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

Liang Heng[*1,2,3] **Haoran Geng**[*†1] **Kaifeng Zhang**[3]
Pieter Abbeel[1] **Jitendra Malik**[1]
[1]University of California, Berkeley [2]Peking University [3]Sharpa
[*]Equal Contribution † Project Lead [*]

Abstract: Dexterous manipulation is crucial for robots to interact with the physical world. While vision-based methods have advanced rapidly, tactile sensing remains essential for fine-grained control, especially under occlusion. We present ViTacFormer, a cross-modal framework that fuses vision and touch via cross-attention and predicts future tactile states with an autoregressive head. A curriculum gradually shifts from ground-truth to predicted tactile inputs, stabilizing representation learning. On real-world benchmarks covering both short- and long-horizon tasks, ViTacFormer improves success rates by about 50% over strong baselines, and is the first to complete 11-stage dexterous manipulation with 2.5 minutes of continuous operation.

Keywords: Dexterous Manipulation, Visuo-Tactile Fusion, Cross-Attention, Autoregressive Tactile Forecasting, Imitation Learning

1 Introduction

Recent years have witnessed rapid advances in robotic manipulation [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], with behavior cloning (BC) [11, 12, 13] emerging as a promising paradigm for high-precision real-world tasks. However, most existing approaches remain restricted to simple hand configurations [14] and generalize poorly—largely due to the underutilization of tactile sensing [15, 16, 17, 18], which is indispensable for fine-grained control.

While some studies have attempted to integrate tactile feedback into dexterous manipulation [19, 20], the learned tactile features are often shallow and underexplored. Self-supervised methods have also been applied to tactile data [21, 22, 23, 24, 25], but a unified cross-modal representation for visuo-tactile dexterous manipulation remains missing [26, 27].

We introduce **ViTacFormer**, a visuo-tactile framework that addresses this gap. Our key idea is to fuse high-resolution vision and tactile cues with cross-attention layers at every stage of the policy, and to enforce predictive modeling of future tactile states. This tactile-prediction head drives the latent space to encode actionable touch dynamics, providing richer cues than perceiving only current signals.

Since autoregressive tactile forecasting is inherently challenging, we design a **two-phase curriculum**: during the first 75% of training, ground-truth tactile signals stabilize learning; during the final 25%, predicted signals are used to promote robust cross-modal reasoning.

To evaluate ViTacFormer, we construct the first comprehensive real-world benchmark for visuo-tactile dexterous manipulation, covering four short-horizon tasks and a very long-horizon, 11-stage task. Across all benchmarks, ViTacFormer achieves about **50%** higher success rates than strong baselines, and is the first system to complete continuous dexterous operation for **2.5 minutes** over **11 sequential stages**.

[*]Corresponding to Haoran Geng <ghr@berkeley.edu>

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

In summary, our contributions include:

- A real-world experimental setup with bi-manual anthropomorphic hands, teleoperation, and a benchmark for visuo-tactile dexterous manipulation.
- A multimodal learning framework that couples cross-attention fusion with autoregressive tactile prediction and curriculum training.
- Strong empirical results, including significant gains over baselines and the first successful demonstration of long-horizon dexterous manipulation on a real robot.

2 Related Work

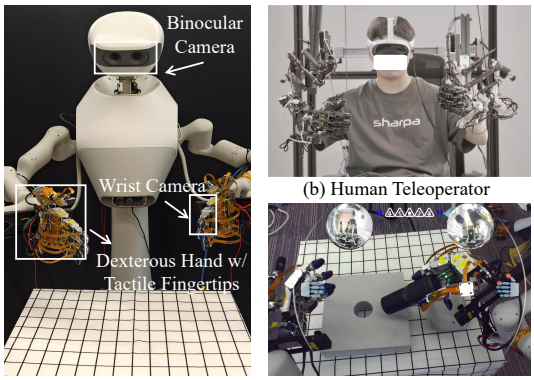
Dexterous Manipulation. Dexterous manipulation has been widely studied in grasping, in-hand control, articulated and deformable object interaction [16, 17, 18, 28, 29, 30, 31, 32, 33]. Behavior cloning (BC) provides an end-to-end paradigm for these tasks. Diffusion Policy (DP) [34] learns expert actions with diffusion models [35, 36], and its 3D variant (DP3) [37] leverages point clouds for improved generalization. Action Chunking Transformer (ACT) [38] instead formulates BC as a conditional variational auto-encoder, showing stronger data efficiency [14]. HATO [20] further extends diffusion policy with tactile inputs. Our ViTacFormer builds on ACT and introduces cross-modal representation learning for visuo-tactile dexterous manipulation.

Manipulation with Tactile Signals. Prior works have explored tactile sensing through force feedback or self-supervised feature learning [19, 21, 22, 23, 24, 25, 39]. Contrastive and masked auto-encoding have been particularly effective for high-resolution tactile images. However, existing models treat vision and touch independently or fuse them naively. In contrast, ViTacFormer uses cross-attention to tightly couple modalities and autoregressive tactile forecasting to enhance policy learning, enabling robust performance on both short- and long-horizon dexterous tasks.

3 Problem Formulation and Hardware Setup

3.1 Problem Formulation

We study imitation learning for dexterous bi-manual manipulation. Given a set of 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, the goal is to learn a policy π_θ that maps observations to actions, $a_t = \pi_\theta(o_t)$. The policy is trained to imitate expert behavior and evaluated on both short- and long-horizon manipulation tasks.



(a) Hardware System (b) Human Teleoperator (c) First-person VR View

Figure 1: System overview. (a) Dual-arm platform with dexterous hands, wrist cameras, and stereo camera. (b) Teleoperation with exoskeleton gloves and VR headset. (c) VR interface with visual and tactile feedback.

3.2 Hardware Setup

Our platform (Fig. 1) consists of two robot arms with anthropomorphic dexterous hands, equipped with wrist-mounted and stereo cameras for vision and fingertip sensors for touch. We collect demonstrations via a custom teleoperation system where operators use exoskeleton gloves and a VR interface with real-time visual–tactile feedback. This setup enables intuitive control and provides high-quality multimodal trajectories.

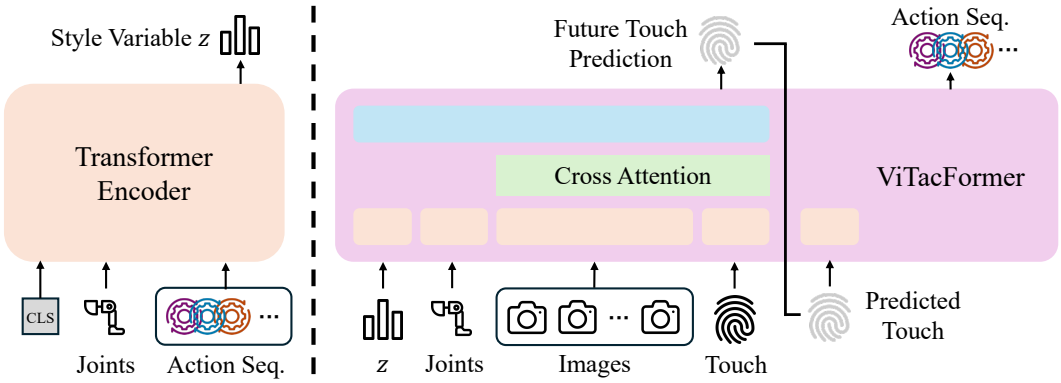


Figure 2: ViTacFormer architecture. A transformer-based encoder maps action sequence and robot proprioception to a style variable z. A transformer-based encoder-decoder fuses z with visuo-tactile observations to autoregressively predict future tactile signals and generate actions.

4 Method

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

4.1 Cross-Attention-Based Multimodal Integration

Visual observations and tactile signals provide complementary cues but naive fusion fails to capture their correlations. Cross-attention enables queries from one modality to attend to keys and values from the other, allowing ViTacFormer to extract semantically relevant dependencies between vision and touch. The resulting features are concatenated into hidden states for downstream action generation.

4.2 Auto-Regressive Modeling with Tactile Signal Forecasting

Beyond perceiving current signals, anticipating future tactile feedback improves robustness. ViTacFormer predicts future tactile tokens using z, proprioception, and visuo-tactile observations, and concatenates them with current inputs for action generation.

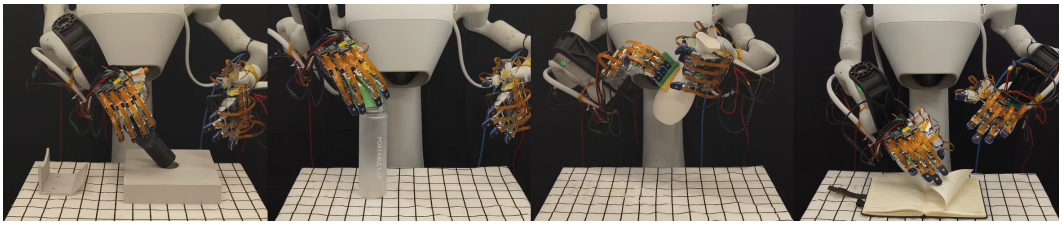
4.3 Neural Network Architecture and Learning Procedure

Fig. 2 illustrates the architecture of ViTacFormer, formulated as a conditional variational auto-encoder. On the left, a transformer-based encoder maps the robot's proprioception and expert action sequence into a latent style variable z. On the right, a transformer-based encoder-decoder integrates z with visuo-tactile observations via cross-attention, predicts future tactile signals autoregressively, and generates actions accordingly. During training, z is sampled from expert demonstrations, while during inference it is fixed to zero, ensuring consistent action generation without stochastic variation.

Since autoregressive tactile forecasting is challenging at the start of training, we adopt a two-phase curriculum: the first 75% of epochs use ground-truth tactile signals to stabilize learning, and the final 25% gradually switch to predicted signals, encouraging robust cross-modal reasoning under realistic conditions [40].

5 Experiment

We evaluate ViTacFormer on four short-horizon dexterous tasks (Fig. 3) and one long-horizon task (making hamburgers). Each task uses only 50 demonstrations, making the setting challenging.



Peg Insertion | Cap Twist | Vase Wipe | Book Flip

Figure 3: Four short-horizon visuo-tactile tasks: peg insertion, cap twist, vase wipe, and book flip.

Metrics We evaluate algorithms with two metrics: success rates and Human Normalized Score (HNS). Success rate reflects overall task completion, but may overlook execution quality in long-horizon tasks. HNS provides a stage-wise evaluation: each task is split into N stages, with raw scores $s_i \in \{0, 1, 2, 3\}$ and weights w_i indicating tactile reliance. The normalized score is defined as:

$$\text{HNS} = \frac{\sum_{i=1}^{N} w_i \cdot s_i}{3 \cdot \sum_{i=1}^{N} w_i}. \tag{1}$$

This captures both stage performance and tactile sensitivity, complementing success rates.

Baselines We compare against Diffusion Policy (DP) [34], HATO [20], ACT [38], and ACTw/T [38]. DP and ACT do not use tactile inputs, while HATO and ACTw/T fuse tactile signals naively.

5.1 Algorithm Comparison

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 [34]	$2/10$	$0/10$	$3/10$	$1/10$
ACT [38]	$4/10$	$4/10$	$3/10$	$2/10$
HATO [20]	$4/10$	$1/10$	$4/10$	$3/10$
ACTw/T [38]	$6/10$	$6/10$	$4/10$	$4/10$
Ours	$\mathbf{10}/10$	$\mathbf{10}/10$	$\mathbf{9}/10$	$\mathbf{9}/10$

To evaluate the effectiveness of our approach, we conduct experiments on four short-horizon dexterous manipulation tasks, with each algorithm tested for 10 rollouts per task. As shown in Tab. 1, ViTacFormer consistently outperforms all baselines, achieving markedly higher success rates and nearly solving these challenging benchmarks.

We further evaluate ViTacFormer on a challenging 11-stage long-horizon task of making hamburgers. As shown in Fig. 4, ViTacFormer is the first to complete the full sequence on a real robot, achieving over 80% overall success rate. To quantify long-horizon performance beyond binary success, Table 2 reports the stage-wise raw scores and the overall Human Normalized Score (HNS); our method maintains consistently high stage scores and attains an overall HNS of 0.88, while the baseline without tactile input drops to 0.61.

Figure 4: Successful rollout of ViTacFormer on the long-horizon hamburger task.

Table 2: Long-horizon hamburger task: stage-wise raw scores (0–3) and overall HNS (0–1). ViTacFormer shows strong and stable performance across all 11 stages and a higher overall HNS.

Stage	1	2	3	4	5	6	7	8	9	10	11	Overall
w/o Touch [38]	2.4	2.5	1.9	2.0	0.7	2.2	1.6	2.8	2.2	2.2	0.7	0.61
Ours	**2.9**	**3.0**	**1.9**	1.8	**2.7**	**2.9**	**2.0**	**2.8**	**2.4**	**2.5**	**3.0**	**0.88**

5.2 Ablation Study

Results in Tab. 3 show that removing any component leads to noticeable performance drops. Cross-attention enhances vision-touch fusion, autoregressive tactile forecasting improves stability, and the two-stage curriculum stabilizes training. Together, these components enable ViTacFormer to achieve the best performance.

Table 3: Ablation study on four short-horizon dexterous tasks. Each proposed component improves performance over the baseline.

Task	Peg Insertion	Cap Twist	Vase Wipe	Book Flip
w/o Tactile	4/10	4/10	3/10	2/10
w/o CrossAttention	9/10	7/10	7/10	7/10
w/o AutoRegressive	7/10	7/10	6/10	7/10
w/o Two-Stage	7/10	6/10	4/10	6/10
Ours	**10/10**	**10/10**	**9/10**	**9/10**

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 visuo-tactile framework for dexterous manipulation that integrates cross-modal fusion and predictive tactile modeling. With a curriculum strategy to stabilize training, ViTacFormer achieves robust control and outperforms strong baselines by large margins. Notably, it is the first system to complete long-horizon dexterous tasks on a real robot, highlighting the potential of vision-touch integration for generalizable manipulation.

Acknowledgments

If a paper is accepted, the final camera-ready version will (and probably should) include acknowledgments. All acknowledgments go at the end of the paper, including thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

[1] H. Geng, F. Wang, S. Wei, Y. Li, B. Wang, B. An, C. T. Cheng, H. Lou, P. Li, Y.-J. Wang, Y. Liang, D. Goetting, C. Xu, H. Chen, Y. Qian, Y. Geng, J. Mao, W. Wan, M. Zhang, J. Lyu, S. Zhao, J. Zhang, J. Zhang, C. Zhao, H. Lu, Y. Ding, R. Gong, Y. Wang, Y. Kuang, R. Wu, B. Jia, C. Sferrazza, H. Dong, S. Huang, Y. Wang, J. Malik, and P. Abbeel. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning, 2025. URL https://arxiv.org/abs/2504.18904.

[2] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. *arXiv preprint arXiv:2211.05272*, 2022.

[3] R. Luo*, H. Geng*, C. Deng, P. Li, Z. Wang, B. Jia, L. Guidbas, and S. Huang. Physpart: Physically plausible part completion for interactable objects. *International Conference on Robotics and Automation (ICRA)*, 2025. URL https://arxiv.org/abs/2408.13724.

[4] H. Geng, Z. Li, Y. Geng, J. Chen, H. Dong, and H. Wang. Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations. *arXiv preprint arXiv:2303.16958*, 2023.

[5] Y. Seo, C. Sferrazza, H. Geng, M. Nauman, Z.-H. Yin, and P. Abbeel. Fasttd3: Simple, fast, and capable reinforcement learning for humanoid control, 2025. URL https://arxiv.org/abs/2505.22642.

[6] J. Zhang, H. Geng, Y. You, C. Deng, P. Abbeel, J. Malik, and L. Guibas. Rodrigues network for learning robot actions, 2025. URL https://arxiv.org/abs/2506.02618.

[7] H. Geng, S. Wei, C. Deng, B. Shen, H. Wang, and L. Guibas. Sage: Bridging semantic and actionable parts for generalizable articulated-object manipulation under language instructions, 2023.

[8] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, H. Wang, and Y. Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. *arXiv preprint arXiv:2407.04689*, 2024.

[9] Y. Kuang, H. Geng, A. Elhafsi, T.-D. Do, P. Abbeel, J. Malik, M. Pavone, and Y. Wang. Skillblender: Towards versatile humanoid whole-body loco-manipulation via skill blending. *arXiv preprint arXiv:2506.09366*, 2025.

[10] Y. Ding, H. Geng, C. Xu, X. Fang, J. Zhang, S. Wei, Q. Dai, Z. Zhang, and H. 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.

[11] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning (CoRL)*, pages 785–799. PMLR, 2023.

[12] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning (CoRL)*, pages 894–906. PMLR, 2022.

[13] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, et al. Octo: An open-source generalist robot policy, 2023.

[14] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024.

[15] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. *arXiv preprint arXiv:2303.00938*, 2023.

[16] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, and H. Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. *arXiv preprint arXiv:2304.00464*, 2023.

[17] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. *arXiv preprint arXiv:2210.02697*, 2022.

[18] J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, and H. Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *8th Annual Conference on Robot Learning*.

[19] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3):582–596, 2020.

[20] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *IEEE International Conference on Robotics & Automation (ICRA)*, 2025.

[21] F. Yang, C. Feng, Z. Chen, H. Park, D. Wang, Y. Dou, Z. Zeng, X. Chen, R. Gangopadhyay, A. 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.

[22] Z. Xu, R. Uppuluri, X. Zhang, C. Fitch, P. G. Crandall, W. Shou, D. Wang, and Y. She. Unit: Unified tactile representation for robot learning. *arXiv preprint arXiv:2408.06481*, 2024.

[23] K. Yu, Y. Han, Q. Wang, V. Saxena, D. Xu, and Y. Zhao. Mimictouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. *arXiv preprint arXiv:2310.16917*, 2023.

[24] C. Higuera, A. Sharma, C. K. Bodduluri, T. Fan, P. Lancaster, M. Kalakrishnan, M. Kaess, B. Boots, M. Lambeta, T. Wu, et al. Sparsh: Self-supervised touch representations for vision-based tactile sensing. *arXiv preprint arXiv:2410.24090*, 2024.

[25] L. Fu, G. Datta, H. Huang, W. C.-H. Panitch, J. Drake, J. Ortiz, M. Mukadam, M. Lambeta, R. Calandra, and K. Goldberg. A touch, vision, and language dataset for multimodal alignment. *arXiv preprint arXiv:2402.13232*, 2024.

[26] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023.

[27] Y. Yuan, H. Che, Y. Qin, B. Huang, Z.-H. Yin, K.-W. Lee, Y. Wu, S.-C. Lim, and X. Wang. Robot synesthesia: In-hand manipulation with visuotactile sensing. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6558–6565. IEEE, 2024.

[28] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang. Rotating without seeing: Towards in-hand dexterity through touch. *arXiv preprint arXiv:2303.10880*, 2023.

[29] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, Y. Narang, J.-F. Lafleche, D. Fox, and G. State. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. *arXiv*, 2022.

[30] C. Bao, H. Xu, Y. Qin, and X. 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.

[31] T. Jiang, L. Ma, Y. Guan, J. Meng, W. Chen, Z. Zeng, L. Li, D. Wu, J. Xu, and R. Chen. Dexsim2real^2: Building explicit world model for precise articulated object dexterous manipulation, 2024. URL https://arxiv.org/abs/2409.08750.

[32] S. Li, Z. Huang, T. Chen, T. Du, H. Su, J. B. Tenenbaum, and C. Gan. Dexdeform: Dexterous deformable object manipulation with human demonstrations and differentiable physics, 2023. URL https://arxiv.org/abs/2304.03223.

[33] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel. Learning to manipulate deformable objects without demonstrations, 2020. URL https://arxiv.org/abs/1910.13439.

[34] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[35] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[36] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[37] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

[38] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[39] C. Sferrazza, Y. Seo, H. Liu, Y. Lee, and P. 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.

[40] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *https://arxiv. org/abs/2410.24164*.

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 (180×320) from top-mounted ZED Mini cameras (Fig. 5**(a)**, **(c)**), and two fisheye wrist-mounted views (256×280) for left and right hands (Fig. 5**(b)**, **(d)**). All frames are encoded into image tokens via a vision backbone before cross-modal integration.

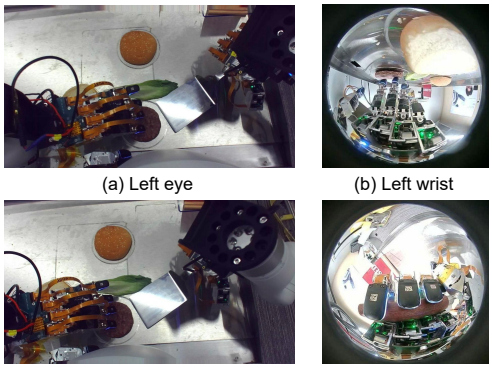

(a) Left eye
(b) Left wrist
(c) Right eye
(d) Right wrist

Figure 5: **Four types of camera views**

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.

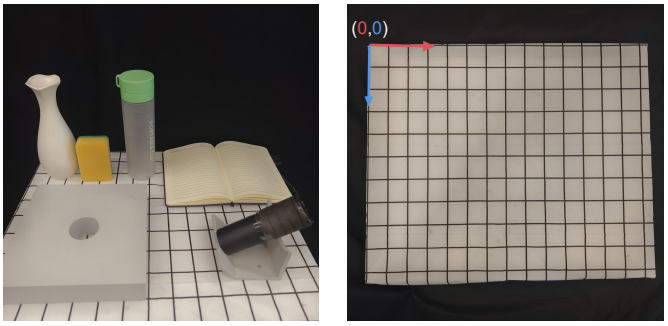
A.4 Inference Details

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 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. 6**(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. 6**(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.



(a) Objects (b) Workspace

Figure 6: **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. 7.

Scoring Scheme

Table 4: **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

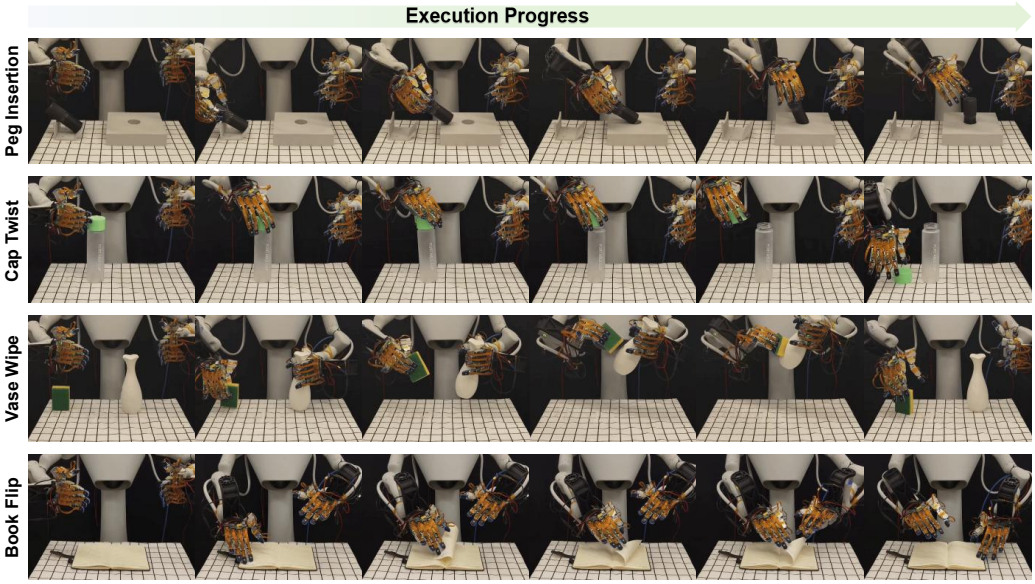


Figure 7: **Execution examples for short-horizon tasks.** Representative keyframes from four tasks: peg insertion, cap twist, vase wipe, and book flip. Each task demonstrates a full execution sequence from perception to manipulation.

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.

Inference Results

Table 5: **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 5 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 8, 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.

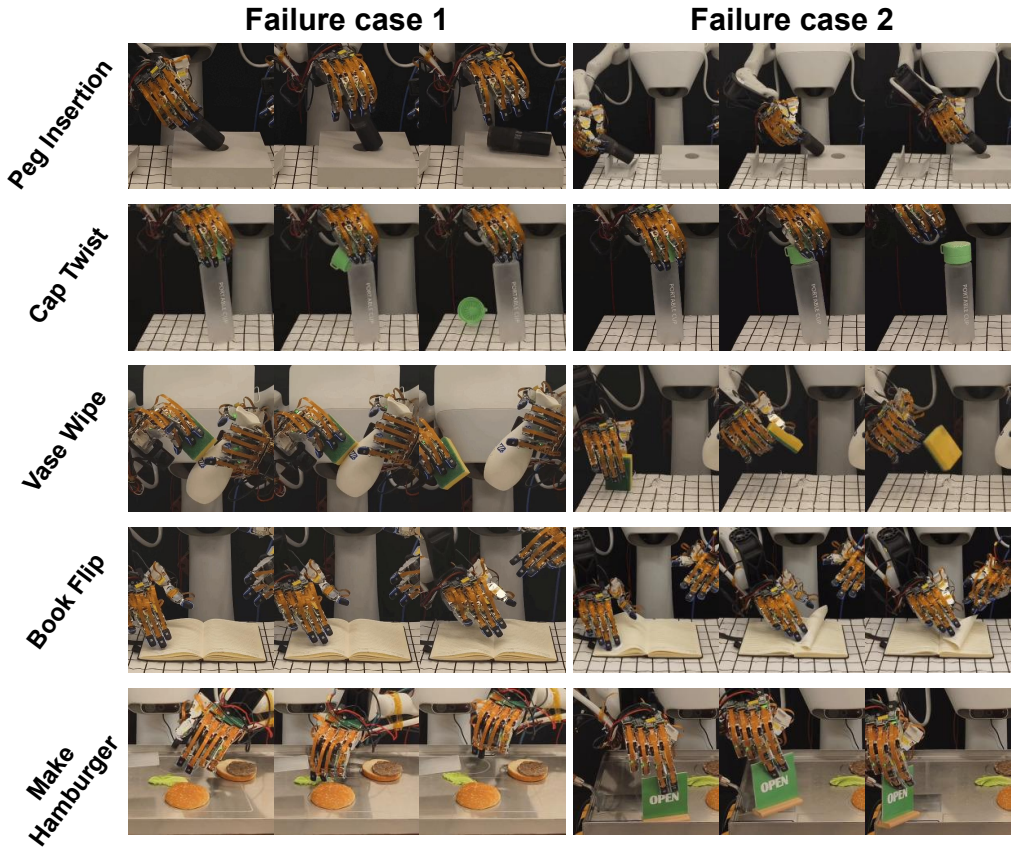


Figure 8: **Representative failure cases across all tasks.** Each row corresponds to one task, with two failure case sequences shown side by side.

B.1.2 Cap Twist

Task Description

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. 7.

Scoring Scheme

Table 6: **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 7: **Cap Twist: inference results across models.**

Model	Stage 1	Stage 2	HNS	Success Rate
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 7 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. 8, 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. 7.

Scoring Scheme

Table 8: **Scoring criteria for Vase Wipe.**

Stage 1: Pick (weight 1)	Description
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
Stage 2: Wipe (weight 2)	**Description**
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 9: **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%

Table 9 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. 8 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 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. 7.

Scoring Scheme

Table 10: **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 11 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 8, 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

Table 11: **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%

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. 9. 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.

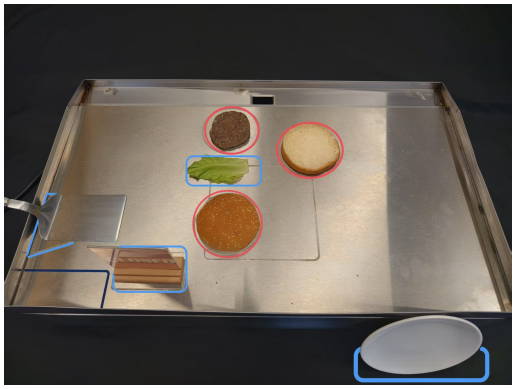


Figure 9: **Long-horizon task setup.** Seven components are placed in predefined zones—circular (ingredients) or rectangular (tools). Objects are randomly initialized within these areas to test spatial generalization.

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 12 details the scoring criteria and weights for each stage.

Table 12: **Scoring criteria for the long-horizon hamburger task.**

Stage	Action	Weight	Score Description
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. 8 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.