

---

# ViTacFormer: Learning Cross-Modal Representation for Visuo-Tactile Dexterous Manipulation

---

Liang Heng<sup>\*1,2,3</sup> Haoran Geng<sup>\*†1</sup> Kaifeng Zhang<sup>3</sup>

Pieter Abbeel<sup>1</sup> Jitendra Malik<sup>1</sup>

<sup>1</sup>University of California, Berkeley <sup>2</sup>Peking University <sup>3</sup>Sharpai

<sup>\*</sup>Equal Contribution <sup>†</sup> Project Lead

## Abstract

Dexterous manipulation is a cornerstone capability for robotic systems aiming to interact with the physical world in a human-like manner. Although vision-based methods have advanced rapidly, tactile sensing remains crucial for fine-grained control—particularly in unstructured or visually occluded settings. We present ViTacFormer, a representation-learning approach that couples a cross-attention encoder to fuse high-resolution vision and touch with an autoregressive tactile-prediction head that anticipates future contact signals. Building on this architecture, we devise an easy-to-challenging curriculum that steadily refines the visual-tactile latent space, boosting both accuracy and robustness. The learned cross-modal representation drives imitation learning for multi-fingered hands, enabling precise and adaptive manipulation. Across a suite of challenging real-world benchmarks, our method achieves approximately 50% higher success rates than prior state-of-the-art systems. To our knowledge, it is also the first to autonomously complete long-horizon dexterous manipulation tasks that demand highly precise control with an anthropomorphic hand—successfully executing up to 11 sequential stages and sustaining continuous operation for 2.5 minutes.

## 1 Introduction

Recent years have seen rapid advances in robotic manipulation[12, 9, 22, 10, 25, 45, 11, 17, 18, 7], with behavior cloning [28, 27, 30] emerging as a promising method for high-precision tasks in real-world settings. However, most existing work remains limited to simple hand configurations[47] and exhibits poor generalization—largely due to the underutilization of tactile sensing[36, 31, 32, 44], which is essential for fine-grained control.

While some studies have begun integrating tactile feedback into dexterous manipulation [19, 21], the learned tactile representations are often shallow and underexplored. Some works focus on replicating the success of self-supervised learning to learn representations for tactile signals [38, 37, 40, 14, 8]. However, there is still a lack of an effective model that learns cross-modal representations for visuo-tactile dexterous manipulation [24, 41]. We address this limitation with ViTacFormer, a unified visuo-tactile framework that enables fine-grained, generalizable manipulation through deep cross-modal representation learning.

We address this gap with **ViTacFormer**, a unified visuo-tactile framework for dexterous manipulation. Our key idea is a cross-modal representation built with cross-attention layers that fuse visual and tactile cues at every stage of the policy. Crucially, we argue, and empirically confirm, that **predicting future tactile states** is more informative than merely perceiving current ones. ViTacFormer therefore

---

Corresponding to Haoran Geng <ghr@berkeley.edu>

adds a dedicated tactile-prediction head that forces the shared latent space to encode actionable touch dynamics.

Experiments show that learning representations from predicted tactile signals in an autoregressive manner is challenging. To address this, we propose a two-phase curriculum: during the first 75% of training, we use ground-truth tactile inputs to stabilize the representation learning; in the final 25%, we transition to predicted tactile signals, promoting robust cross-modal reasoning.

To evaluate ViTacFormer, we construct the first comprehensive real-world benchmark for visuo-tactile dexterous manipulation, spanning both short- and long-horizon tasks. Across all benchmarks, ViTacFormer improves the success rate by roughly **50%** over strong baselines and is, to our knowledge, the first system to successfully complete very long-horizon dexterous manipulation tasks on a real robot, achieving **11 sequential stages** and sustaining continuous manipulation for over **2.5 minutes**.

In summary, our contributions include:

- A real-world experimental setup featuring bi-manual dexterous robotic hands, a teleoperation system, a high-quality dataset for real-world dexterous manipulation, and a comprehensive benchmark suite for evaluating visuo-tactile manipulation performance.
- A novel multimodal representation learning framework that integrates cross-attention for effective modality fusion, employs autoregressive modeling to forecast tactile signals, and introduces a tailored curriculum to enhance policy learning and generalization.
- We demonstrate strong results showcasing versatile and dexterous manipulation capabilities, including success in complex, long-horizon tasks. Our method outperforms strong baselines by approximately 50% in success rate and, to our knowledge, is the first to achieve very long-horizon dexterous manipulation on a real robot, completing 11 sequential stages with 2.5 minutes of continuous operation.

## 2 Related Work

### 2.1 Dexterous Manipulation

Dexterous manipulation has emerged as a critical research frontier, with applications in tasks like grasping [32, 44, 31], in-hand manipulation [39, 13, 23, 34, 4], in-hand orientation [3], articulated object manipulation [1, 16, 5, 43, 11, 9, 10], and deformable object manipulation [20, 35, 48, 33]. Meanwhile, behavior cloning (BC) [28, 27, 30] empowers dexterous manipulation with an end-to-end general solution.

Among BC models, diffusion policy (DP) [6] leverages diffusion models [29, 15] to learn the expert actions conditioned on robot observations. Since diffusion models [29, 15] are good at capturing the multi-modalities from diverse data inputs, diffusion policy shows promising results on robotic applications [47]. 3D diffusion policy (DP3) [42] utilizes 3D point clouds as robot observations. It is more generalizable compared to DP since the learned representation captures geometric information from 3D data. Action chunking transformer (ACT) [46] views BC model as a conditional variational auto-encoder. It learns the multi-modal information from diverse expert data inputs. Empirical studies [47] show that ACT [46] outperforms DP [6] when the collected data is limited.

Our ViTacFormer is built on top of ACT [46], leveraging the advantage of capturing multi-modalities in diverse expert data inputs. It offers a cross-modal representation learning for visuo-tactile dexterous manipulation. The learned representation enables precise and adaptive manipulation on multifingered dexterous hands.

### 2.2 Manipulation with Tactile Signals

Prior works with tactile signals focus mainly on learning tactile representations for robotics. Some of these works focus on leveraging force values as tactile signals. For example, [19] builds some proxy tasks such as predicting robot optical flow to extract tactile representations. HATO [21] sets up a bimanual dexterous visuo-tactile manipulation system with a diffusion policy [6] to learn the expert behaviors. Other works focus on leveraging self-supervised learning to extract rich representations specifically from high-resolution tactile images [38, 37, 40, 14, 8]. Among these works, contrastive

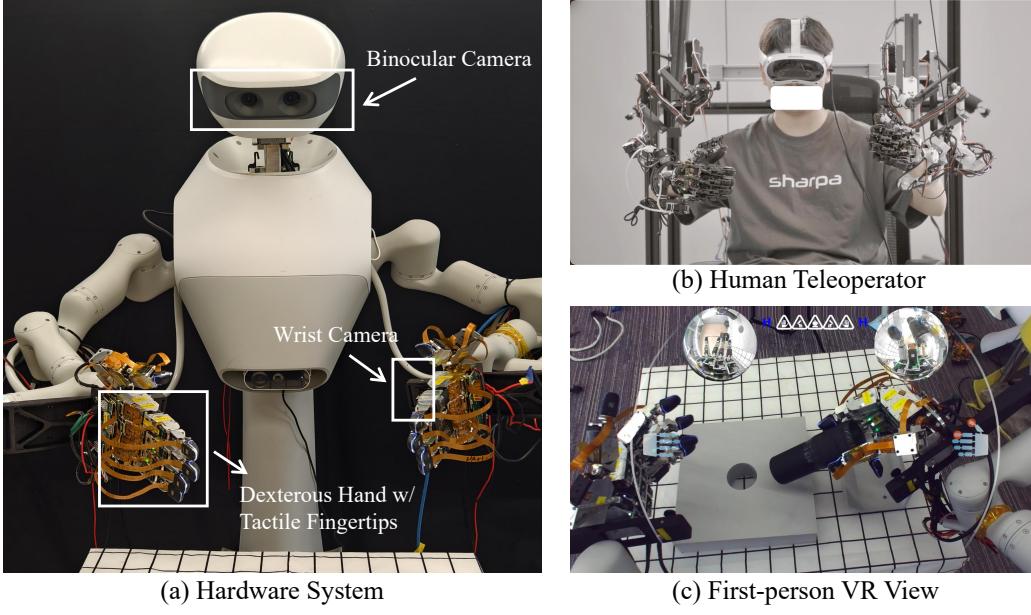


Figure 1: An overview of our system hardware and teleoperation setup. (a) Our hardware system setup. The hardware consists of two Realman robot arms, each equipped with a SharpaWave dexterous hand. Visual observations are obtained via two wrist cameras and one binocular camera. (b) Teleoperator with exoskeleton gloves and VR headset. (c) VR interface with binocular and wrist views, and tactile feedback overlay.

learning [38] and masked auto-encoding [26] are two popular streams to extract the representation from raw tactile images.

However, there is still a lack of an effective model that learns cross-modal representations for visuo-tactile dexterous manipulation [24, 41]. Our ViTacFormer proposes a cross-attention-based auto-regressive model for future tactile forecasting and action generation. Empirical studies show that ViTacFormer unlocks the power of visuo-tactile representation for dexterous manipulation. In particular, ViTacFormer is capable of mastering long-horizon dexterous robotic tasks.

### 3 Problem Formulation and Hardware Setup

#### 3.1 Problem Formulation

We address imitation learning for dexterous bi-manual manipulation. Given  $N$  expert trajectories  $\mathcal{D} = \{\tau_i\}_{i=1}^N$ , where each  $\tau_i = \{(o_t^i, a_t^i)\}_{t=1}^{T_i}$  consists of multimodal observations  $o_t^i$ , and corresponding actions  $a_t^i$ . Here, the multimodal observations  $o_t^i$  include robot proprioception  $j_t^i$ , visual observations,  $v_t^i$  and tactile observations  $h_t^i$  from tactile fingertips.

The goal is to learn a policy  $\pi_\theta$  that maps observations to actions:  $a_t = \pi_\theta(o_t)$ . The policy  $\pi_\theta$  is trained to imitate expert behavior and is evaluated in task space, measuring success on manipulation tasks under diverse and long-horizon conditions.

#### 3.2 Hardware Setup

Our hardware system consists of two Realman robot arms, each equipped with a SharpaWave dexterous hand (Fig.1(a)). Each hand is anthropomorphic, featuring 5 digits with 17 degrees of freedom (DoFs). Note that it is the developing version of SharpaWave. Visual observations are captured using two wrist-mounted fisheye cameras for close-up task views and a top-mounted ZED Mini stereo camera for global scene awareness. Tactile sensing is enabled by high-resolution (320×240) tactile sensors embedded in the fingertips, developed by Sharpa.

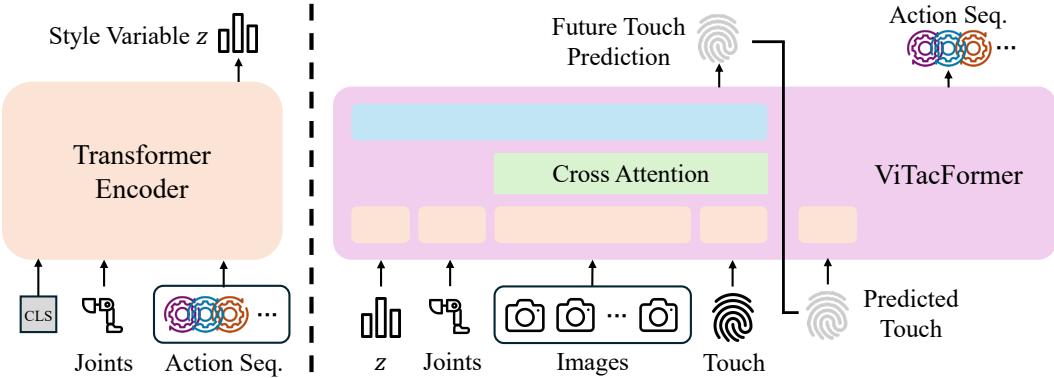


Figure 2: The neural network architecture for ViTacFormer is a conditional variational auto-encoder. Left: a transformer-based encoder maps action sequence and robot proprioception to action style variable  $z$ . Right: a transformer-based encoder-decoder uses style variable  $z$ , robot proprioception (joints), and visuo-tactile observations to auto-regressively predict future tactile signals and generate actions.

We adopt a custom exoskeleton-based teleoperation system to collect high-quality visuo-tactile demonstrations (Fig.1(b)). The operator wears a pair of mechanical exoskeleton gloves that are mechanically coupled to the SharpaWave hands, faithfully capturing finger joint motions. A VR headset provides immersive visual feedback through a first-person interface that integrates multimodal sensory input (Fig.1(c)). The interface combines (i) a stereo top-down view from the ZED Mini, (ii) wrist-mounted local views from both arms, and (iii) real-time tactile overlays on the fingertips that highlight contact activations. This unified perception setup enables the operator to intuitively control both hands in complex, contact-rich tasks. All data streams — including RGB frames, joint states, and compressed tactile maps — are time-synchronized and logged to construct multimodal expert trajectories.

## 4 Method

In section 4.1, we introduce a cross-attention-based multimodal integration framework that fuses the visual and tactile observation inputs. In section 4.2, we present autoregressive modeling with tactile forecasting, which better generates actions with predicted future tactile signals. In section 4.3, we summarize the network architecture and learning procedure for our ViTacFormer.

### 4.1 Cross-Attention-Based Multimodal Integration

Visual observations and tactile signals share similar semantic information. Traditional neural network architecture fuses visual and tactile observation inputs as naive token fusion. These models don't take the relevant information between visual and tactile observations into consideration.

Cross-attention is a mechanism commonly used in transformers, particularly in tasks involving multi-modal data or interacting with external knowledge. It allows the model to attend to different parts of two input sequences simultaneously, enabling it to capture interactions between them. Consequently, cross-attention-based multimodal integration motivates the agent to capture dependencies between diverse data inputs.

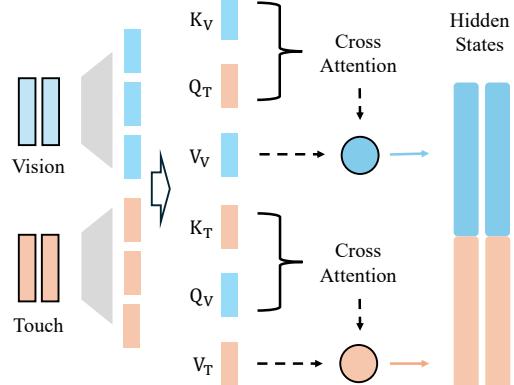


Figure 3: Cross-attention-based multimodal integration between visual and tactile observations.

In ViTacFormer, we apply cross attention to visual and tactile observations for learning better representations. This motivates the extraction of the relevant semantic information between visual and tactile signals. Fig. 3 shows the neural network architecture of multimodal integration based on cross-attention. The keys and values from visual observations are calculated with the queries from tactile signals and vice versa. Finally, the cross-attention-based features are concatenated into hidden states for further learning.

## 4.2 Auto-Regressive Modeling with Tactile Signal Forecasting

Forecasting future tactile signals motivates the agent to be aware of the change in contact signals. In detail, it motivates the latent representations involving potential future outcomes. Auto-regressively leveraging these predicted future tactile signals motivates the agent to use this prior contact knowledge for better generating actions.

To take advantages above, we formulate the action-generating procedure in two steps. First, we predict the future tactile tokens with style variables  $z$ , current robot proprioception (joints), and visuo-tactile observations. Next, we concatenate the predicted future tactile signals with the current input tokens for generating actions. Note that we conduct cross-attention-based multimodal integration twice between visuo-tactile signals, both in predicting future tactile signals and generating actions.

In practice, the representation would be broken since the cross-attention integration between visual observations and predicted noisy tactile signals [2]. To better maintain the representation model, we propose a two-phase curriculum. First, we train the action generation procedure with ground truth future tactile tokens at the beginning 75% training epochs. Finally, at the final 25% training epochs, we train the action generation procedure with predicted future tactile signals. Similar findings are investigated in [2].

## 4.3 Neural Network Architecture and Learning Procedure

Fig. 2 shows the neural network architecture of our ViTacFormer. This architecture is basically a conditional variational auto-encoder. On the left of Fig. 2, there is a transformer-based encoder. It maps the robot’s proprioception (joints) and expert action sequence into a style variable  $z$ . On the right of Fig. 2, there is a transformer-based encoder-decoder. First, it extracts the representation from visual and tactile observations with a cross-attention-based multimodal integration framework. Next, it auto-regressively predicts the future tactile signals and thus generates the actions with predicted future tactile signals. The style variable  $z$  is sampled from expert demonstrations during training, while it is sampled from Gaussian noise during inference.

We find that a combination of supervision between arm end-effectors’ position and arm/hand joint angles (JA) is more effective for dexterous manipulation than arm/hand JA supervision only. The training loss of ViTacFormer is shown as:

$$\mathcal{L} = w_1 \cdot \mathcal{L}_{KL} + w_2 \cdot \mathcal{L}_{JA} + w_3 \cdot \mathcal{L}_{tactile} + w_4 \cdot \mathcal{L}_{arm}, \quad (1)$$

where  $w_{1,2,3,4}$  are hyper-parameters,  $\mathcal{L}_{KL}$  is KL divergence between the action style variables and Gaussian distribution,  $\mathcal{L}_{JA}$  is the  $L_1$  loss based on predicted action and ground truth action,  $\mathcal{L}_{tactile}$  is the  $L_1$  loss based on future tactile signals and ground truth. In particular,  $\mathcal{L}_{arm}$  is:

$$\mathcal{L}_{arm} = \lambda_1 \cdot \mathcal{L}_{position} + \lambda_2 \cdot \mathcal{L}_{rotation}, \quad (2)$$

where  $\lambda_{1,2}$  are hyper-parameters,  $\mathcal{L}_{arm}$  indicates the supervision based on arm end-effectors,  $\mathcal{L}_{position}$  is the  $L_2$  loss between arm end-effector’s position, and  $\mathcal{L}_{rotation}$  is the  $L_1$  loss between arm end-effector’s rotation. Empirically, we find  $\mathcal{L}_{arm}$  is very useful in training dexterous manipulation skills.

At the beginning of the training procedure, the predicted future tactile signals are noisy. It’s hard for the whole training pipeline to work. Consequently, we propose a two-phase curriculum. First, we apply ground truth future tactile signals when training action generation at the beginning. Finally, we apply predicted future tactile signals for action generation at the final 1/4 training epochs. Similar findings are investigated in [2].

## 5 Experiment

In this section, we evaluate the effectiveness of our proposed ViTacFormer. The experiments are designed to answer two questions: (1) How does our algorithm perform compared to other state-of-the-art imitation learning algorithms? The results are presented in section 5.3. (2) Is each component of our algorithm effective? The results are introduced in section 5.4.

### 5.1 Benchmark and Environment Setup

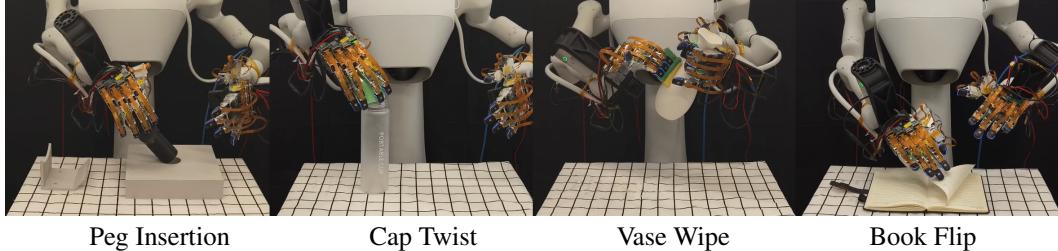


Figure 4: Four short-horizon visuo-tactile tasks, from left to right, i.e., peg insertion, cap twist, vase wipe, and book flip.

In this section, we introduce the tasks and environment setup. The tasks we conduct include 4 simple dexterous manipulation tasks and a very long-horizon visuo-tactile task. Fig. 4 shows the 4 simple dexterous manipulation tasks involving peg insertion, cap twist, vase wipe, and book flip. We also conduct our ViTacFormer on a very long-horizon task, i.e., making hamburgers.

We conduct algorithm comparison on all tasks and ablation study on four simple tasks. These tasks range from easy to complex dexterous manipulation. The results show that our ViTacFormer outperforms other state-of-the-art imitation learning algorithms by over 50% success rates. The ablation study illustrates that each component in our ViTacFormer improves the manipulation performance. Note that we use only 50 trajectories in each task for experiments. This makes the tasks more challenging than before.

### 5.2 Metrics and Baselines

**Metrics** We evaluate the algorithms with two established metrics: human normalized score and success rates. Note that the success rates may not reflect the dexterous manipulation process in detail, especially for long-horizon manipulation tasks. We define a new metric to measure the dexterous manipulation performance.

We propose Human Normalized Score (HNS) to evaluate the manipulation process in detail. First, we split the manipulation process into several stages. In each stage, we evaluate the process with 0 – 3 raw scores. Finally, we normalize the score for a fair comparison. The HNS score is presented as:

$$HNS = \frac{\sum_{i=1}^N w_i \cdot s_i}{3 * \sum_{i=1}^N w_i}, \quad (3)$$

which  $N$  represents there are  $N$  stages for a certain manipulation task,  $w_i$  represents tactile reliance in stage  $i$ ,  $s_i$  counts from 0 to 3 which is the raw score for stage  $i$ .

**Baselines** Diffusion Policy (DP) [6] is good at mimicking expert behaviors by capturing multi-modalities from training data with a diffusion model [15]. Additionally, HATO [21] adds the tactile signals as conditions for diffusion policy [6]. ACT [46] builds a conditional variational auto-encoder for learning expert behaviors from demonstrations. ACTw/T [46] adds the tactile signal in its input tokens. Empirical studies [47] show that ACT [46] outperforms DP [6] with limited training data.

In this section, we use DP [6], HATO [21], ACT [46], ACTw/T [46] as our baselines. Among these baselines, DP and ACT are without tactile inputs. HATO and ACTw/T take tactile signals with a naive token fusion.

### 5.3 Algorithm Comparison

*Question 1: How does ViTacFormer perform compared to other SoTA imitation learning algorithms?*

To test the effectiveness of our ViTacFormer, we conduct experiments on four short-horizon dexterous manipulation tasks. We test each algorithm on a certain task with inference 10 times. If the results show that ViTacFormer achieves higher success rates compared to SoTA imitation learning baselines, we could prove the efficacy of our ViTacFormer.

Table 1: Success rate comparison on four short-horizon dexterous manipulation tasks. Our ViTacFormer achieves over 50% success rates compared to the baselines.

Task	Peg Insertion	Cap Twist	Vase Wipe	Book Flip
DP [6]	2/10	0/10	3/10	1/10
ACT [46]	4/10	4/10	3/10	2/10
HATO [21]	4/10	1/10	4/10	3/10
ACTw/T [46]	6/10	6/10	4/10	4/10
<b>Ours</b>	<b>10/10</b>	<b>10/10</b>	<b>9/10</b>	<b>9/10</b>

Tab. 1 shows the success rates on four short-horizon dexterous manipulation tasks. Our ViTacFormer achieves the best performance compared to other SoTA imitation learning algorithms. In particular, ViTacFormer outperforms other baselines over 50% success rates, therefore almost solving these tasks. On the other hand, tactile observation input greatly improves the manipulation performance since ACTw/T [46] and HATO [21] outperform ACT [46] and DP [6], respectively.

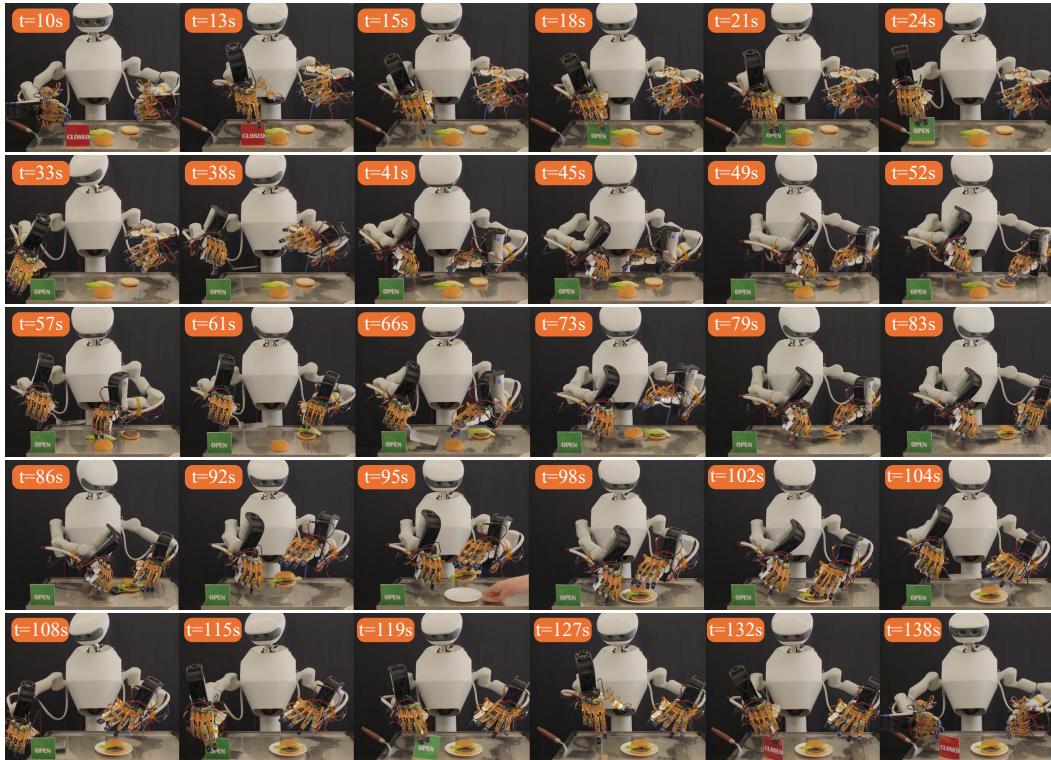


Figure 5: Successful model rollout on long-horizon task, i.e., making hamburger. We show the successful model rollout with keyframes in 11 stages. The first row represents the robot hand turning the brand to "open". The second row represents the robot hand shoveling meat to bread. The third row represents the robot hand assembling the hamburger. The fourth row represents the robot hand handing over the hamburger to the plate. The fifth row represents the robot hand turning the brand to "close".

*Question 2: How does ViTacFormer perform on complex long-horizon manipulation tasks?*

To show that our ViTacFormer is effective on long-horizon tasks, we conduct experiments on an 11-stage task, i.e., making hamburgers. To the best of our knowledge, ViTacFormer is the first system to complete very long-horizon dexterous manipulation tasks on a real robot. Fig. 5 shows a successful model rollout of our ViTacFormer. Our ViTacFormer masters 11 stages of making hamburgers.

Table 2: Human evaluation score comparison on a very long-horizon dexterous manipulation task. ViTacFormer shows promising results on this long-horizon task.

Stage	1	2	3	4	5	6	7	8	9	10	11	Overall
w/o Touch [46]	2.4	2.5	1.9	2	0.7	2.2	1.6	2.8	2.2	2.2	0.7	0.61
<b>Ours</b>	<b>2.9</b>	<b>3</b>	<b>1.9</b>	1.8	<b>2.7</b>	<b>2.9</b>	<b>2</b>	<b>2.8</b>	<b>2.4</b>	<b>2.5</b>	<b>3</b>	<b>0.88</b>

Tab. 2 shows the human normalized score (HNS) for each stage in the task, i.e., making hamburgers. Note that once the score is less than 1 for one stage, the model fails on this task. In experiments, we correct the mistake for further stage testing. Specifically, stage 1 corresponds to the first row of Fig. 5; Stage 2-4 corresponds to the second row; Stage 5-7 corresponds to the third row; Stage 8-10 corresponds to the fourth row; and stage 11 corresponds to the fifth row. For the baseline (w/o Touch), the model easily fails at stage 5 and stage 11. It leads the baseline to almost 0 success rates on this long-horizon task. In contrast, our ViTacFormer shows promising results on all stages and achieve **over 80% success rates** on this task.

#### 5.4 Ablation Study

*Question 3: How does each component in our ViTacFormer contribute to the baseline?*

Tactile information enhances the dexterous manipulation process with force feedback. It can improve the robustness and stability in the dexterous manipulation process. To evaluate the effectiveness of each component in our ViTacFormer, we conduct a series of ablation studies based on the naive implemented baseline (w/o Touch). If each component of our ViTacFormer improves the manipulation performance, we could prove the effectiveness of our ViTacFormer.

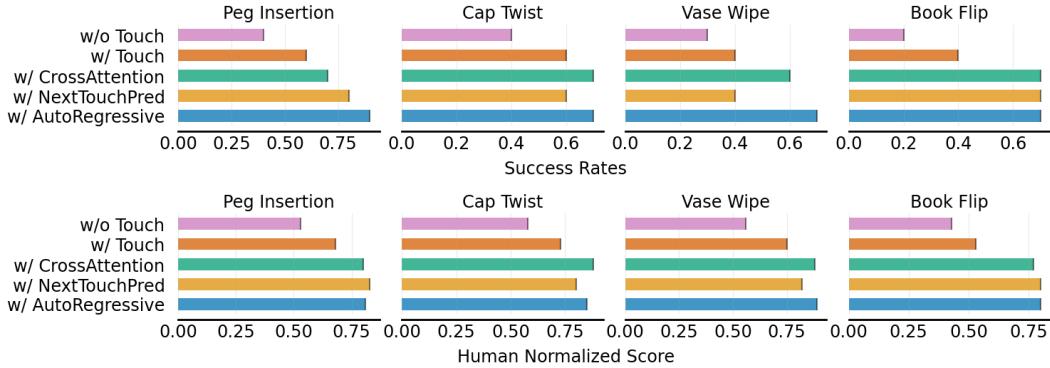


Figure 6: Ablation Study. The comparison between several ablated algorithms. Each component of our ViTacFormer shows its effectiveness.

Fig. 6 shows the success rates and human normalized score for each ablated algorithm. Each component in our ViTacFormer improves the performance based on the baseline without touch inputs (w/o Touch) and with naive touch token fusion (w/ Touch). In detail, cross attention between visual observations and tactile observations (w/ CrossAttention) improves the knowledge fusion process of vision and touch. Therefore, it greatly improves the imitation performance. Predicting the next tactile tokens (w/ NextTouchPred) captures the temporal difference between tactile observations. It improves the manipulation stability. Auto-regressively leveraging the predicted future tactile tokens as inputs (w/ AutoRegressive) simplifies the process of reasoning action tokens. It also improves the performance compared to the baselines.

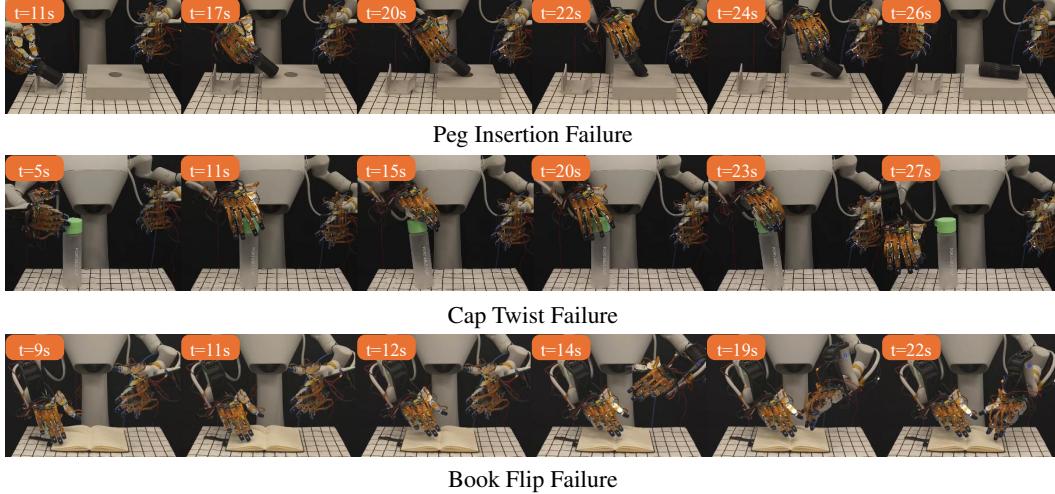


Figure 7: Failure Study. The first row is peg insertion failure, the second row is cap twist failure, and the third row is book flip failure.

#### *Question 4: How does the failure occur in baselines?*

There are several failure modes from the baselines. Evaluating these failure cases helps us understand the effective factors in our ViTacFormer. Fig. 7 shows the classical failure cases from baselines (w/o Touch and w/ Touch). The established tasks are highly tactile-dependent. Consequently, how to leverage the tactile signals in imitation learning is of great significance.

Fig. 7 shows the peg insertion failure. When the peg is moved to the hole, the robot hand isn't aware of the position of the hole and thus fails to insert the peg into the hole. This failure shows the importance of predicting future tactile signals in ViTacFormer. Predicting the future tactile tokens motivates the dexterous hand to be aware of the temporal force difference. When the peg is near the hole, the temporal force difference would be changed, therefore showing the hole is nearby. Consequently, it could improve the robustness of inserting the peg into the hole in ViTacFormer.

Fig. 7 shows the cap twist failure. The robot hand fails to twist the cap. The reason can be traced to the fact that the hand isn't aware of whether the cap is open or closed. It motivates the importance of the autoregressive architecture of our ViTacFormer. Auto-regressively predicting the future tactile signals and leveraging these signals simplifies reasoning about the actions under such complex situations.

Fig. 7 shows the book flip failure. The dexterous hand isn't aware of the book. It flips the book in the air. This originates from the lack of visual and tactile observation fusion. In our ViTacFormer, we use cross-attention-based multimodal integration to fuse the visual and tactile observations. This improves the performance of dexterous manipulation.

## 6 Limitation

Due to the inherent constraints of imitation learning, our policy lacks the capability to autonomously generalize to novel tasks. It still depends on human teleoperation for data collection. While our method demonstrates strong performance in long-horizon and challenging visuo-tactile manipulation tasks, its stability can be affected in scenarios where tactile feedback is less critical. This is primarily due to sensor noise and the limitations of the representation learning process.

## 7 Conclusion

We present ViTacFormer, a unified visuo-tactile framework for dexterous robotic manipulation that leverages deep cross-modal representation learning. By fusing vision and touch at every stage of the policy and incorporating predictive tactile modeling, ViTacFormer enables robust, fine-grained control across a diverse set of manipulation tasks. Our curriculum-based training strategy further enhances

representation stability, allowing the system to effectively reason over predicted tactile signals. Empirical results demonstrate that ViTacFormer significantly outperforms strong baselines—achieving approximately 50% higher success rates—and is the first to complete long-horizon dexterous tasks on a real robot. We believe this work opens new possibilities for generalizable, high-precision robotic manipulation through the principled integration of vision and touch.

## References

- [1] Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21190–21200, June 2023.
- [2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al.  $\pi_0$ : A vision-language-action flow model for general robot control, 2024. <https://arxiv.org/abs/2410.24164>.
- [3] Tao Chen, Jie Xu, and Pukit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pages 297–307. PMLR, 2022.
- [4] Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2804–2818, 2024. doi: 10.1109/TPAMI.2023.3339515.
- [5] Yuanpei Chen, Chen Wang, Yaodong Yang, and C. Karen Liu. Object-centric dexterous manipulation from human motion data, 2024. URL <https://arxiv.org/abs/2411.04005>.
- [6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.
- [7] Yufei Ding, Haoran Geng, Chaoyi Xu, Xiaomeng Fang, Jiazhao Zhang, Songlin Wei, Qiyu Dai, Zhizheng Zhang, and He Wang. Open6dor: Benchmarking open-instruction 6-dof object rearrangement and a vlm-based approach. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7359–7366, 2024. doi: 10.1109/IROS58592.2024.10802733.
- [8] Letian Fu, Gaurav Datta, Huang Huang, William Chung-Ho Panitch, Jaimyn Drake, Joseph Ortiz, Mustafa Mukadam, Mike Lambeta, Roberto Calandra, and Ken Goldberg. A touch, vision, and language dataset for multimodal alignment. *arXiv preprint arXiv:2402.13232*, 2024.
- [9] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. *arXiv preprint arXiv:2211.05272*, 2022.
- [10] Haoran Geng, Ziming Li, Yiran Geng, Jiayi Chen, Hao Dong, and He Wang. Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations. *arXiv preprint arXiv:2303.16958*, 2023.
- [11] Haoran Geng, Songlin Wei, Congyue Deng, Bokui Shen, He Wang, and Leonidas Guibas. Sage: Bridging semantic and actionable parts for generalizable articulated-object manipulation under language instructions, 2023.
- [12] Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, Yutong Liang, Dylan Goetting, Chaoyi Xu, Haozhe Chen, Yuxi Qian, Yiran Geng, Jiageng Mao, Weikang Wan, Mingtong Zhang, Jiangran Lyu, Siheng Zhao, Jiazhao Zhang, Jialiang Zhang, Chengyang Zhao, Haoran Lu, Yufei Ding, Ran Gong, Yuran Wang, Yuxuan Kuang, Ruihai Wu, Baoxiong Jia, Carlo Sferrazza, Hao Dong, Siyuan Huang, Yue Wang, Jitendra Malik, and Pieter Abbeel. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning, 2025. URL <https://arxiv.org/abs/2504.18904>.

- [13] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, Yashraj Narang, Jean-Francois Lafleche, Dieter Fox, and Gavriel State. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. *arXiv*, 2022.
- [14] Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, et al. Sparsh: Self-supervised touch representations for vision-based tactile sensing. *arXiv preprint arXiv:2410.24090*, 2024.
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [16] Taoran Jiang, Liqian Ma, Yixuan Guan, Jiaoqiao Meng, Weihang Chen, Zecui Zeng, Lusong Li, Dan Wu, Jing Xu, and Rui Chen. Dexsim2real<sup>2</sup>: Building explicit world model for precise articulated object dexterous manipulation, 2024. URL <https://arxiv.org/abs/2409.08750>.
- [17] Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. *arXiv preprint arXiv:2407.04689*, 2024.
- [18] Yuxuan Kuang, Haoran Geng, Amine Elhafsi, Tan-Dzung Do, Pieter Abbeel, Jitendra Malik, Marco Pavone, and Yue Wang. Skillblender: Towards versatile humanoid whole-body locomotion via skill blending. *arXiv preprint arXiv:2506.09366*, 2025.
- [19] Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020.
- [20] Sizhe Li, Zhaio Huang, Tao Chen, Tao Du, Hao Su, Joshua B. Tenenbaum, and Chuang Gan. Dexdeform: Dexterous deformable object manipulation with human demonstrations and differentiable physics, 2023. URL <https://arxiv.org/abs/2304.03223>.
- [21] Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. *IEEE International Conference on Robotics & Automation (ICRA)*, 2025.
- [22] Rundong Luo\*, Haoran Geng\*, Congyue Deng, Puhalo Li, Zan Wang, Baoxiong Jia, Leonidas Guidbas, and Siyuan Huang. Physpart: Physically plausible part completion for interactable objects. *International Conference on Robotics and Automation (ICRA)*, 2025. URL <https://arxiv.org/abs/2408.13724>.
- [23] 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.
- [24] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023.
- [25] Younggyo Seo, Carmelo Sferrazza, Haoran Geng, Michal Nauman, Zhao-Heng Yin, and Pieter Abbeel. Fasttd3: Simple, fast, and capable reinforcement learning for humanoid control, 2025. URL <https://arxiv.org/abs/2505.22642>.
- [26] Carmelo Sferrazza, Younggyo Seo, Hao Liu, Youngwoon Lee, and Pieter Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9698–9705. IEEE, 2024.
- [27] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliprot: What and where pathways for robotic manipulation. In *Conference on robot learning (CoRL)*, pages 894–906. PMLR, 2022.

- [28] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning (CoRL)*, pages 785–799. PMLR, 2023.
- [29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [30] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy, 2023.
- [31] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. *arXiv preprint arXiv:2304.00464*, 2023.
- [32] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzheng Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. *arXiv preprint arXiv:2210.02697*, 2022.
- [33] Yuran Wang, Ruihai Wu, Yue Chen, Jiarui Wang, Jiaqi Liang, Ziyu Zhu, Haoran Geng, Jitendra Malik, Pieter Abbeel, and Hao Dong. Dexgarmentlab: Dexterous garment manipulation environment with generalizable policy, 2025. URL <https://arxiv.org/abs/2505.11032>.
- [34] 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. URL <https://arxiv.org/abs/2409.17549>.
- [35] Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. Learning to manipulate deformable objects without demonstrations, 2020. URL <https://arxiv.org/abs/1910.13439>.
- [36] Yinzheng Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. *arXiv preprint arXiv:2303.00938*, 2023.
- [37] 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.
- [38] Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26340–26353, 2024.
- [39] 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.
- [40] Kelin Yu, Yunhai Han, Qixian Wang, Vaibhav Saxena, Danfei Xu, and Ye Zhao. Mimictouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. *arXiv preprint arXiv:2310.16917*, 2023.
- [41] Ying Yuan, Haichuan Che, Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Kang-Won Lee, Yi Wu, Soo-Chul Lim, and Xiaolong Wang. Robot synesthesia: In-hand manipulation with visuotactile sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6558–6565. IEEE, 2024.
- [42] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.
- [43] Hui Zhang, Sammy Christen, Zicong Fan, Luocheng Zheng, Jemin Hwangbo, Jie Song, and Otmar Hilliges. Artigrasp: Physically plausible synthesis of bi-manual dexterous grasping and articulation. In *2024 International Conference on 3D Vision (3DV)*, pages 235–246, 2024. doi: 10.1109/3DV62453.2024.00016.

- [44] Jialiang Zhang, Haoran Liu, Danshi Li, XinQiang Yu, Haoran Geng, Yufei Ding, Jiayi Chen, and He Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *8th Annual Conference on Robot Learning*.
- [45] Jialiang Zhang, Haoran Geng, Yang You, Congyue Deng, Pieter Abbeel, Jitendra Malik, and Leonidas Guibas. Rodrigues network for learning robot actions, 2025. URL <https://arxiv.org/abs/2506.02618>.
- [46] 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.
- [47] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.
- [48] Sun Zhaole, Jihong Zhu, and Robert B. Fisher. Dexdlo: Learning goal-conditioned dexterous policy for dynamic manipulation of deformable linear objects. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16009–16015, 2024. doi: 10.1109/ICRA57147.2024.10610754.

**Appendix A.** Implementation and training details, including sensor modalities, action formats, and inference settings.

**Appendix B.** Expanded experimental results on all tasks, including task description, scoring schemes, inference results and failure analysis.

## A Additional Method Details

### A.1 Input Modalities

Our model takes multimodal inputs from the robot system, including visual observations, robot proprioception, and tactile signals.

#### Visual Input

We use four synchronized camera views as visual input: a stereo pair ( $180 \times 320$ ) from top-mounted ZED Mini cameras (Fig. 8(a), (c)), and two fisheye wrist-mounted views ( $256 \times 280$ ) for left and right hands (Fig. 8(b), (d)). All frames are encoded into image tokens via a vision backbone before cross-modal integration.

#### Proprioception Input

The robot’s internal state at each timestep is represented by a 58-dimensional vector, consisting of: 7-DoF left arm state, 17-DoF left hand state, 7-DoF right arm state, 17-DoF right hand state, and 2-DoF neck state—structured as [7, 17, 7, 17, 2]. A temporal horizon of 6 frames is used, resulting in a proprioceptive input of shape (6, 50).

#### Tactile Input

Each of the 10 fingertips is equipped with force and torque sensors along 3 axes, resulting in 20 tactile channels. For each channel, we collect 18 frames of data ([18, 3]), which are concatenated into a raw tactile tensor of shape [18, 60]. We additionally compute frame-wise deltas to obtain relative changes ([18, 60]), and concatenate them with the raw signal to produce the final tactile input of shape [18, 120].

### A.2 Action Output

The policy generates high-frequency action sequences with shape (100, 50) per rollout, where 50 corresponds to the full control dimension of the robot: 7-DoF left arm, 17-DoF left hand, 7-DoF right arm, 17-DoF right hand, and 2-DoF neck—matching the structure of the proprioceptive state. The 100-frame horizon supports fine-grained dexterous motion across extended manipulation stages.

### A.3 Data and training details

We train each task using 50 expert demonstrations and 100 epochs on 2 NVIDIA H20 GPUs. Short-horizon tasks typically converge within half a day, while long-horizon tasks (e.g., Make Hamburger) require up to 2 days. The model is optimized using the Adam optimizer with a learning rate of 1e-4 and a batch size of 128. Training supervision includes KL divergence on latent action style, L1 losses on both predicted actions and tactile signals, and auxiliary supervision on end-effector positions and rotations. All input modalities are temporally aligned and normalized prior to training.

### A.4 Inference Details

During deployment, the policy runs at 10Hz, producing a 100-frame (100, 50) high-frequency action sequence at each decision step. To ensure smooth and physically stable execution, we apply temporal

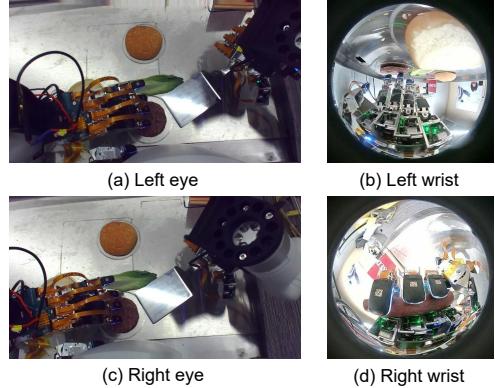


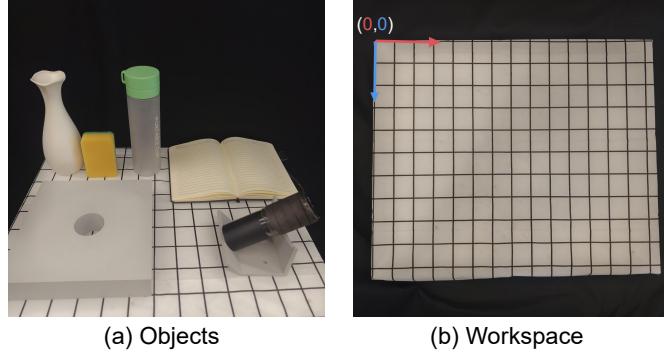
Figure 8: **Four types of camera views**

smoothing over the predicted action trajectory before sending commands to the robot. The system is deployed on a real dual-arm platform with synchronized visuo-tactile observation streams and low-latency control.

## B Additional Experiment Details

### B.1 Short-horizon tasks

The four short-horizon tasks share a standardized tabletop workspace and a common set of objects, as shown in Fig. 9(a). The workspace is discretized using a printed grid (5cm per square), with the top-left corner defined as the origin (0, 0), as illustrated in Fig. 9(b). During training, each object is placed at a designated grid coordinate. For generalization, we randomly perturb the object’s position within a circular region of half-grid radius (i.e., 2.5cm) around its original anchor point.



**Figure 9: Short-horizon task setup.** (a) All four short-horizon tasks share a common set of objects. (b) The tabletop workspace is marked with a grid; the top-left corner is defined as the origin (0, 0). Each object is positioned at a predefined grid point during training.

#### B.1.1 Peg Insertion

##### Task Description

The robot uses its right hand to grasp a cylindrical peg from the vertical rack, then moves it diagonally along the sloped platform toward the insertion hole. Upon reaching the vicinity of the hole, the robot is expected to insert the peg smoothly and stably into the hole. This task involves visual alignment, precise grasping, and tactile-guided insertion. Representative execution frames are shown in the first row of Fig. 10.

##### Scoring Scheme

Table 3: Scoring criteria for Peg Insertion.

Stage 1: Grasp (weight 1)	Description
0	No grasp
1	Grasped but slipped or dropped
2	Poor or tilted grasp
3	Stable grasp

Stage 2: Insertion (weight 2)	Description
0	No insertion
1	Misaligned, dropped
2	Partial insertion
3	Fully inserted

The task is divided into two stages: peg grasping (weight 1) and insertion (weight 2). Each stage is scored from 0 to 3 based on qualitative criteria such as grasp stability and insertion completeness. The human normalized score (HNS) is computed as a weighted average. A total score of 3 for stage 1 and  $\geq 2$  for stage 2 is considered successful.

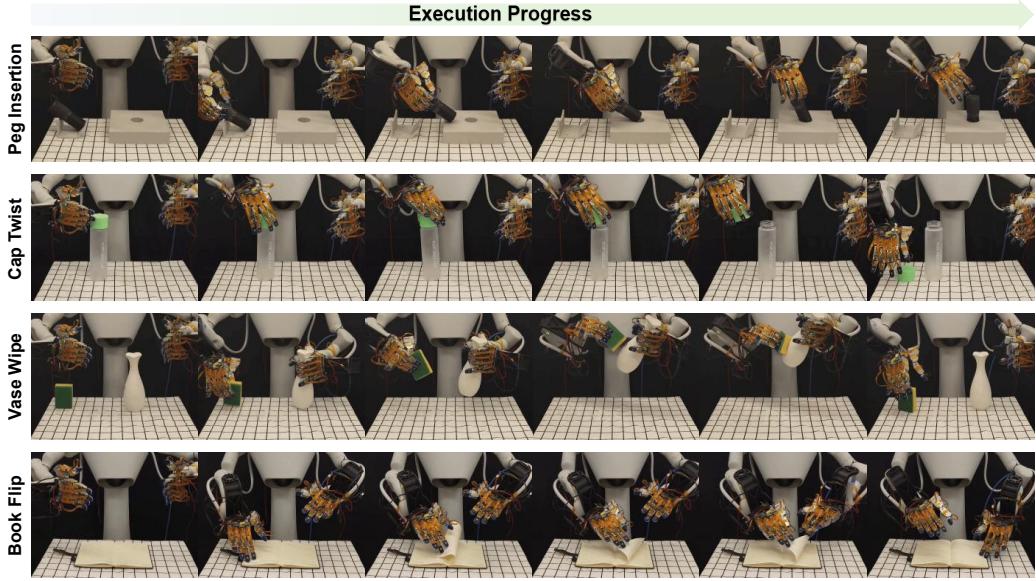


Figure 10: **Execution examples for short-horizon tasks.** Representative keyframes from four tasks: peg insertion, cap twist, vase wipe, and book flip. Each task demonstrates a full execution sequence from perception to manipulation.

## Inference Results

Table 4: **Peg Insertion: inference results across models.**

Model	Stage 1	Stage 2	HNS	Success Rate
DP	1.6	0.9	0.37	20%
ACT	2.6	1.1	0.53	40%
HATO	2.4	1.1	0.51	40%
ACT w/T	2.6	1.8	0.68	60%
ACT w/CrossAttention	3.0	2.1	0.80	70%
ACT w/NextTouchPred	2.7	2.4	0.83	80%
ACT w/AutoRegressive	2.9	2.2	0.81	90%
Ours	3.0	2.7	0.93	100%

Table 4 summarizes the quantitative performance on the peg insertion task. We report the average stage-wise scores, human normalized score (HNS), and success rate across baselines and ablations. Our method achieves the highest HNS (0.93) and 100% success rate, demonstrating strong performance across both stages.

## Failure Case Analysis

Figure 11, first row, shows two representative failure cases in the peg insertion task. In the first case, the robot fails to locate the insertion hole accurately and attempts to insert the peg at an incorrect position, leading to task failure despite a seemingly stable grasp. In the second case, the robot grasps the cylindrical peg with an imprecise hand posture, causing the thumb to slip during the transport phase. As a result, the peg deviates from the planned trajectory and misses the hole entirely.

### B.1.2 Cap Twist

#### Task Description

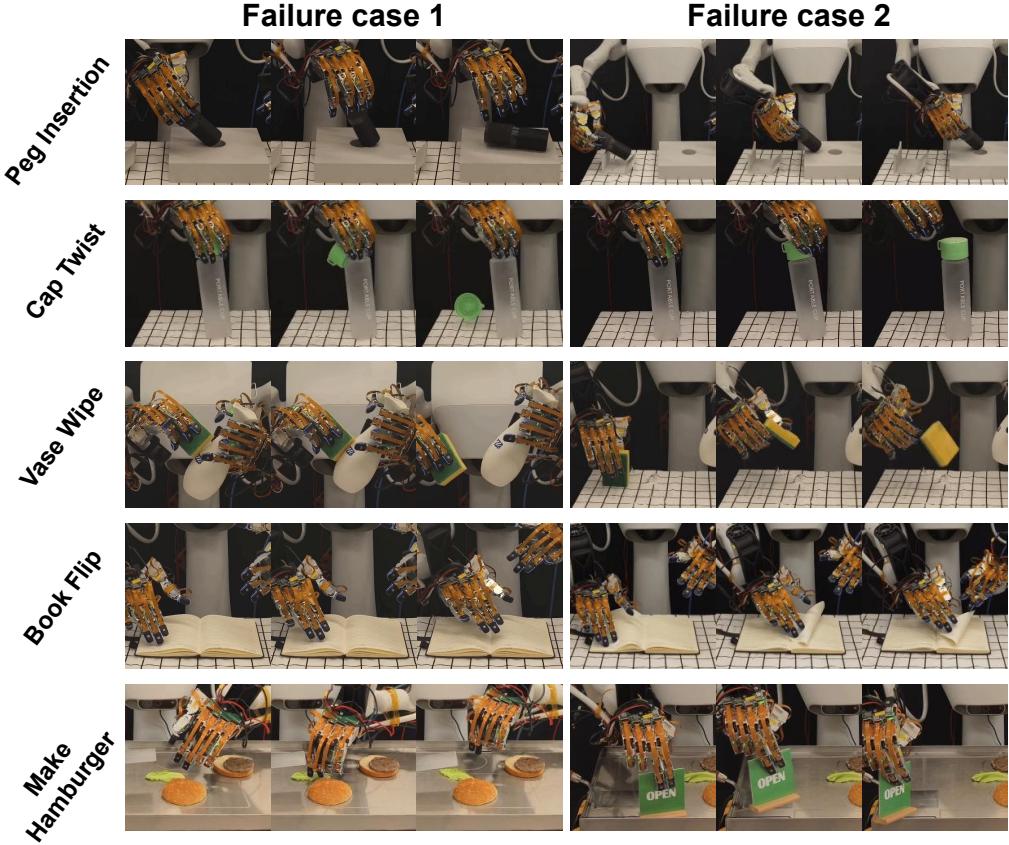


Figure 11: **Representative failure cases across all tasks.** Each row corresponds to one task, with two failure case sequences shown side by side.

The robot uses its right hand to rotate a cap off a bottle and place it on the table. The cap is initially tightened at a clockwise offset of about 100 degrees from the open position. Representative execution frames are shown in the second row of Fig. 10.

### Scoring Scheme

Table 5: Scoring criteria for Cap Twist.

Stage 1: Rotate (weight 2)	Description
0	No contact with the cap
1	Rotated 0–50°
2	Rotated 50–100°, or over-rotated
3	Fully unscrewed, cap held securely
Stage 2: Place (weight 2)	Description
0	Dropped immediately or stuck on bottle
1	Released before full separation
2	Partially placed or fell off
3	Stably placed on the table

The task is divided into two stages: rotation and placement. Each is scored from 0 to 3, and a task is considered successful if the cap is fully unscrewed and placed stably (stage 1 score 3, stage 2  $\geq 2$ ).

### Inference Results

Table 6: **Cap Twist: inference results across models.**

<b>Model</b>	<b>Stage 1</b>	<b>Stage 2</b>	<b>HNS</b>	<b>Success Rate</b>
DP	1.1	0.3	0.23	0%
ACT	2.4	1.1	0.58	40%
HATO	1.8	0.5	0.38	10%
ACT w/T	2.6	1.8	0.73	60%
ACT w/CrossAttention	3.0	2.3	0.88	70%
ACT w/NextTouchPred	2.8	2.0	0.80	60%
ACT w/AutoRegressive	2.9	2.2	0.85	70%
Ours	3.0	2.9	0.98	100%

Table 6 presents the model performance on the cap twist task. Our method achieves the best HNS score (0.98) and 100% success rate, highlighting the advantage of fine-grained tactile reasoning.

#### Failure Case Analysis

In the second row of Fig. 11, two failure cases from the cap twist task are shown. In the first case, the robot fails to detect that the cap has already loosened and continues to apply torque unnecessarily, resulting in over-rotation that destabilizes the object. In the second case, the fingers lose contact during the twisting motion, leading to slippage and an insufficient rotation angle, which prevents the cap from being successfully removed.

#### B.1.3 Vase Wipe

##### Task Description

The robot uses its left hand to pick up a vase and its right hand to grasp a sponge. It then wipes away the blue ink mark located at the center of the vase. Representative execution frames are shown in the third row of Fig. 10.

##### Scoring Scheme

Table 7: **Scoring criteria for Vase Wipe.**

<b>Stage 1: Pick (weight 1)</b>	<b>Description</b>
0	Failed to grasp the sponge
1	Grasped only a corner of sponge
2	Unstable grasp with partial control
3	Firm 3-finger grasp with full control
<b>Stage 2: Wipe (weight 2)</b>	<b>Description</b>
0	No contact with the ink mark
1	Wiped less than 50%
2	Wiped 50–90%, some ink remains
3	Fully wiped the ink area clean

The task is divided into two stages: sponge grasping (pick) and vase wiping (wipe), both scored from 0 to 3. If the operator intervenes to re-adjust the vase grasp during stage 1, the score is reduced by 1. The task is considered successful only if both stages score 3.

#### Inference Results

Table 8 shows the quantitative performance on the vase wiping task. Our method again achieves the best HNS (0.98) and 90% success rate, showing reliable grasping and contact-driven wiping.

##### Failure Case Analysis

The third row of Fig. 11 illustrates two typical failure modes in the vase wiping task. In the first case, the robot applies insufficient force during the wiping motion, resulting in incomplete surface contact between the sponge and the vase. Consequently, the ink mark is not fully removed. In the

Table 8: **Vase Wipe: inference results across models.**

Model	Stage 1	Stage 2	HNS	Success Rate
DP	1.8	1.3	0.49	30%
ACT	2.0	1.5	0.56	30%
HATO	2.5	1.7	0.65	40%
ACT w/T	3.0	1.9	0.75	40%
ACT w/CrossAttention	3.0	2.5	0.89	60%
ACT w/NextTouchPred	3.0	2.2	0.82	40%
ACT w/AutoRegressive	3.0	2.5	0.89	70%
Ours	3.0	2.9	0.98	90%

second case, excessive force is applied during the grasping phase, causing the sponge to slip out of the robot’s fingers before the wiping action begins.

#### B.1.4 Book Flip

##### Task Description

The robot uses its right-hand middle finger to flip up a single page and then presses the page down using its left hand. Representative execution frames are shown in the fourth row of Fig. 10.

##### Scoring Scheme

Table 9: **Scoring criteria for Book Flip.**

Stage 1: Flip (weight 2)	Description
0	No contact with the page
1	Touched but failed to lift / flipped multiple pages
2	Lifted halfway but stopped
3	Fully flipped one page
Stage 2: Press (weight 2)	Description
0	No contact with the page
1	Insufficient force, page rebounds
2	Pressed down, but misaligned
3	Fully and correctly pressed the page down

This task includes two stages: flipping and pressing. Each stage is scored from 0 to 3. The task is considered successful if stage 1 scores 3 and stage 2 scores  $\geq 2$ .

##### Inference Results

Table 10: **Book Flip: inference results across models.**

Model	Stage 1	Stage 2	HNS	Success Rate
DP	1.5	0.5	0.35	10%
ACT	1.9	0.7	0.43	20%
HATO	2.0	0.6	0.43	30%
ACT w/T	2.3	0.9	0.53	40%
ACT w/CrossAttention	2.7	1.9	0.77	70%
ACT w/NextTouchPred	2.9	1.9	0.80	70%
ACT w/AutoRegressive	2.7	2.1	0.80	70%
Ours	3.0	2.6	0.93	90%

Table 10 shows performance on the book flip task. Our method achieves the highest HNS (0.93) and 90% success rate, outperforming all ablations.

## Failure Case Analysis

Figure 11, fourth row, presents two failure modes in the book flip task. In the first case, the robot fails to perceive the presence or precise location of the page edge, resulting in a poking motion that completely misses the page during the flipping attempt. In the second case, the robot applies excessive downward force before initiating the flip, which presses the page flat against the book and prevents it from being lifted.

## B.2 Long-horizon task: Make Hamburger

### Workspace Setup

The long-horizon task is conducted on a customized metallic tabletop with seven designated ingredient/tool zones, as shown in Fig. 12. Each object is placed within either a circular or rectangular region marked on the tray. These regions serve as initialization zones with controlled spatial variability to support generalization. During both training and evaluation, each item is placed randomly within its assigned zone (up to 3cm positional jitter), ensuring that the policy must perform robust multimodal perception and execution.

### Task Description

The long-horizon task involves a full hamburger assembly sequence requiring precise tool use and multi-stage coordination. The robot begins by flipping a wooden card from “closed” to “open” to indicate the start of service. It then uses its right hand to grasp a spatula and sequentially completes the following steps: (1) lift and place the meat patty onto the bottom bread, (2) place a piece of lettuce, and (3) lift and place the top bread. Once the hamburger is assembled, the robot places it onto a plate handed over by a human. Finally, it returns the spatula to its original position and flips the sign back to “closed” to indicate task completion.

### Scoring Scheme

The long-horizon hamburger task is decomposed into 11 sequential stages, covering symbolic interaction (sign flipping), tool use (spatula manipulation), ingredient assembly (meat patty, lettuce, bun), and final delivery. Each stage is scored from 0 to 3, where 0 indicates failure or no attempt, 1–2 denote partial or unstable execution, and 3 represents correct and stable completion. To better reflect task complexity and tactile sensitivity, each stage is assigned a specific weight: for example, sign flipping and deformable object handling (lettuce, bun) are given higher weights due to their reliance on fine-grained control and multi-finger dexterity.

The weighted stage scores are used to compute a Human Normalized Score (HNS), which reflects the overall task performance. A stage is considered successful if the score is at least 1. The entire task is marked as successful only when all 11 stages meet this threshold. Table 11 details the scoring criteria and weights for each stage.

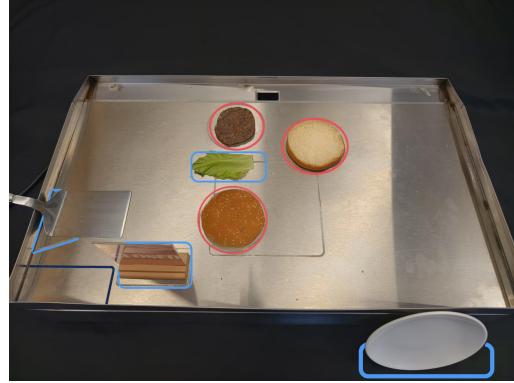


Figure 12: **Long-horizon task setup.** Seven components are placed in predefined zones—circular (ingredients) or rectangular (tools). Objects are randomly initialized within these areas to test spatial generalization.

Table 11: Scoring criteria for the long-horizon hamburger task.

<b>Stage</b>	<b>Action</b>	<b>Weight</b>	<b>Score Description</b>
1	Flip sign (start)	2	0: miss/fail; 1–2: partial (0–180°); 3: clean flip
2	Grab spatula	2	0: miss; 1–2: unstable grasp; 3: secure grasp
3	Lift meat patty	1	0: failed; 1–2: partial lift; 3: stable lift
4	Place meat patty	1	0: miss; 1–2: partial/inaccurate; 3: centered
5	Grasp lettuce	2	0: miss; 1–2: loose grasp; 3: stable placement
6	Lift top bread	2	0: failed; 1–2: unstable or too forceful; 3: correct
7	Place top bread	1	0: miss; 1–2: inaccurate; 3: clean stack
8	Lift hamburger	1	0: failed; 1–2: unstable; 3: correct lift
9	Place on plate	1	0: miss; 1–2: off-center; 3: perfect placement
10	Return spatula	1	0: drop/fail; 1–2: misaligned; 3: accurate return
11	Flip sign (end)	2	0: fail; 1–2: partial rotation; 3: clean close flip

### Failure Case Analysis

The fifth row of Fig. 11 shows two failure cases from the long-horizon hamburger assembly task. In the first case, the robot fails during stage 5 (grasping the lettuce): the grasp is unstable and incomplete, resulting in the lettuce slipping from the fingers before it can be placed. In the second case, the failure occurs in stage 1 (flipping the sign): although the sign is flipped, an incorrect grasp orientation causes the sign to rotate unintentionally during the movement, leading to a collision with the edge of the stove and blocking task progression.