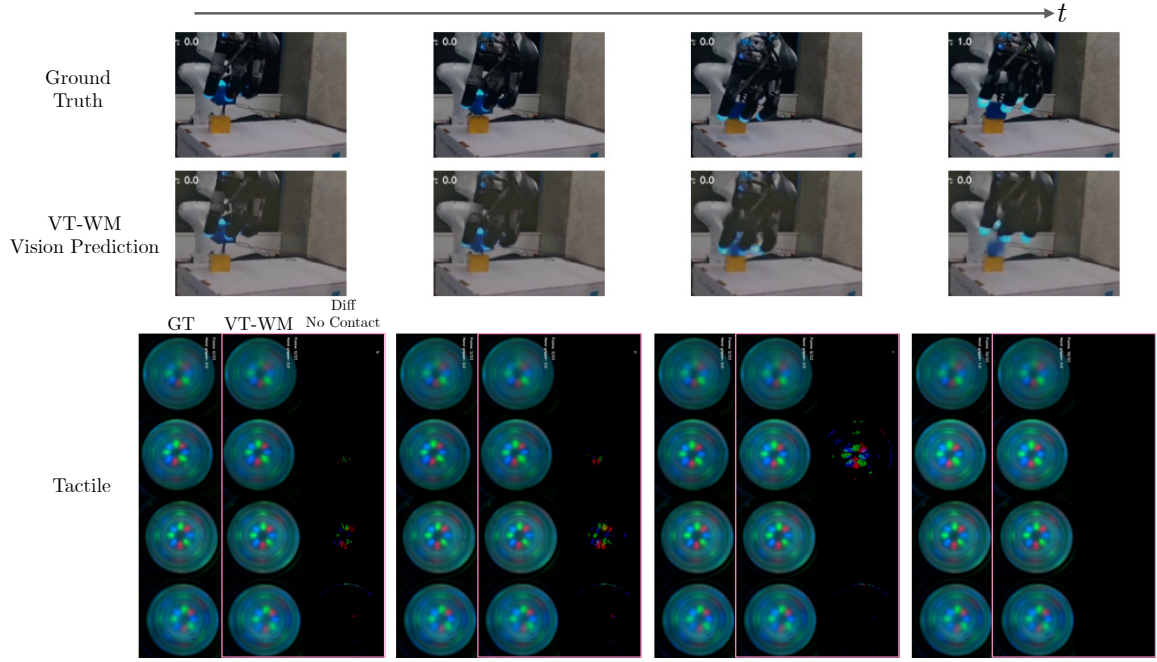
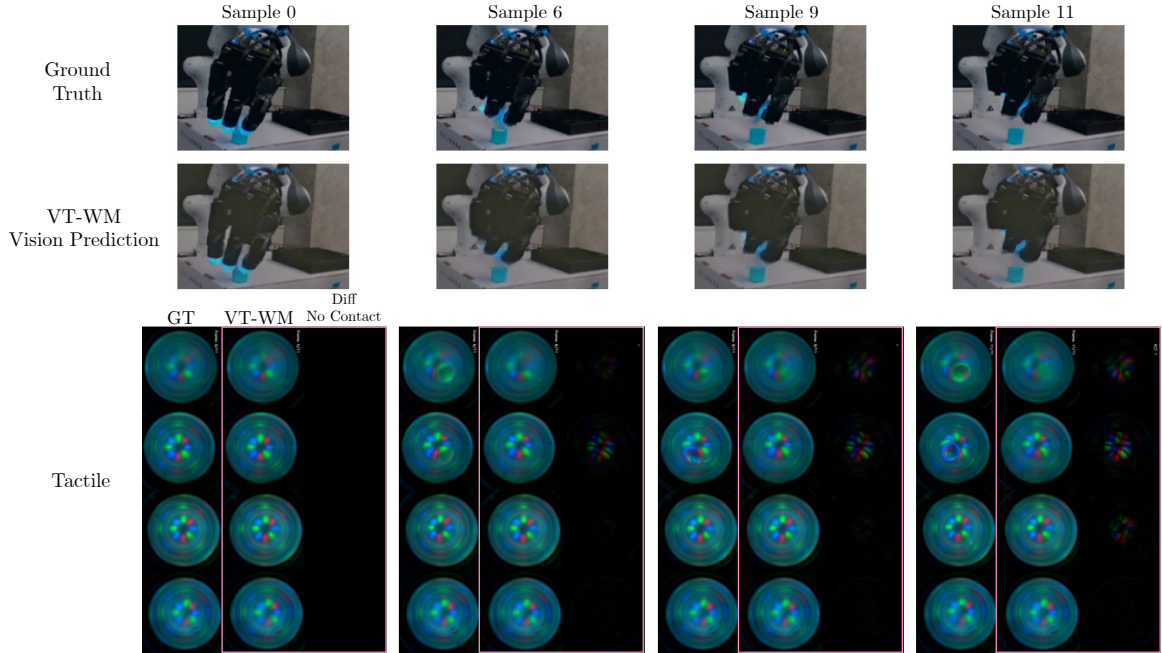
(a) Rollout for *cube stacking* task.(b) Rollout for *insert table leg* task.

Figure D.4: Snapshots of Visuo-Tactile World Model rollouts. For both tasks: *Top*: ground truth vision state; *Middle*: predicted vision state; *Bottom*: ground truth and predicted tactile signatures. Note the consistency in contact matching between ground truth and VT-WM predictions.

Algorithm 1 Planning in WM imagination via CEM

Require: WM (world model)

Require: Goal Image X_{rgb}^{goal}

Require: Context (current state) X_{rgb}^0, X_{touch}^0

$f \leftarrow 6$ ▷ WM frequency

$H \leftarrow 2$ ▷ Prediction horizon in seconds

$P \leftarrow 36$ ▷ Number of particles for CEM algorithm

$N \leftarrow 10$ ▷ Number of iterations for CEM algorithm

$d \leftarrow 7$ ▷ Action dimensionality [X,Y,Z,roll,pitch,yaw,gripper]

max-trials $\leftarrow 3$ ▷ Number of calls to CEM algorithm

for trials $<$ max-trials **do**

Update context (current state) X_{rgb}^0, X_{touch}^0 ▷ Read from sensors

$Z_{rgb}^{goal} \leftarrow \text{vision-encoder}(X_{rgb}^{goal})$ ▷ Encode goal image

$Z_{rgb}^0 \leftarrow \text{vision-encoder}(X_{rgb}^0)$ and $Z_{touch}^0 \leftarrow \text{touch-encoder}(X_{touch}^0)$ ▷ Encode context

▷ Initialize CEM action distribution parameters

$\mu \leftarrow \text{zeros}(1, H * f, d)$ and $\sigma \leftarrow \text{ones}(1, H * f, d)$

best-cost $\leftarrow \infty$

best-action $\leftarrow \text{None}$

for $n < N$ **do**

▷ Generate action particles

$actions \leftarrow (\mu.\text{repeat}(N) + \sigma.\text{repeat}(N)) * \text{rand}(N, H * f, d)$

$\hat{Z}_{rgb}^{1:H}, \hat{Z}_{touch}^{1:H} \leftarrow \text{WM.rollout}(Z_{rgb}^0, Z_{touch}^0, actions)$ ▷ Rollout WM

$costs \leftarrow \ell_2(Z_{rgb}^{goal}, \hat{Z}_{rgb}^H)$ ▷ Compute distance with target latent state

elite-actions $\leftarrow actions[\text{topk}(costs)]$ ▷ Choose top 5 particles with lowest cost

▷ Update distribution parameters

$\mu \leftarrow \text{elite-actions.mean}()$ and $\sigma \leftarrow \text{elite-actions.std}()$

if $costs.min() < \text{best-cost}$ **then**

best-action $\leftarrow actions[costs.argmax()]$ ▷ Found a new action that gets closer

to goal

end if

end for

Execute sequence of robot commands from best-action

end for

D.3.2 CEM plans and zero-shot transfer to real world

Reach Button: In this task, the robot must approach and press the center of a button starting from varied initial poses directly above it. Success therefore requires planning a sequence of actions that align the end-effector laterally with the button and then move downward to establish contact. Fig. D.5 illustrated the plans produced by each world model using CEM rollouts in imagination, alongside the corresponding executions on the real robot. We observe that both V-WM and VT-WM generate feasible trajectories that transfer zero-shot to the real system. This is expected since reaching primarily involves spatial reasoning and gross kinematic alignment, which vision alone can capture reliably.

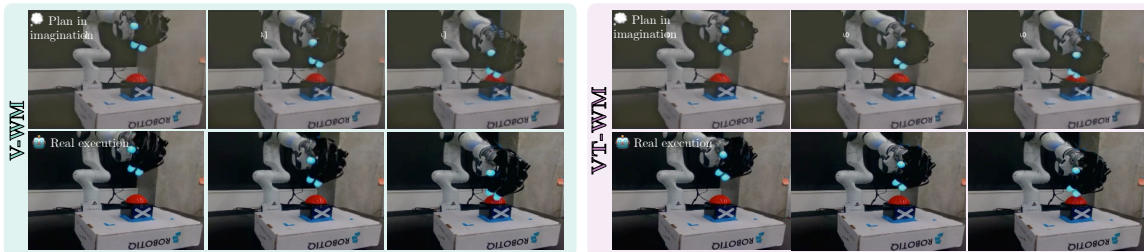


Figure D.5: Reach Button task. Plans generated by V-WM and VT-WM with CEM in imagination (top) and their zero-shot executions on the real robot (bottom). Both models produce feasible trajectories, as reaching relies mainly on spatial reasoning and kinematic alignment.

Push Fruits: In this task, the robot hand begins directly in front of a target object that must be pushed downwards (toward the robot base). A successful plan requires maintaining persistent but gentle contact, allowing the object to slide across the table rather than topple.

Fig. D.6 compares imagined rollouts and real executions for both models. While both V-WM and VT-WM produce plausible plans, we observe notable artifacts in the imagined rollouts, most prominently visual distortions of the green fruit when the hand occludes it. These artifacts are less pronounced in VT-WM, which better preserves the object’s geometry in imagination. The deployment of the V-WM plan not only results in shorter object displacement but also lead to physical failures in execution, where the object occasionally topples instead of sliding without orientation changes.

Reach & Push: This task requires a two-stage plan, first reaching the object to establish contact, then pushing it downward toward the robot base. Both subgoals are illustrated in Fig. D.7, which shows the imagined plans and real executions.

In the V-WM rollout, the hand consistently hovers slightly above the object during the reach phase. As a result, the subsequent push proceeds without contact, and

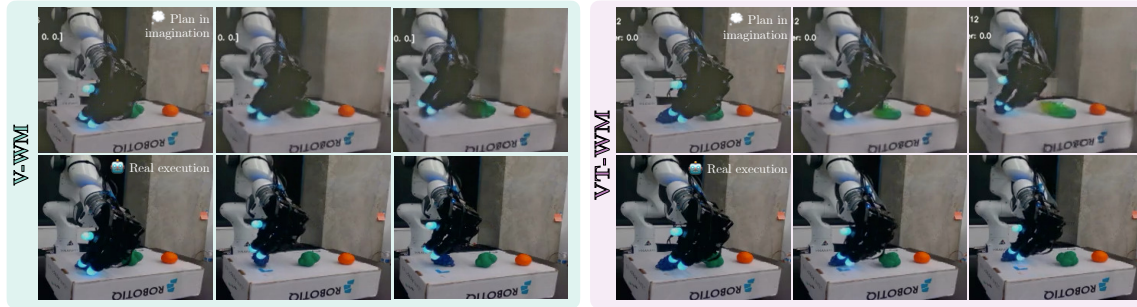


Figure D.6: Push Fruits task. VT-WM preserves object geometry in imagination and transfers to stable sliding, while V-WM introduces distortions and often causes toppling in execution.

the execution on the real robot fails to move the object. By contrast, the VT-WM rollout explicitly brings the hand into contact during the reach, enabling the push plan to apply move the object effectively. When deployed, this produces the desired behavior, with both the reach and push subgoals successfully achieved. This highlights how tactile grounding resolves cases of visual aliasing, ensuring reliable contact in imagination and execution.

Wipe Cloth: This task consists of two subgoals, first reaching the cloth to establish contact, and then wiping it horizontally across the table. Both stages are illustrated in Fig. D.8, which compares imagined rollouts with real executions.

In the V-WM rollout, the reach phase frequently results in the hand hovering slightly above the cloth, leading to ineffective wiping during execution. Even when a wiping trajectory is imagined, the visualizations exhibit noticeable artifacts such as geometric distortions of the cloth and hand. These artifacts reflect the model’s uncertainty about contact dynamics and correspond to execution failures where the cloth barely moves.

In contrast, the VT-WM rollout shows clearer geometry and maintains consistent contact with the cloth in imagination. As a result, the subsequent wiping action produces a stable horizontal displacement of the cloth when deployed on the real robot. This example underscores the advantages of visuo-tactile world model in tasks that require sustained contact to manipulate objects.

Stack Cubes: This task requires transporting a blue cube to the stacking location and then placing it stably on top of a yellow cube. Both subgoals are illustrated in Fig. D.9, which shows imagined rollouts alongside real executions.

While the V-WM generates reasonable transport trajectories, failures arise during placement. In imagination, the cube intermittently disappears from the hand, revealing artifacts that indicate the model is tracking only the hand–scene geometry (e.g.,

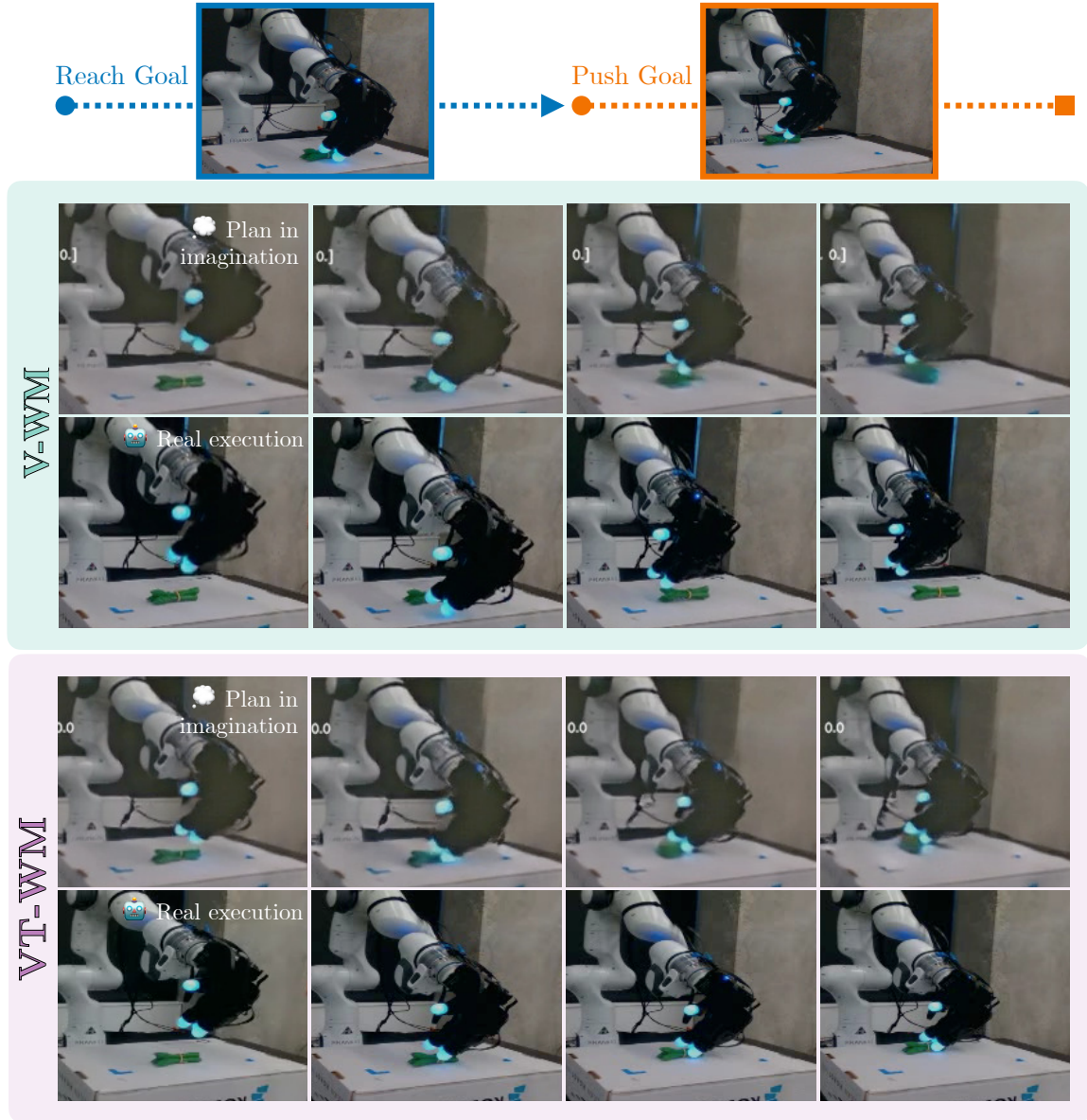


Figure D.7: Reach & Push task. V-WM fails to establish contact, leading to ineffective pushes, while VT-WM ensures contact in imagination and execution, successfully completing both subgoals.

alignment with the target yellow cube) rather than maintaining a consistent hand-object relationship. This disconnect leads to execution failures, where the cube is not reliably placed.

However, visuo-tactile world model accurately captures the object-hand interaction, throughout both transport and placement, the cube remains consistently represented in the rollout. When transferred zero-shot to the real robot, these plans result in

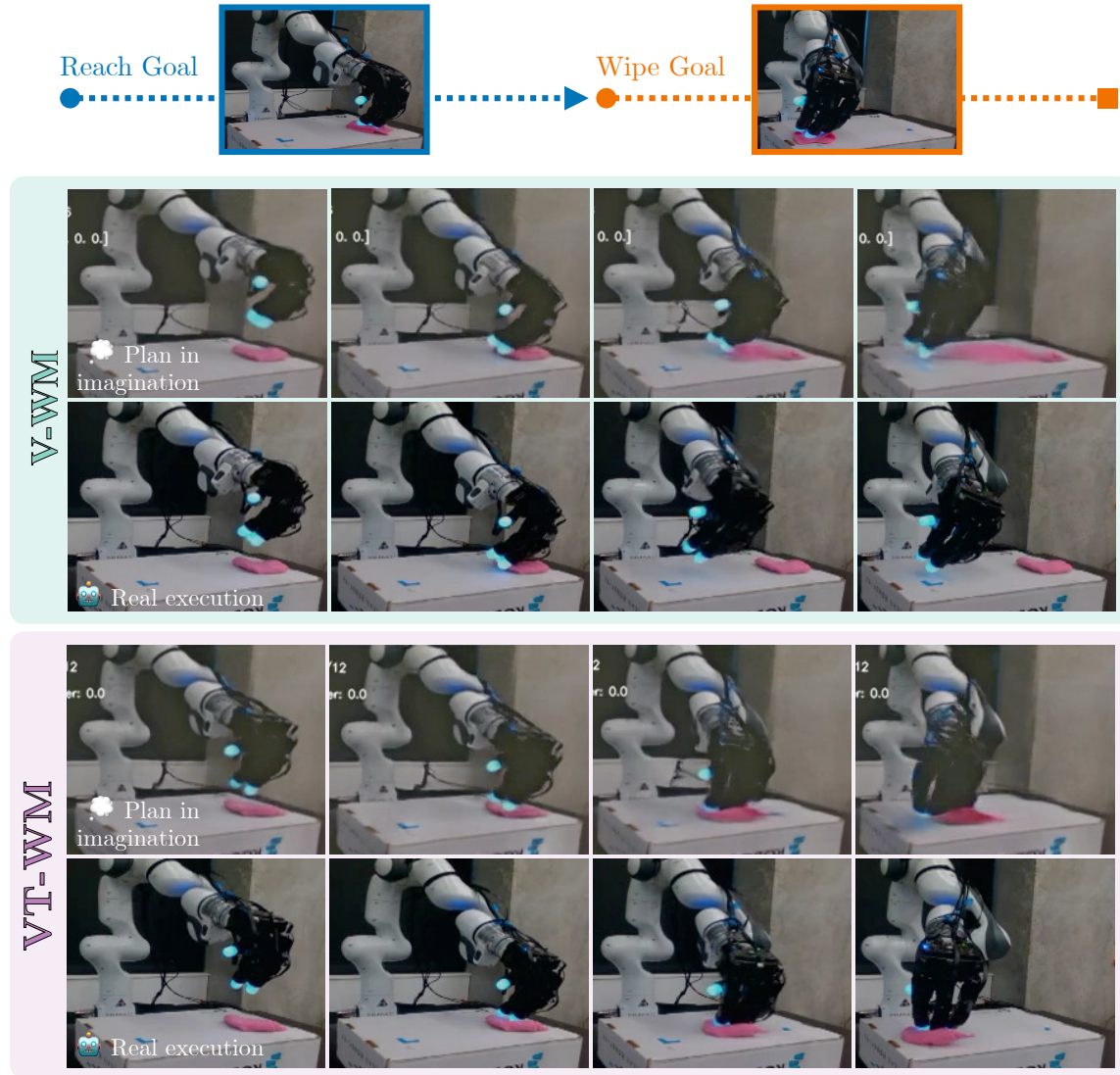


Figure D.8: Wipe Cloth task. V-WM rollouts show artifacts and miss contact, leading to ineffective wiping, while VT-WM maintains contact and produces consistent cloth displacement in execution.

stable stacking, highlighting the advantage of VT-WM for tasks that demand precise, contact-rich manipulation.

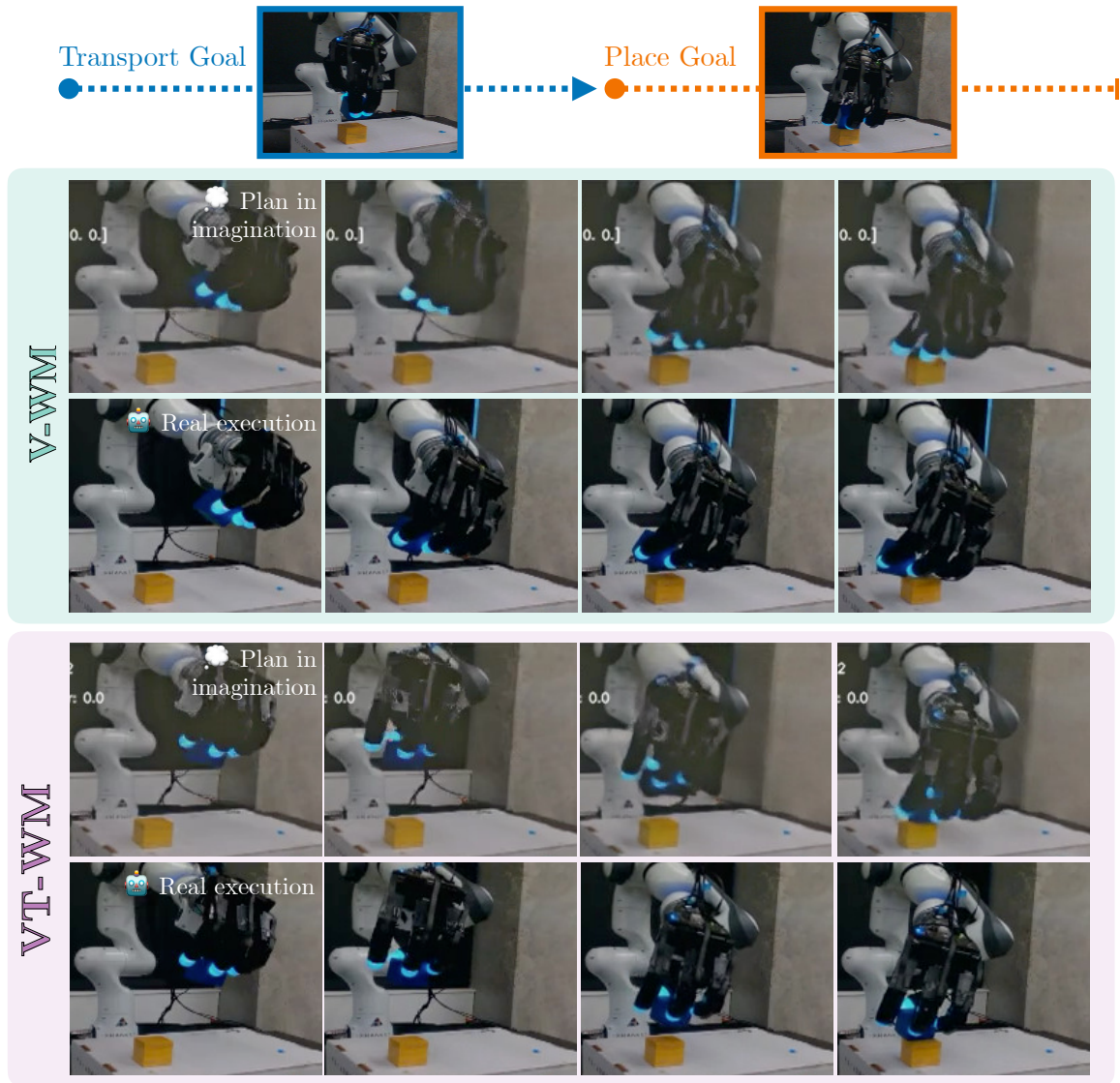


Figure D.9: Stack Cubes task. V-WM imagined rollouts during planning lose track of the cube for the placement subgoal, leading to failed stacking, while VT-WM preserves hand-object interaction and transfers to successful stacks.