# Robot Synesthesia: In-Hand Manipulation with Visuotactile Sensing

**Ying Yuan**[*2], **Haichuan Che**[*1], **Yuzhe Qin**[*1], **Binghao Huang**[3], **Zhao-Heng Yin**[4], **Kang-Won Lee**[5], **Yi Wu**[2], **Soo-Chul Lim**[5], **Xiaolong Wang**[1]

UC San Diego[1] Tsinghua University[2] University of Illinois Urbana-Champaign[3]
University of California Berkeley[4] Dongguk University[5]

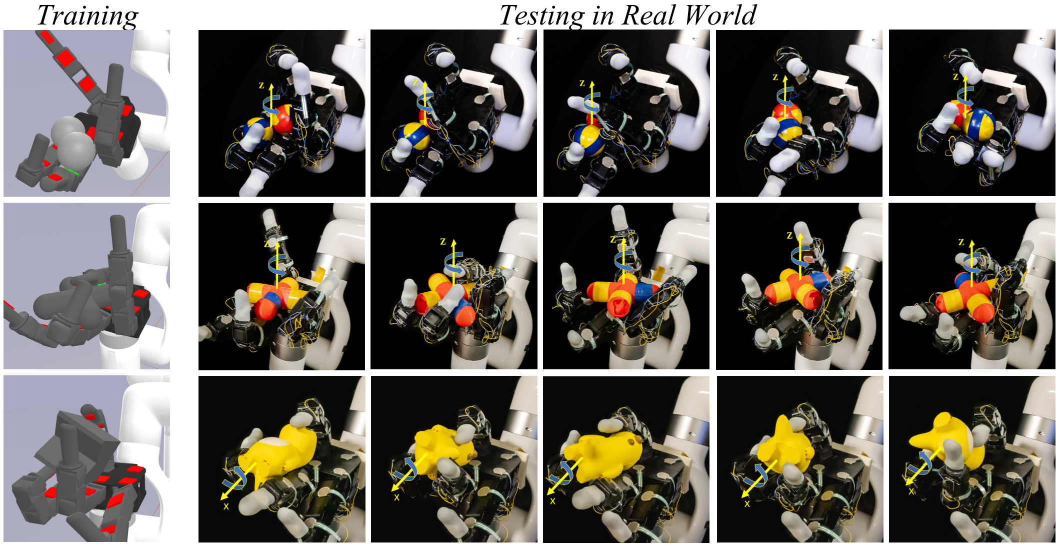


Figure 1: We propose **Robot Synesthesia**, a novel visuotactile approach to perform in-hand object rotation with visual and tactile modalities.

## Abstract

Executing contact-rich manipulation tasks necessitates the fusion of tactile and visual feedback. However, the distinct nature of these modalities poses significant challenges. In this paper, we introduce a system that leverages visual and tactile sensory inputs to enable dexterous in-hand manipulation. Specifically, we propose **Robot Synesthesia**, a novel point cloud-based tactile representation inspired by human tactile-visual synesthesia. This approach allows for the simultaneous and seamless integration of both sensory inputs, offering richer spatial information and facilitating better reasoning about robot actions. Comprehensive ablations are performed on how the integration of vision and touch can improve reinforcement learning and Sim2Real performance. Our project page is available at https://yingyuan0414.github.io/visuotactile/.

## 1 Introduction

In everyday life, humans effortlessly perform complex manipulation tasks, intuitively using a combination of vision and touch. Considering the intricate task of threading a needle, we begin by visually

[*] Indicates equal contribution.

NeurIPS 2023 Workshop on Touch Processing: a new Sensing Modality for AI .

locating the needle's eye and estimating its size. Holding the needle steady in hand, we use touch information to guide the thread. Our visual sensing guides us in aligning the thread with the needle's eye, while it's the sense of touch from our fingertips that helps us feel the thread's position, even when it's occluded for our eyes to discern accurately. This synergy of vision and touch enables us to interact with our environment with remarkable flexibility and great robustness against occlusion.

Yet, for robots tasked with similar manipulation tasks, achieving this level of sophistication remains a challenge. There are two primary hurdles that stand in the way of replicating the same level of synergy for robot learning algorithms. (i) *Tactile and visual modality are distinct in nature.* Tactile information is typically sparse and low-dimensional, captured from distinct tactile sensors and provides little contextual details. On the other hand, visual data is dense and high-dimensional, offering a rich tapestry of environmental details. When integrating these two types of data into a single neural network, the model must process and interpret each modality effectively, while also finding a way to synergize this information to facilitate intelligent decision-making. (ii) *Vast amounts of training data are required for such tasks*, which is typically generated within a simulated environment. However, transferring the visuotactile skills learned in a simulator to the real world is a non-trivial problem. Each modality, vision and touch, has its own domain gap. Bridging them concurrently for a combined visual-tactile model even heightens the complexity significantly.

In this paper, we aim to equip the robot with a policy that effectively leverages multi-modal feedback. In neuroscience studies, certain individuals can perceive color when they touch things, which refers to Tactile-Visual Synesthesia (1; 2). Inspired by it, instead of processing each modality separately in representation learning and merging the learned features later, we propose a novel point cloud-based tactile representation. We formulate this representation in a way that "paints" the tactile data from Force-Sensing Resistor (FSR) in conjunction with the point cloud from the camera into a unified 3D space. This approach preserves the spatial relationship among the robot links, FSR sensors, and the object being manipulated. Effectively, the robot is equipped to "see" its tactile interactions, a concept we call **Robot Synesthesia**. This method allows for the seamless integration of both sensory inputs from the outset, which offers abundant spatial information, facilitating better reasoning about robot actions. Furthermore, we can easily generate these tactile point clouds in both simulated and real-world using the robot's kinematics. This strategy can reduce the compounding errors of vision and touch during sim-to-real transfer, by treating these two modalities as an integrated entity.

## 2 Visuotactile Dexterous Manipulation

To obtain visual observations, we place a Microsoft Azure Kinect camera beside the hand in both real and simulated settings. We then generate the point cloud using the depth image. The point clouds in simulation closely mirror those in reality, especially compared with RGB images. We create an augmented point cloud (3) from the robot's proprioception to model the spatial relationship between the hand and the object, by sampling on the robot's mesh at the current pose.

### 2.1 Benchmark Problems

In this paper, we mainly focus on three distinct benchmark problems: **(i) Wheel-Wrench Rotation**: Inspired by scenarios where a user must switch handles on a wrench during use, this task involves rotating an artificial multi-way wheel wrench along the z-axis in hand without dropping it. To successfully complete this task, the robot must visually identify the next "possible" handle for interaction while concurrently sensing the wrench's rotation via touch. **(ii) Double-Ball Rotation**: This task requires the simultaneous manipulation of two identical balls to rotate around each other along the z-axis. Given that tactile feedback alone cannot distinguish between the balls, it is crucial for the robot to visually locate both. **(iii) Three-Axis Rotation**: This task extends beyond the z-axis to require the robot to rotate objects around a fixed x or y-axis. Moreover, the policy should demonstrate the ability to manipulate a variety of objects with different shapes, extending its dexterity to objects not included in the training set.

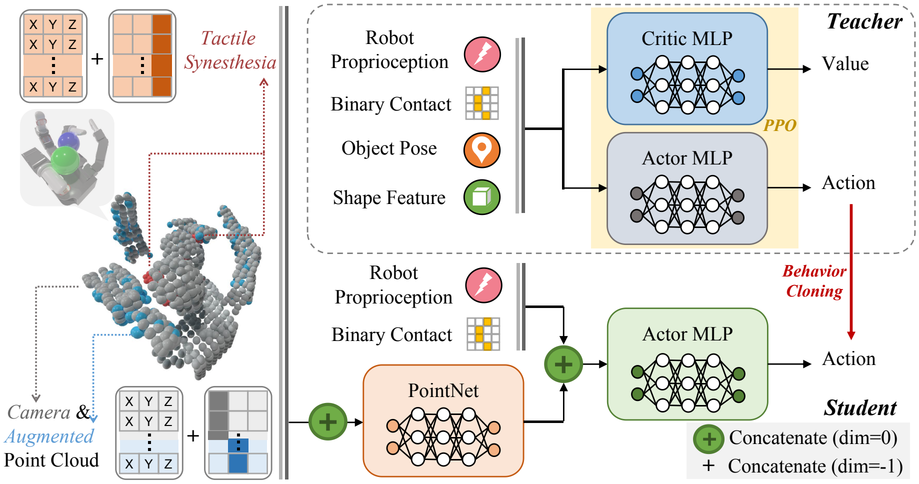


Figure 2: **Training Pipeline.** Our teacher policy takes robot proprioception, binary contact, object pose, and object shape embedding as input. After training the teacher policy via RL, we distill it to a visuotactile-based student policy. Besides robot proprioception and touch signal, the student policy takes a point cloud from depth-camera, an augmented point cloud based on robot proprioception, and the proposed tactile point cloud. We use one-hot vectors to distinguish point clouds. Note that we've eliminated noise from the point clouds for better clarity here.

## 3 Learning Visuotactile Dexterity

### 3.1 Tactile-Visual Synesthesia

Instead of processing tactile and visual modalities separately for feature extraction, we unify tactile and visual modalities by projecting them onto a single 3D space, similar to how humans might simultaneously perceive touch and visual stimuli in their minds. Concretely, for each tactile sensor on the hand that detects a signal (i.e., $o_{t,i} = 1$), we sample points on the sensor's meshes to create a tactile-based point cloud $P_t^{touch}$. When combining $P_t^{touch}$ with $P_t^c$ and $P_t^a$, we provide the policy network with a spatial relationship of all the observation entities. In our implementation, we set the number of sampled points $N_c = 512, N_a = 8n_{link}$, and $N_t = 8n_{touch}$, where $n_{link} = 21$ is the number of links on hand and $n_{touch} \in \{0, \cdots, 16\}$ is the number of triggered tactile sensors. Our experiments demonstrate that points sampled from active tactile sensors are implicitly chosen by our learned policy for representation learning. Note that we transform all point clouds to the hand palm (4) frame before feeding them into the neural network.

### 3.2 Teacher-Student Training Pipeline

Learning the controller $\pi$ using RL is data inefficient when the observation is high-dimensional, e.g., point clouds. To mitigate this issue, we employ a teacher-student learning approach to obviate training vision policies with RL, as shown in Figure 2.

## 4 Experiments

In this part, we compare our robot synesthesia approach to several baselines in both the simulation and the real. Specifically, we are interested in the following questions:

1) *How much benefit do visual and tactile modalities offer?*
2) *Is teacher-student pipeline necessary for efficient training given both tactile and visual modalities?*
3) *How does our policy network process the two modalities through synesthesia?*

We answer these questions through an extensive case study on the benchmark problems.

### 4.1 Stage I: RL training with different sensing capabilities

Table 1: Evaluation of RL policies of different sensing capabilities on three benchmark problems in the simulation. Each policy is tested for 500 episodes. The results are averaged over 3 policies trained on 3 seeds. Each trial lasts 50 seconds.

| Obs Type | 4-way Wrench | | Double Balls | | Multi-Object (x-axis) | | Multi-Object (y-axis) | | Multi-Object (z-axis) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CRR | TTF | CRR | TTF | CRR | TTF | CRR | TTF | CRR | TTF |
| Visual RL | 10.9$_{\pm 2.2}$ | 8.1$_{\pm 3.2}$ | 127.8$_{\pm 78.6}$ | 10.5$_{\pm 3.7}$ | 15.3$_{\pm 8.2}$ | 16.8$_{\pm 11.8}$ | 22.4$_{\pm 8.8}$ | 21.4$_{\pm 17.8}$ | 29.5$_{\pm 7.1}$ | 2.9$_{\pm 0.4}$ |
| PS | 440.7$_{\pm 590.3}$ | 22.6$_{\pm 18.5}$ | 620.9$_{\pm 39.9}$ | 28.8$_{\pm 0.7}$ | 446.1$_{\pm 137.7}$ | 33.1$_{\pm 7.1}$ | 552.1$_{\pm 318.7}$ | 33.5$_{\pm 8.3}$ | 878.7$_{\pm 528.3}$ | 36.9$_{\pm 15.4}$ |
| Ours | **1011.1**$_{\pm 329.9}$ | **47.5**$_{\pm 0.4}$ | **1045.3**$_{\pm 64.9}$ | **36.2**$_{\pm 2.3}$ | **985.9**$_{\pm 174.1}$ | **45.1**$_{\pm 2.6}$ | **987.3**$_{\pm 141.9}$ | **46.8**$_{\pm 1.0}$ | **1353.7**$_{\pm 123.8}$ | **48.2**$_{\pm 0.4}$ |

Table 2: Evaluation of student policies of different sensing capabilities on three benchmark problems in the simulation. Each policy is tested for 500 episodes. Each trial lasts 50 seconds.

| Obs Type | 4-way Wrench | | Double Balls | | X-Axis | | Y-Axis | | Z-Axis) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CRR | TTF | CRR | TTF | CRR | TTF | CRR | TTF | CRR | TTF |
| Touch | 363.2 | 23.6 | 317.1 | 13.6 | 390.9 | 24.2 | 710.9 | 42.6 | 702.4 | 35.6 |
| Cam+Aug | 94.6 | 15.2 | 162.7 | 9.6 | 630.9 | 40.3 | **743.5** | **42.9** | 624.2 | 29.2 |
| Touch+Cam+Aug | 344.1 | 21.1 | 148.6 | 9.6 | **881.1** | **47.4** | 619.0 | 41.3 | 909.8 | 37.7 |
| Ours | **504.0** | **29.2** | **407.7** | **17.1** | 846.9 | 39.9 | 686.8 | 41.2 | **1035.0** | **41.3** |

In this section, we evaluate policies trained on the benchmark problems for 500 episodes as is shown in Table 1. Our approach outperforms PS and Visual RL in all the tasks. This indicates that the ground-truth object pose is essential for robust and meticulous manipulation, especially when the object, e.g. multi-way wrenches, requires different actions as its direction varies. Also, learning vision-based RL policies is data-inefficient, probably because the policy needs to extract features from high-dimensional inputs and learn high-rewarding actions simultaneously. More information about the definition of evaluation metric can be found in the appendix.

### 4.2 Stage II: Imitation learning with different sensing capabilities

In this section, we distill teacher policies to visuotactile-based policies and perform ablation study for different sensing capabilities. As shown in Table 2, **Touch** refers to binary contact, **Cam** refers to camera-based point clouds, **Aug** refers to augmented point clouds, and **Syn** refers to proposed tactile point clouds. For rotating objects of regular shapes, visual policies achieve similar dexterity to each other. However, when it comes to more challenging objects like multi-way wrenches and two balls, our visual-tactile synesthesia method outperforms all the baselines.

### 4.3 Real-world Deployment

A rudimentary concatenation of tactile signals and the extracted features of point clouds could increase the challenge for the policy to comprehend their underlying relationship. In contrast, our proposed visual-tactile synesthesia approach generally offers benefits. We also observe that visual policies tend to operate more cautiously, making occasional adjustments to nudge the object back to the palm center, while policies lacking visual perception tend to execute an almost fixed sequence of motion, irrespective of the object's deviation or instances of it becoming stuck.

## 5 Conclusion

This paper introduces a system for in-hand dexterous manipulation utilizing visuotactile sensing. We propose a novel tactile representation based on point clouds, and a paired network architecture to leverage it. Our results show that the policy, which has been trained in a simulator using vision and touch input, effectively transfers to the real world. It can solve complex tasks such as double-ball rotation and generalize to novel objects. Future work may encompass goal-conditioned object rotation tasks and the integration of optical tactile sensors. We are committed to releasing the code for our simulation environment and training pipeline.

# References

[1] J. Simner and V. U. Ludwig, “The color of touch: A case of tactile–visual synaesthesia,” *Neurocase*, vol. 18, no. 2, pp. 167–180, 2012.

[2] A. M. A. Davies and R. C. White, “A sensational illusion: vision-touch synaesthesia and the rubber hand paradigm,” *Cortex*, vol. 49, no. 3, pp. 806–818, 2013.

[3] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, “Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 594–605.

[4] M. Liu, X. Li, Z. Ling, Y. Li, and H. Su, “Frame mining: a free lunch for learning robotic manipulation from 3d point clouds,” *arXiv preprint arXiv:2210.07442*, 2022.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” *CoRR*, vol. abs/1612.00593, 2016.

[7] Y.-H. Wu, J. Wang, and X. Wang, “Learning generalizable dexterous manipulation from human grasp affordance,” in *Conference on Robot Learning (CoRL)*, 2022.

[8] S. Ross, G. J. Gordon, and J. A. Bagnell, “No-regret reductions for imitation learning and structured prediction,” *CoRR*, vol. abs/1011.0686, 2010.

[9] 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.

[10] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, “Isaac gym: High performance gpu-based physics simulation for robot learning,” *arXiv preprint arXiv:2108.10470*, 2021.

[11] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik, “In-hand object rotation via rapid motor adaptation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 1722–1732.

[12] A. M. Okamura, N. Smaby, and M. R. Cutkosky, “An overview of dexterous manipulation,” in *IEEE International Conference on Robotics and Automation. Symposia Proceedings*, 2000.

[13] N. Chavan-Dafle and A. Rodriguez, “Sampling-based planning of in-hand manipulation with external pushes,” in *Robotics Research: The 18th International Symposium ISRR*. Springer, 2020, pp. 523–539.

[14] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, “Learning dexterous in-hand manipulation,” *The International Journal of Robotics Research (IJRR)*, vol. 39, no. 1, pp. 3–20, 2020.

[15] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, “Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system,” *arXiv preprint arXiv:2307.04577*, 2023.

[16] K. Shaw and D. Pathak, “Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning,” *Submission, ICRA*, 2023.

[17] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang, “Learning continuous grasping function with a dexterous hand from human demonstrations,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[18] M. Cherif and K. K. Gupta, “Planning quasi-static fingertip manipulations for reconfiguring objects,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 837–848, 1999.

[19] J. Shi, J. Z. Woodruff, P. B. Umbanhowar, and K. M. Lynch, “Dynamic in-hand sliding manipulation,” *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 778–795, 2017.

[20] L. Han, Y.-S. Guan, Z. Li, Q. Shi, and J. C. Trinkle, “Dextrous manipulation with rolling contacts,” in *International Conference on Robotics and Automation (ICRA)*, 1997.

[21] L. Han and J. Trinkle, “Dextrous manipulation by rolling and finger gaiting,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 1998.

[22] A. Bicchi and R. Sorrentino, “Dexterous manipulation through rolling,” in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 1, 1995, pp. 452–457 vol.1.

[23] Z. Doulgeri and L. Droukas, “On rolling contact motion by robotic fingers via prescribed performance control,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[24] M. Lepert, C. Pan, S. Yuan, R. Antonova, and J. Bohg, “In-hand manipulation of unknown objects with tactile sensing for insertion,” in *Embracing Contacts-Workshop at ICRA 2023*, 2023.

[25] Y. Aiyama, M. Inaba, and H. Inoue, “Pivoting: A new method of graspless manipulation of object by robot fingers,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1993.

[26] E. Yoshida, P. Blazevic, and V. Hugel, “Pivoting manipulation of a large object: A study of application using humanoid platform,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 1040–1045.

[27] P. Tournassoud, T. Lozano-Pérez, and E. Mazer, “Regrasping,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 1987.

[28] P. Vinayavekhin, S. Kudoh, and K. Ikeuchi, “Towards an automatic robot regrasping movement based on human demonstration using tangle topology,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3332–3339.

[29] A. A. Cole, P. Hsu, and S. S. Sastry, “Dynamic control of sliding by robot hands for regrasping,” *IEEE Transactions on robotics and automation*, vol. 8, no. 1, pp. 42–52, 1992.

[30] V. Kumar, Y. Tassa, T. Erez, and E. Todorov, “Real-time behaviour synthesis for dynamic hand-manipulation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6808–6815.

[31] Y. Bai and C. K. Liu, “Dexterous manipulation using both palm and fingers,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1560–1565.

[32] T. Chen, J. Xu, and P. Agrawal, “A system for general in-hand object re-orientation,” in *Conference on Robot Learning (CoRL)*, 2022, pp. 297–307.

[33] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, “Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation,” in *Conference on Robot Learning (CoRL)*, 2022.

[34] G. Khandate, M. Haas-Heger, and M. Ciocarlie, “On the feasibility of learning finger-gaiting in-hand manipulation with intrinsic sensing,” in *International Conference on Robotics and Automation (ICRA)*, 2022.

[35] 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*, 2023, pp. 21 190–21 200.

[36] B. Huang, Y. Chen, T. Wang, Y. Qin, Y. Yang, N. Atanasov, and X. Wang, “Dynamic handover: Throw and catch with bimanual hands,” in *7th Annual Conference on Robot Learning*, 2023.

[37] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, “Dexmv: Imitation learning for dexterous manipulation from human videos,” in *European Conference on Computer Vision (ECCV)*, 2022.

[38] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," in *Robotics: Science and Systems (RSS)*, 2018.

[39] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto, "Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation," *arXiv preprint arXiv:2203.13251*, 2022.

[40] Y. Qin, H. Su, and X. Wang, "From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 873–10 881, 2022.

[41] X. Liu, D. Pathak, and K. M. Kitani, "Herd: Continuous human-to-robot evolution for learning from human demonstration," *arXiv preprint arXiv:2212.04359*, 2022.

[42] A. Patel, A. Wang, I. Radosavovic, and J. Malik, "Learning to imitate object interactions from internet videos," *arXiv preprint arXiv:2211.13225*, 2022.

[43] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto, "Holo-dex: Teaching dexterity with immersive mixed reality," *arXiv preprint arXiv:2210.06463*, 2022.

[44] A. Sivakumar, K. Shaw, and D. Pathak, "Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube," *arXiv preprint arXiv:2202.10448*, 2022.

[45] A. Bhatt, A. Sieler, S. Puhlmann, and O. Brock, "Surprisingly robust in-hand manipulation: An empirical study," in *Robotics: Science and Systems (RSS)*, 2021.

[46] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *Conference on Robot Learning*. PMLR, 2020, pp. 1101–1112.

[47] A. S. Morgan, K. Hang, B. Wen, K. Bekris, and A. M. Dollar, "Complex in-hand manipulation via compliance-enabled finger gaiting and multi-modal planning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4821–4828, 2022.

[48] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam *et al.*, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," in *International Conference on Robotics and Automation (ICRA)*, 2023.

[49] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, "Visual dexterity: In-hand dexterous manipulation from depth," in *Icml workshop on new frontiers in learning, control, and dynamical systems*, 2023.

[50] 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 *7th Annual Conference on Robot Learning*, 2023.

[51] P. Jenmalm and R. S. Johansson, "Visual and somatosensory information about object shape control manipulative fingertip forces," *Journal of Neuroscience*, vol. 17, no. 11, pp. 4486–4499, 1997.

[52] Y. Chen, A. Sipos, M. Van der Merwe, and N. Fazeli, "Visuo-tactile transformers for manipulation," *arXiv preprint arXiv:2210.00121*, 2022.

[53] P. Falco, S. Lu, A. Cirillo, C. Natale, S. Pirozzi, and D. Lee, "Cross-modal visuo-tactile object recognition using robotic active exploration," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5273–5280.

[54] S. Wang, J. Wu, X. Sun, W. Yuan, W. T. Freeman, J. B. Tenenbaum, and E. H. Adelson, "3d shape perception from monocular vision, touch, and shape priors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1606–1613.

[55] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[56] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, “More than a feeling: Learning to grasp and regrasp using vision and touch,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.

[57] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, “Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8943–8950.

[58] R. Bischoff and V. Graefe, “Integrating vision, touch and natural language in the control of a situation-oriented behavior-based humanoid robot,” in *IEEE SMC’99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*, vol. 2. IEEE, 1999, pp. 999–1004.

[59] A. Petrovskaya and O. Khatib, “Global localization of objects via touch,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 569–585, 2011.

[60] D. Driess, P. Englert, and M. Toussaint, “Active learning with query paths for tactile object shape exploration,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[61] J. Liang, A. Handa, K. Van Wyk, V. Makoviychuk, O. Kroemer, and D. Fox, “In-hand object pose tracking via contact feedback and gpu-accelerated robotic simulation,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6203–6209.

[62] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer *et al.*, “Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3838–3845, 2020.

[63] W. Yuan, S. Dong, and E. H. Adelson, “Gelsight: High-resolution robot tactile sensors for estimating geometry and force,” *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[64] J. Hansen, F. Hogan, D. Rivkin, D. Meger, M. Jenkin, and G. Dudek, “Visuotactile-rl: learning multimodal manipulation policies with deep reinforcement learning,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8298–8304.

# A More Details on RL Formulation

**State.** The state of the system consists of the joint position $q_t \in \mathbb{R}^{16}$ of the Allegro hand, the binary tactile signal $o_t \in \{0,1\}^{16}$, the rotation axis $k \in \mathbb{S}^2$, the previous position target $\hat{q}_t \in \mathbb{R}^{16}$, the camera point cloud $P_t^c \in \mathbb{R}^{N_c \times 3}$, the augmented point cloud $P_t^a \in \mathbb{R}^{N_a \times 3}$, and the tactile point cloud $P_t^{touch} \in \mathbb{R}^{N_a \times 3}$.

**Action.** At each step, the action provided by the policy network is a relative control command $a_t \in \mathbb{R}^{16}$ and a PD controller drives the robot hand to approach $\hat{q}_{t+1} = \hat{q}_t + \hat{a}_t$. Note that we employ an exponential moving average in our implementation, i.e., $\hat{a}_t = \eta a_t + (1-\eta)\hat{a}_{t-1}, t \geq 1$ and let $\hat{a}_0 = 0$. We set $\eta = 0.8$ in our experiments.

**Reward.** We design a reward function for robust and transferable in-hand rotation, which is a weighted composition of several components:

$$r_t = c_1 r_{rot} + c_2 r_{vel} + c_3 r_{dist} + c_4 r_{torq} + c_5 r_{work} + c_6 r_{ctrl}. \tag{1}$$

$r_{rot}$ rewards the object's rotation angle. $r_{vel}$ penalizes the object's linear velocity to discourage motions that translate the object. $r_{dist}$ is a decreasing function regarding the distance between the object and the fingertips, encouraging the fingers to approach the object in hand and interact with it. $r_{torq}$ penalizes large torques, $r_{work}$ penalizes the work of the controller, and $r_{ctrl}$ penalizes the control error between command targets and real robot motion. We additionally implement a large penalty when the object falls off the hand. $c_1, c_2, \cdots, c_6$ are tuned hyper-parameters.

# B More Details on Teacher Student Training.

Learning the controller $\pi$ using RL is data inefficient when the observation is high-dimensional, e.g., point clouds, because the policy is required to extract essential information from high-dimensional observations and discern which actions are high-rewarding. To mitigate this issue, we employ a teacher-student learning approach to obviate training vision policies with RL, as shown in Figure 2.

**Teacher policy training.** We first use the proximal policy optimization (PPO)(5) to train teacher policies with low-dimensional states. Its input consists of the joint position of the Allegro hand $q_t$, the binary tactile signal $o_t$, the rotation axis $k$, the previous position target $\hat{q}_t$, the object's position $x_t \in \mathbb{R}^3$, its velocity $v_t \in \mathbb{R}^3$, its angular velocity $w_t \in \mathbb{R}^3$, and the object's shape feature embedding $f \in \mathbb{R}^{32}$. For tasks that require generalizability over multiple objects, we encode the shape information via a pre-trained PointNet(6) encoder in (7). Note that for each object, the shape feature embedding remains the same throughout the training. Given the state information, we use a Multi-Layer Perceptron (MLP) for both policy and value networks. We stack the current state with 3 historical states as input for better perception.

**Student policy training.** After using RL to train the teacher policy, we distill it to a student policy with visuotactile input. Concretely, its input includes the joint position $q_t$, the binary tactile signal $o_t$, the rotation axis $k$, and the previous position target $\hat{q}_t$. We stack it with 3 historical states. For the visual observation, the input includes camera point cloud $P_t^c$, augmented point cloud $P_t^a$, and the proposed tactile point cloud $P_t^{touch}$. We attach a one-hot vector to each point and concatenate them together as $P_t$.

We use PointNet(6) as the point cloud encoder and feed the latent vector and other inputs into an MLP. We adopt a two-stage distillation pipeline: We first collect a teacher dataset $\mathcal{D}$ of $5120k$ transitions and use Behavior Cloning (BC) to pre-train our student policy network; in the second stage, we use Dataset Aggregation (DAgger)(8) to fine-tune the network for more robust behavior.

# C Real Robot Setup

As is shown in Figure 3, our hardware setup consists of an XArm6 robot arm and a 16-DOF Allegro Hand with a depth camera. We attach 16 Force-Sensing Resistors (FSR) as tactile sensors to the palm and finger links of the robot hand as suggested by (9). We gather the contact signal from each sensor, then binarize the measurement according to a predetermined threshold $\theta_{th}$ to abridge the Sim-to-Real gap. We use Isaac Gym(10) as the rigid body physics simulator. The simulation setup is visualized in Figure 1. We simulate 16 contact sensors and calculate binary contact signals in the same way as in real.

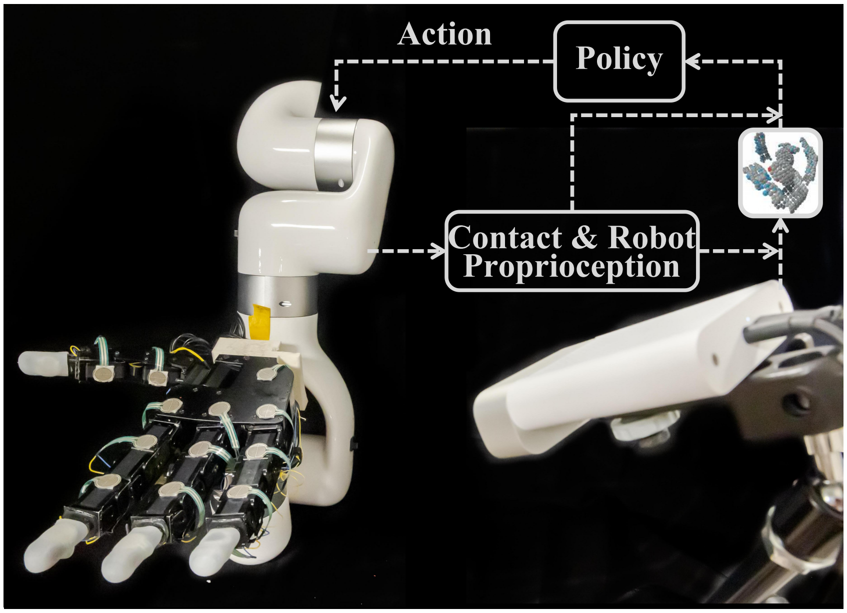
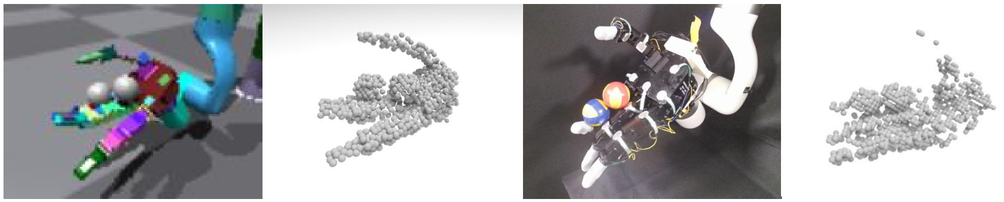


Figure 3: **Real-World Setup.** We use an Allegro Hand attached with 16 Force-Sensing Resistors. A Microsoft Azure Kinect camera is placed facing forward the robot.



Sim Image Sim Point Cloud Real Image Real Point Cloud

Figure 4: **Point Cloud Visualization in Sim and Real.** The Sim-to-Real gap is notably larger for RGB images compared to point clouds, leading us to select point clouds as the visual observation for our policy.

To obtain visual observations, we place a Microsoft Azure Kinect camera beside the hand in both real and simulated settings. We then generate the point cloud using the depth image. As illustrated in Figure 4, the point clouds in simulation closely mirror those in reality, especially compared with RGB images. We create an augmented point cloud (3) from the robot's proprioception to model the spatial relationship between the hand and the object, by sampling on the robot's mesh at the current pose. To differentiate between the camera-generated point cloud and the augmented one, we append a one-hot vector to each point. Point cloud visualizations are shown in Figure 2. The control frequency remains consistent at 10Hz in both simulated and real environments.

## D Experiemental Setup

**Evaluation Metric.** To evaluate the performance of a trained policy, we use the following metrics as suggested by (11).

1) **Cumulative Rotation Reward (CRR)** is the reward our agent obtains in an episode. We use it to evaluate the rotation capability of a policy in the simulation.
2) **Cumulative Rotation Angle (CRA)** is the angle (by rounds) the object rotates along the axis in an episode. We use it to evaluate a policy in the real.
3) **Time-to-Fall (TTF/Duration)** is the length of an episode (by seconds). TTF varies when the object falls off before the maximal episode length.

**Baselines.** In the experiments, we mainly compare our approach to the following baselines to study how much benefit visual and tactile modalities provide in training.

1) **No-Tactile.** Given a teacher policy, we train the student policy with no tactile information, where the robot can perceive neither the point cloud-based tactile representation nor the state-based binary contact. The only way to infer the object's spatial information is through vision.
2) **No-Synesthesia.** Given a teacher policy, we train the student policy with no tactile synesthesia. The robot only perceives the camera and augmented point clouds as visual input. For contact information, the policy takes only the state-based binary contact as input and has to infer specific positions of the contacts implicitly based on visual perceptions.
3) **No-Pointcloud.** Given a teacher policy, we train the student policy with no visual information. The policy only takes robot proprioception and binary contact as input.

We also compare our approach to single-stage baselines to substantiate the use of a teacher-student pipeline.

1) **Vision RL.** We train a visuotactile-based policy from scratch using RL. The policy observes robot proprioception, binary contact, and all the point clouds.
2) **Partial-State(PS).** We train a non-visual policy via RL for real deployment as in(9). Since the object's ground-truth pose and shape embedding are not available in real, the policy only observes robot proprioception and binary contact.

## E Real World Results

We transfer the visuotactile policies to the real robot without any fine-tuning and test whether visual policies continue to provide benefits. The results are shown in Table 3. Although visual policy might show comparable performance for simple geometry in simulation, the advantage of integrating vision and touch becomes more significant when deployed to the real world. **These results highlight the low domain gap of our proposed tactile point cloud representation.** A rudimentary concatenation of tactile signals and the extracted features of point clouds could increase the challenge for the policy to comprehend their underlying relationship. In contrast, our proposed visual-tactile synesthesia approach generally offers benefits. We also observe that visual policies tend to operate more cautiously, making occasional adjustments to nudge the object back to the palm center, while policies lacking visual perception tend to execute an almost fixed sequence of motion, irrespective of the object's deviation or instances of it becoming stuck.

Table 3: Evaluation of policies (CRA/TTF) in the real-world deployment. The above two lines refer to non-visual methods and the rest are visual policies. Each policy is tested for 5 episodes. Each trial lasts 60 seconds.

| Obs Type<br>(CRA/TTF) | 4-way Wrench | Double Balls | Multi-Object (x-axis) | Multi-Object (y-axis) | Multi-Object (z-axis) |
|---|---|---|---|---|---|
| Non-visual RL | 0.25/60.0 | 0.2/28.6 | 0.35/60.0 | 1.0/60.0 | 8.6/60.0 |
| Touch | 0.25/60.0 | 8.7/11.3 | 0.7/60.0 | 0.2/60.0 | 7.4/60.0 |
| Cam+Aug | 0.25/60.0 | 8.3/11.5 | 0.25/60.0 | 1.0/33.3 | 5.1/60.0 |
| Touch+Cam+Aug | 0.25/60.0 | 7.3/10.2 | 0.5/60.0 | **1.4/28.3** | 5.1/57.1 |
| Touch+Cam+Aug+Syn | **1.5/43.0** | **11.9/17.0** | **2.1/26.6** | 0.9/29.3 | **10.2/60.0** |

## F Qualitative Analysis: Visualization of PointNet intermediates

In PointNet, the input point cloud is fed into a local MLP extracting features of each point before a Max Pooling layer over points in each dimension of the features. Thus, PointNet is trained to implicitly select no more than $c_{out}$ points for representation learning, where $c_{out}$ is the output dimension of PointNet. We visualize in Figure 5 the points selected by our policy during evaluation. Interestingly, we find that our policy uses 42.7% of tactile-based points on average and the rest points are mainly from the tips or edges of fingers and the palm. This indicates that the point cloud encoder can extract meaningful features based on our visual-tactile synesthesia design.

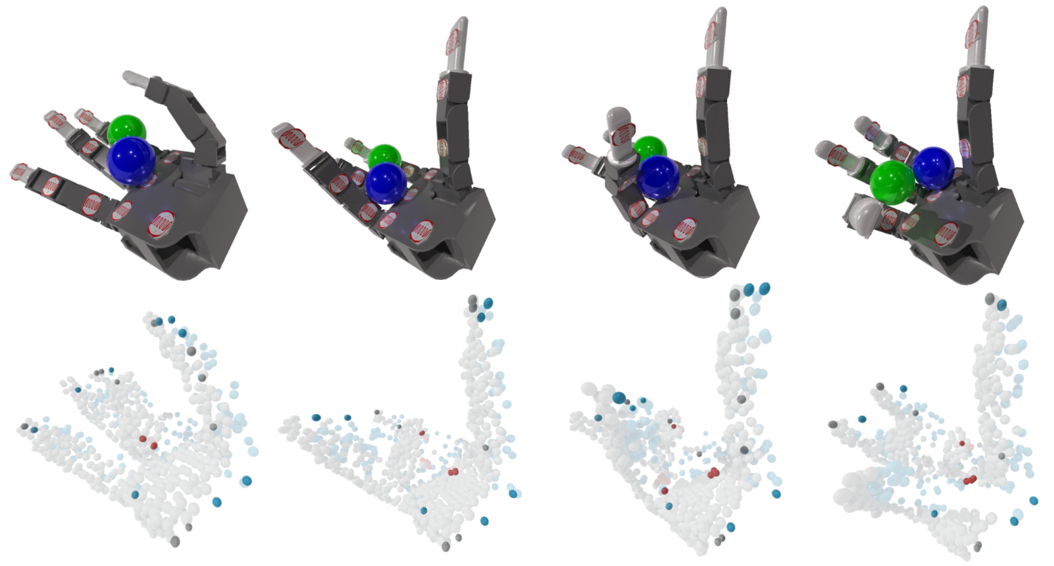


Figure 5: Visualization of selected point clouds (foreground) among observed point clouds (background) during evaluation. Red points belong to the proposed tactile point cloud.

## G Related Work

**Dexterous Manipulation** presents a wealth of opportunities for broad applications (12; 13; 14; 15; 16; 17). It enables the execution of intricate manipulative tasks, such as sliding (18; 19), rolling (20; 21; 22; 23; 24), pivoting (25; 26), and regrasping (27; 28; 29). Earlier methods addressing dexterous manipulation were grounded in classical control (30; 31). However, these methods rely on expert-engineered dynamics models, which restricts their utility for more complex tasks. To overcome these limitations, recent research has leveraged deep model-free RL for dexterous manipulation (32; 33; 34; 35; 36). Further enriching these advancements, imitation learning and combining common RL with demonstrations has led to higher sample efficiency and more natural manipulation behaviors (37; 38; 39; 40; 7; 41; 42; 43; 44). Dexterous in-hand manipulation has been a focal point of research interest recently (45; 46; 47; 11; 9; 48). To generalize to new objects, researchers have explored different sensors to capture object geometry and dynamic properties. Qi *et al.*(11) demonstrated that policy could infer object position and physical properties using proprioceptive history. But without explicit object information, it was only effective for z-axis rotation tasks. Yin *et al.* (9) proposed integrating binary tactile signals with proprioception for this task. However, the tactile signal was too sparse to capture detailed geometric attributes and thus could not handle objects with non-convex shapes. Chen (49) utilized depth information to aid object rotation but relied on an external pose estimation module. Most similar to us, a recent study (50) utilizes RGB images from optical tactile sensors and depth images for in-hand rotation. However, it necessitated continuous contact between the object and tactile sensors, constraining it to smaller objects that can be rotated on the fingertips. In contrast, our work does not impose any specific requirements on the object's initial location and can handle objects of diverse shapes and sizes. Furthermore, our tactile point cloud representation provides explicit 3D information about the object and tactile sensors' location, while their models are based on 2D images. As a result, our policy can solve tasks requiring more complex 3D spatial reasoning, such as rotating two balls simultaneously.

**Visuotactile Manipulation**, the integration of visual and tactile modalities, is a fundamental mechanism for human interaction with the environment (51), which presents significant potential for enhancing robot manipulation capabilities (52; 53; 54; 55). The visual modality offers a comprehensive, non-contact perspective of the environment, while the tactile modality complements this by offering detailed, contact-dependent properties such as texture, temperature, hardness, and weight. The key to integrating these modalities in robotic manipulation lies in two aspects: (i) the representation of each modality, and (ii) the strategy employed to fuse these modalities. In the visual modality, RGB images are a common choice due to their widespread availability (53; 56). But these images do not inherently capture distance information, which is often critical in manipulation tasks. To address this limitation, researchers (50; 49) proposed using depth to handle more contact-intensive tasks such as in-hand rotation. Different from these methods, our approach utilizes the point cloud captured by a camera, inherently incorporating 3D information into the visual representation. For the tactile modality, raw sensor readings are a natural choice (57; 58). However, for a smaller sim-to-real gap, the binary contact vector has been used (9; 59; 60; 61). Notably, vision-based tactile sensors, such as DIGIT (62) and GelSighT (63), can also encode tactile information into RGB images. Another critical consideration is the design of a multi-modal learning paradigm. Most existing approaches favor combining modalities at the feature level, where separate feature extractors are trained for each modality, and the predicted features are concatenated as a multi-modal representation (57; 64). When utilizing optical tactile sensors with tactile images, the combination can also be performed at the input level, as both vision and touch are represented as RGB images (54; 56). In contrast, our method represents tactile data as a point cloud and merges with the camera point cloud at the input level. This approach treats the combined visual and tactile data as a single input to the policy network. This innovative design, which we term Robot Synesthesia, enriches the contextual understanding for both modalities and encodes the sensory data into a cohesive 3D space.