Analyzing Key Objectives in Human-to-Robot Retargeting for Dexterous Manipulation

Chendong Xin*, Mingrui Yu*, Yongpeng Jiang, Zhefeng Zhang, and Xiang Li†, *Senior Member, IEEE*

***Abstract*—Kinematic retargeting from human hands to robot hands is essential for transferring dexterity from humans to robots in manipulation teleoperation and imitation learning. However, due to mechanical differences between human and robot hands, completely reproducing human motions on robot hands is impossible. Existing works on retargeting incorporate various optimization objectives, focusing on different aspects of hand configuration. However, the lack of experimental comparative studies leaves the significance and effectiveness of these objectives unclear. This work aims to analyze these retargeting objectives for dexterous manipulation through extensive real-world comparative experiments. Specifically, we propose a comprehensive retargeting objective formulation that integrates intuitively crucial factors appearing in recent approaches. The significance of each factor is evaluated through experimental ablation studies on the full objective in kinematic posture retargeting and real-world teleoperated manipulation tasks. Experimental results and conclusions provide valuable insights for designing more accurate and effective retargeting algorithms for real-world dexterous manipulation. Supplementary materials are available at https://mingrui-yu.github.io/retargeting.**

***Index Terms*—Dexterous manipulation, multi-fingered hand, human-to-robot retargeting, teleoperation.**

I. Introduction

TRANSFERRING dexterity from humans to robots is a promising field in dexterous manipulation research, as the complexity of dexterous manipulation tasks poses challenges for classical analytical approaches [1]–[4]. In either collecting robot demonstrations through human teleoperation or learning from offline human manipulations, an essential component is *kinematic retargeting*, which refers to kinematically translating the human hand configuration to robot hand joint positions. One key challenge of retargeting is that, due to the differences in morphology and degrees of freedom (DoFs) between human hands and current robot hands, the human hand configurations can not be exactly reproduced by the robot hand. Consequently, it is inevitable to focus on only partial configuration of the human hand that are more crucial to manipulation tasks when designing the retargeting algorithms.

The most straightforward retargeting approach is direct joint-to-joint mapping, where the robot hand joints follow a manually defined transformation of the human hand joints [5]–[12]. These approaches require considerable manual efforts to define the joint-space transformation for a certain robot hand. Task-space retargeting is used more widely, which relies on inverse kinematics (IK) solving to determine robot hand joint configuration based on human hand keypoints. The global fingertip position is the fundamental retargeting objective to align human and robot hand in task-space [13]–[17]. Most works use vectors from wrist to fingertips as the retargeting objective and minimize their discrepancy between human and robot hands to preserve overall hand shape [18]–[29]. Some works further incorporate fingertip orientation into the objective, using vectors from the palm to the middle phalanx to represent finger bending information [19], or using the directions of proximal phalanges and distal phalanges [30] to represent orientations of the finger-roots and fingertips [31]. Additionally, relative position among fingertips can be included to address the challenge of pinch grasps [28], in which a switching weight is used to prioritize the fingers involved in pinching, and the corresponding reference distance on human hand is adjusted to a minimal distance to encourage fingertip contact.

These retargeting approaches have been deployed in real-world manipulation, but they differ in their choices of objectives. Due to the lack of comparative studies, it remains unclear which objectives are more dispensable for which types of manipulation tasks and whether conflicts exist between them. A previous survey [32] reviews and classifies the common retargeting methods, but it does not focus on retargeting objectives and not provide experimental evaluations.

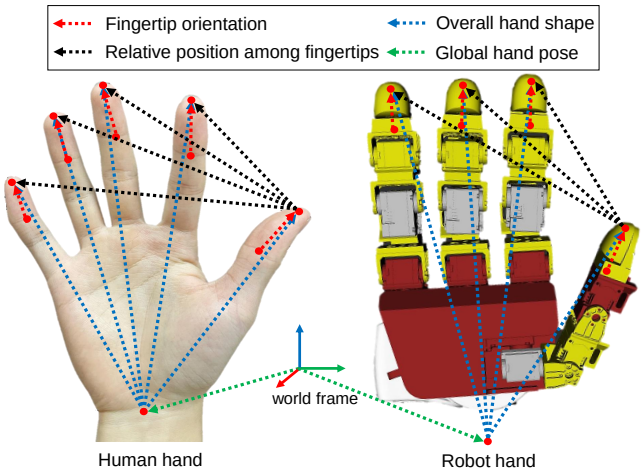


Fig. 1. Crucial objectives in human-to-robot retargeting for dexterous manipulation. This work explores the appropriate formulation of these objectives and experimentally analyzes their significance for different manipulation tasks.

*Equal contribution.
†Corresponding author: xiangli@tsinghua.edu.cn.
The authors are with the Department of Automation, Beijing National Research Center for Information Science and Technology, Tsinghua University, China. This work was supported in part by Science and Technology Innovation 2030-Key Project under Grant 2021ZD0201404, in part by the National Natural Science Foundation of China under Grant 62461160307 and 623B2059, in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM208, in part by the BNRist project under Grant BNR2024TD03003, and in part by the Institute for Guo Qiang, Tsinghua University.

This work aims to analyze the key objectives in retargeting for dexterous manipulation through experimental comparison. We summarize objectives that are used in recent research on dexterous retargeting and propose a comprehensive retargeting objective formulation that considers all intuitively crucial factors illustrated in Fig. 1. By conducting ablation studies on the full objective in kinematic posture retargeting and real-world manipulation teleoperation, we analyze the significance and effectiveness of different objectives in different tasks. The experimental results and conclusions provide valuable insights for future research on designing retargeting algorithms for learning dexterous manipulation from humans or teleoperation.

II. Method

We intuitively list the factors that are potentially crucial to human-to-robot retargeting for dexterous manipulation:

- **Global hand pose**: the robot's hand pose in the global frame should be aligned with the human's in dexterous manipulation involving large arm motions.
- **Overall hand shape**: the robot hand should replicate a similar overall hand shape to the human hand to achieve similar postures.
- **Relative position among fingertips**: the spatial relationship among each fingertip (e.g., thumb and index) is critical for manipulation tasks that require precise coordination of fingertips.
- **Fingertip orientations**: the accurate fingertip orientations ensure appropriate directions of contact normals during contact-rich dexterous manipulation.

Based on the intuitive factors above, we formulate their corresponding mathematical representations and construct a complete retargeting objective.

Global hand pose: The global hand pose is typically formulated with a wrist position term $\mathcal{L}_{\text{wrist\_pos}}$ and a wrist orientation term $\mathcal{L}_{\text{wrist\_rot}}$. However, accurate tracking of global wrist pose is not necessary in most dexterous manipulation tasks and may reduce accuracy in fingertip positions. Due to different hand morphologies between the human and robot, the fingertip positions relative to the wrist may conflict with the relative positions between fingertips and fingertip orientations. As a result, it can be better to allow adjustment of the wrist pose in exchange for higher fingertip accuracy. Specifically, we replace the wrist position objective with a thumb-tip position objective $\mathcal{L}_{\text{thumb\_pos}}$ and apply joint optimization of the arm-hand joint positions. The retargeted wrist orientation is regularized through the wrist orientation objective $\mathcal{L}_{\text{wrist\_rot}}$ with a relatively small weight. The complete term is formulated as:

$$\mathcal{L}_{\text{hand\_pose}} = \left\| \boldsymbol{p}^h_{\text{thumb}} - \boldsymbol{p}^r_{\text{thumb}} \right\|_2^2 + \beta_{\text{rot}} \operatorname{angle}(\boldsymbol{q}^h_{\text{wrist}}, \boldsymbol{q}^r_{\text{wrist}}), \quad (1)$$

where $\boldsymbol{p}_{\text{thumb}}$ is the thumb fingertip position of human and robot, and $\boldsymbol{q}_{\text{wrist}}$ is the orientation of human and robot wrist.

Overall hand shape: The overall hand shape is represented by the fingertip positions relative to the wrist position. The fingertip position term $\mathcal{L}_{\text{fingertip\_pos}}$ measures the difference in a set of vectors defined from the wrist to the fingertips of the robot and human hand:

$$\mathcal{L}_{\text{fingertip\_pos}} = \sum_{i=1}^{N} \left\| \boldsymbol{v}^h_i - \boldsymbol{v}^r_i \right\|_2^2, \quad (2)$$

where $\boldsymbol{v}_i$ is the vector from the wrist to the i^{th} fingertip on human and robot hand, and N is the number of fingers. Note that this term needs to be coordinated with the following pinch objective, for which the details are provided in Appendix.

Relative position among fingertips: We use vectors from the thumb to primary fingers (index, middle, ring) to represent the relative positions among fingertips, which is crucial for pinching. We adopt a similar formulation to DexPilot [28] with a switching weight function $s(d_i)$ and a distance rescaling function $l(d_i)$:

$$\mathcal{L}_{\text{pinch}} = \sum_{i=1}^{N-1} s(d_i) \left\| \boldsymbol{\gamma}^r_i - l(d_i) \hat{\boldsymbol{\gamma}}^h_i \right\|_2^2, \quad (3)$$

where $\boldsymbol{\gamma}_i$ is the vector from the thumb fingertip to the fingertip of the i^{th} primary finger, $d_i = \left\| \boldsymbol{\gamma}^h_i \right\|$ and $\hat{\boldsymbol{\gamma}}_i = \frac{\boldsymbol{\gamma}_i}{d_i}$. We use a continuous weight function $s(d_i) = \frac{1}{1+e^{10(d_i-\epsilon_1)}}$ to ensure smooth transitions. The distance rescaling function

$$l(d_i) = \begin{cases} 0, & d_i < \epsilon_2 \\ \frac{\epsilon_1}{\epsilon_1 - \epsilon_2}(d_i - \epsilon_2), & \epsilon_2 \le d_i \le \epsilon_1 \\ d_i, & d_i > \epsilon_1, \end{cases} \quad (4)$$

linearly rescales human fingertip distances from $[\epsilon_2, \epsilon_1]$ to $[0, \epsilon_1]$ to ensure a continuous transition within the pinching range and avoid sudden changes around the threshold ϵ_1.

Fingertip orientations: To represent fingertip orientations, we include another set of vectors. In contrast with methods such as DexMV [19] that define the vectors from the wrist to the middle phalanx, our formulation defines the vectors from the distal interphalangeal (DIP) joints to the fingertips, which represent fingertip orientations more directly:

$$\mathcal{L}_{\text{fingertip\_rot}} = \sum_{i=1}^{N} \left\| \boldsymbol{r}^h_i - \boldsymbol{r}^r_i \right\|_2^2, \quad (5)$$

where $\boldsymbol{r}_i$ is the vector from the DIP joint to the fingertip.

The final retargeting optimization objective considering all the above factors can be specified as:

$$\begin{aligned} \mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{thumb\_pos}} + \lambda_2 \mathcal{L}_{\text{wrist\_rot}} + \lambda_3 \mathcal{L}_{\text{fingertip\_pos}} + \\ \lambda_4 \mathcal{L}_{\text{fingertip\_rot}} + \lambda_5 \mathcal{L}_{\text{pinch}} + \mathcal{L}_{\text{joint}} + \mathcal{L}_{\text{vel}}, \end{aligned} \quad (6)$$

where two joint-space regularization terms are additionally considered. The joint position regularization $\mathcal{L}_{\text{joint}} = \sum_{j=1}^{m} w^{\text{pos}}_j \left\| q_j - \bar{q}_j \right\|_2^2$ penalizes the deviation of some joints from a pre-defined joint configuration $\bar{\boldsymbol{q}}$. The joint velocity regularization $\mathcal{L}_{\text{vel}} = \sum_{j=1}^{m} w^{\text{vel}}_j \left\| q_j - q^{\text{prev}}_j \right\|_2^2$ penalizes large changes in joint positions compared to previous timestep to encourage trajectory smoothness. Its effect is illustrated by the joint position/velocity/acceleration profiles in Appendix.

III. Evaluations and Results

A. Evaluation Setup

Implementation: Simulation studies involve a Leap Hand [33] and a Shadow Hand, both mounted on a Franka Emika Panda arm. Real-world experiments are conducted on a teleoperation system with a Leap Hand, a Panda arm, and an Apple

TABLE I
DEFINITION OF THE ABLATIONS FOR COMPARATIVE STUDIES.

Category	Ablation	Definition
Full	Full	Complete retargeting objective $\mathcal{L}_{\text{total}}$ with all terms (6)
Fingertip pinch	A1	Remove the pinch term $\mathcal{L}_{\text{pinch}}$ (7)
	A2	Use actual pinch distance without distance rescaling in (7)
Fingertip orientation	A3	Remove the fingertip orientation term $\mathcal{L}_{\text{fingertip\_rot}}$ (5)
	A4	Replace the vectors from DIP joints to fingertips with vectors from the wrist to DIP joints in (5)
Global wrist pose	A5	Replace the thumb position term in (1) with a wrist position term
	A6	Replace the thumb position term in (1) with a wrist position term and remove $\mathcal{L}_{\text{pinch}}$ and $\mathcal{L}_{\text{fingertip\_rot}}$
Joint regularization	A7	Remove the joint position regularization $\mathcal{L}_{\text{joint}}$
	A8	Replace the thumb position term in (1) with a wrist position term and remove $\mathcal{L}_{\text{pinch}}$, $\mathcal{L}_{\text{fingertip\_rot}}$, and $\mathcal{L}_{\text{joint}}$

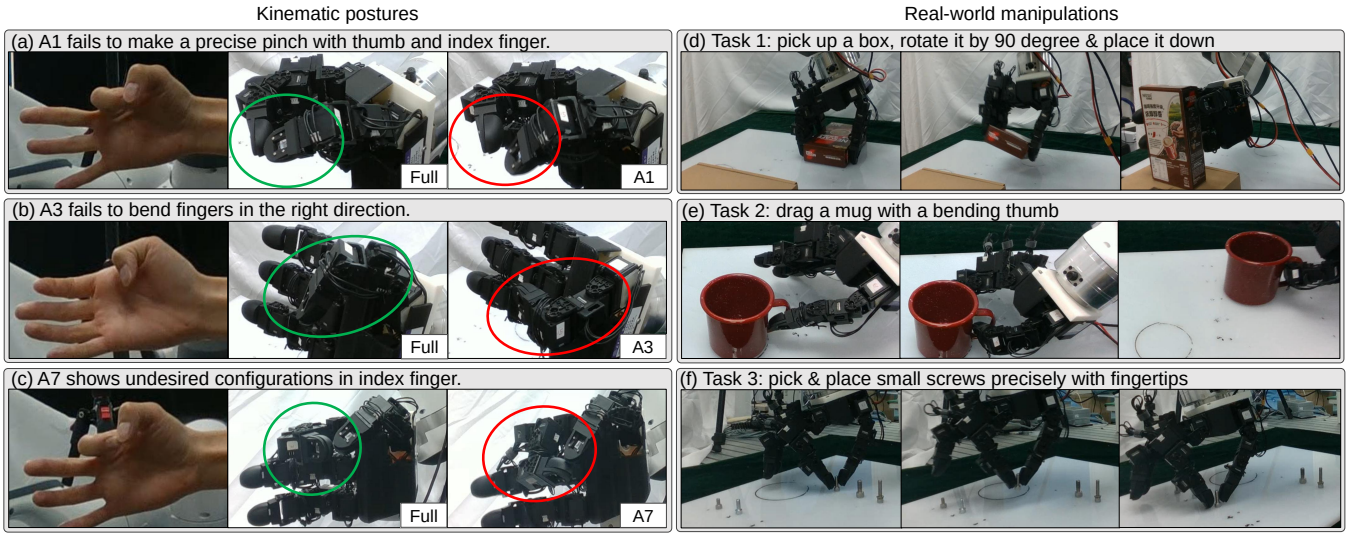


Fig. 2. Snapshots of the real-world kinematic postures retargeting (left) and the three manipulation tasks (right). (a) to (c): each row shows a human hand posture and the corresponding retargeted robot postures using the full objective and an ablation implementation. (d) to (e): each row shows the snapshots of the manipulation process of one task using the full retargeting objective.

Vision Pro for human hand detection [34]. The retargeting objective is optimized in real time using the Sequential Least-Squares Quadratic Programming in the NLopt library [35]. Implementation details such as control approaches, runtime and latency, and robustness are further discussed in Appendix.

Ablation setup: To analyze the significance of each retargeting objective, we implement ablations by removing or changing certain objective terms in $\mathcal{L}_{\text{total}}$ (6). The detail of the ablation setup is described in Table I.

Kinematic posture retargeting tasks: We evaluate the full retargeting objective and eight ablations across offline data of three pinch motions involving the thumb and the index, middle, and ring fingers, respectively. We use four quantitative metrics for kinematic postures: 1) average fingertip position error in the global frame, 2) average fingertip position error relative to the wrist, 3) average fingertip position error relative to the thumb, and 4) average fingertip orientation error. The results on Leap Hand are shown in Fig. 3. The results on another trajectory involving finger crossing motions and results on Shadow Hand are provided in the Appendix. The snapshots of posture retargeting in the real world are shown in Fig. 2.

Real-world manipulation tasks: We design three representative real-world manipulation teleoperation tasks :

1) Pick up a box, rotate it, and place it down, representing common pick-and-place tasks in dexterous teleoperation.
2) Drag a mug through its handle using the bent thumb, where fingertip orientation plays a decisive role.
3) Pick up five different upright-standing screws and place them vertically, where precise pinches are important.

For each task, we sequentially evaluate the full retargeting objective and the seven ablations (A1 to A7), and repeat the entire sequence three times. For task 1 and 2, we use mean completion time to assess performance of pilots, as all trials are successful. For task 3, we use success rate for comparison. The manipulation results are summarized in Fig. 4. Our real-world evaluation is a preliminary case study with only two human pilots. We plan to involve more pilots to capture variability among users for further validation.

B. Analysis of Results

We analyze the results of kinematic posture retargeting and real-world manipulations in *four perspectives* corresponding to the designed objectives.

Fingertip pinch: A1 and A2 are the ablation of fingertip pinch term. The results show that: 1) removing the pinch term in A1 results in significantly higher errors in both fingertip position and relative position to the thumb (A1 in Fig. 3). This results from the potential conflict between the global fingertip

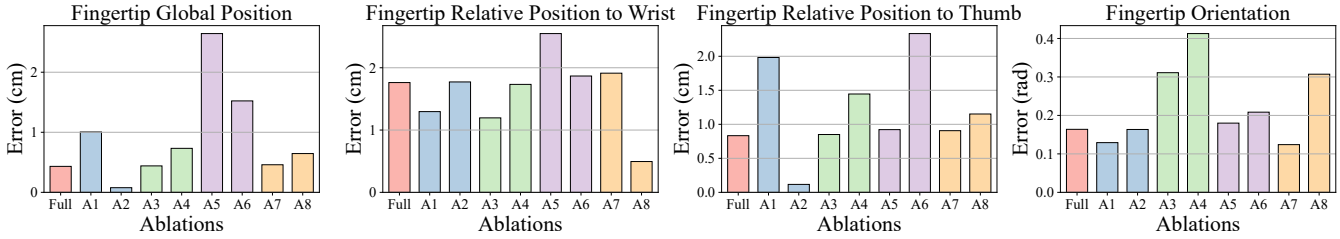


Fig. 3. Results of kinematic posture retargeting on finger pinch trajectories. Each bar shows the error of one ablation implementation and the colors represent the ablation category defined in Table I. For the metrics of fingertip global position and fingertip relative position to the thumb, only the errors of the two fingers involved in the pinching motion are considered.

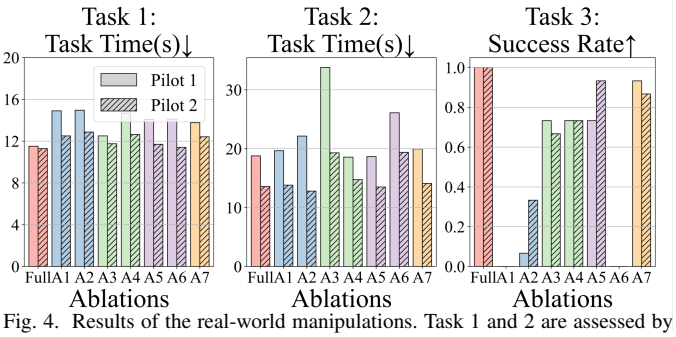


Fig. 4. Results of the real-world manipulations. Task 1 and 2 are assessed by task time, while Task 3 is evaluated by success rate. Each pair of bars shows the error of one ablation implementation and the colors represent the ablation category defined in Table I. The two pilots are distinguished by hatched bars.

position and the fingertip position relative to the wrist, due to different hand morphologies between the human and robot; 2) in real-world manipulations, removing the pinch term in A1 leads to failure in tasks involving pinch motions, as it cannot close the gap between thumb and index fingertips (A1 in Task 3, Fig. 4). This is also displayed in posture (a) in Fig. 2; 3) using actual pinch distances without rescaling in A2 results in minimal error in both metrics (A2 in Fig. 3), but could lead to low success rate in real-world manipulation tasks, primarily due to that using actual pinch distance is vulnerable to human finger tracking inaccuracy (A2 in Task 3, Fig. 4).

Fingertip orientations: A3 and A4 correspond to the ablation of fingertip orientation term. The results suggest: 1) DIP-to-tip vectors hold more explicit information of fingertip orientation than wrist-to-DIP vectors. Replacing the DIP-to-tip vectors with wrist-to-DIP vectors shows little advantage in fingertip orientation error over no orientation consideration (A4 in Fig. 3), as the DIP-to-tip vectors may be changed by fingertip relative position terms even though the wrist-to-DIP vectors are fixed; 2) in real-world manipulation Task 2, no consideration of fingertip orientations results in poor performance as the robot hand fails to replicate the finger bending motion of the human pilots to hook the mug handle (A3 in Task 2, Fig. 4). In addition, it may occur that the finger bends in a wrong direction like posture (b) in Fig. 2; and 3) the missing of fingertip orientation information can also have negative impacts on precise manipulation tasks such as task 3, as inaccuracy in orientation leads to undesired contact normals with the object (A3 in Task 3, Fig. 4).

Global hand pose: A5 and A6 are the ablation of global hand pose. The results show that: 1) determining the global hand pose by the thumb fingertip position term instead of wrist position term leads to a significantly lower error in fingertip global position (A5 in Fig. 3), primarily due to that using the thumb position term sacrifices in wrist position accuracy in exchange for better alignment of fingertips; 2) when using the exact wrist pose, removing the pinch term also leads to higher errors in fingertip relative position to the thumb (A6 in Fig. 3), and removing the fingertip orientation term hampers finger bending (A6 in Task 2 in Fig. 4); 3) in real-world tasks, the choice of thumb fingertip position term and wrist position term does not bring up much difference (A5 in Fig. 4), because human pilots can easily adjust the global hand position to eliminate the influence of fingertip global position inaccuracy.

Joint position regularization: A7 and A8 removes the joint position regularization term. The corresponding results suggest that: 1) from the comparison between full and A7 in Fig. 3, joint position regularization term seems to have little impact in both simulation and real-world manipulation. However, without this term, it may occur that the hand displays undesired joint configurations like posture (c) in Fig. 2. The joint position regularization makes the retargeting result more truthful without negative impacts; 2) the comparison of A8 and A6 shows that when using wrist position term for global hand pose, the joint position regularization term adds to the errors in fingertip global position and relative position to the wrist and the thumb (A8 in Fig. 3). In contrast, when using thumb fingertip position, the joint position regularization term has little impact on position errors.

IV. Conclusion

This work analyzes the significance of different objectives in human-to-robot retargeting for dexterous manipulation through both kinematic posture retargeting and three representative real-world manipulation tasks. The comprehensive results demonstrate that 1) the fingertip pinch objective is crucial for manipulation tasks involving precise fingertip coordination; 2) the fingertip orientation objective should be included for tasks sensitive to finger orientation rather than solely position; 3) allowing wrist pose adjustment instead of using exact human wrist pose benefits the accuracy of fingertip poses; 4) joint position regularization makes the retargeting postures appear more natural while having negligible negative impacts; and 5) all terms using the proposed formulation do not demonstrate conflicts and perform well in all tasks. We believe this study provides valuable insight for designing retargeting algorithms in future work on learning dexterous manipulation from humans or dexterous teleoperation. In future work, we plan to extend the real-world evaluation to more teleoperators and tasks to better capture user variability and validate the generality of the proposed retargeting formulation.

TABLE II
DESCRIPTIONS OF SUPPLEMENTARY MATERIALS

Supplementary material	**Description**
Project website	https://mingrui-yu.github.io/retargeting
Appendix	Supplementary results and implementation details are provided in Appendix, which includes the detailed formulation of the pinch objective, additional quantitative results of kinematic posture retargeting on another trajectory involving finger crossing motions and on the Shadow Hand, explanation of the joint velocity regularization term, comparison with existing retargeting approaches, and implementation details such as hyper-parameter settings, control approaches and discussion on runtime, latency, and robustness. The Appendix is also available on the project website.
Video	The video of the real-world experiments is available on the project website, which demonstrates the real-world kinematic posture retargeting, three real-world manipulation tasks for evaluation, and additional trials on manipulation tasks of higher complexity.
Source code	The code is open-sourced on the project website (GitHub), which includes the implementation of the retargeting algorithm and the evaluation on kinematic postures in simulation. We provide a detailed guideline for setting up everything and to launch the evaluation. Users are encouraged to report issues via GitHub.
Dataset	All human hand motion trajectories recorded by Apple Vision Pro in this study are provided on the project website. The format of the dataset is described by the instructions in the corresponding README file.
CAD files	The CAD files of the fingertips and the URDF of the whole robot (i.e., Panda arm + Leap Hand + fingertips) are provided on the project website.

REFERENCES

[1] T. Pang, H. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *IEEE Trans. Robot.*, 2023.

[2] Y. Jiang, M. Yu, X. Zhu, M. Tomizuka, and X. Li, "Contact-implicit model predictive control for dexterous in-hand manipulation: A long-horizon and robust approach," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 5260–5266.

[3] M. Yu, Y. Jiang, C. Chen, Y. Jia, and X. Li, "Robotic in-hand manipulation for large-range precise object movement: The rgmc champion solution," *IEEE Robotics and Automation Letters*, 2025.

[4] M. Yu, B. Liang, X. Zhang, X. Zhu, L. Sun, C. Wang, S. Song, X. Li, and M. Tomizuka, "In-hand following of deformable linear objects using dexterous fingers with tactile sensing," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 13 518–13 524.

[5] Q. Gao, J. Li, Y. Zhu, S. Wang, J. Liufu, and J. Liu, "Hand gesture teleoperation for dexterous manipulators in space station by using monocular hand motion capture," *Acta Astronautica*, vol. 204, pp. 630–639, 2023.

[6] J. Huang, K. Chen, J. Zhou, X. Lin, P. Abbeel, Q. Dou, and Y. Liu, "Dih-tele: Dexterous in-hand teleoperation framework for learning multiobjects manipulation with tactile sensing," *IEEE/ASME Transactions on Mechatronics*, 2025.

[7] W. Wei, B. Zhou, B. Fan, M. Du, G. Bao, and S. Cai, "An adaptive hand exoskeleton for teleoperation system," *Chinese Journal of Mechanical Engineering*, vol. 36, no. 1, p. 60, 2023.

[8] J. Guo, J. Luo, Z. Wei, Y. Hou, Z. Xu, X. Lin, C. Gao, and L. Shao, "Telephantom: A user-friendly teleoperation system with virtual assistance for enhanced effectiveness," *arXiv preprint arXiv:2412.13548*, 2024.

[9] M. V. Liarokapis, P. K. Artemiadis, and K. J. Kyriakopoulos, "Telemanipulation with the dlr/hit ii robot hand using a dataglove and a low cost force feedback device," in *21st Mediterranean Conference on Control and Automation*, 2013, pp. 431–436.

[10] G. Giudici, B. Omarali, A. A. Bonzini, K. Althoefer, I. Farkhatdinov, and L. Jamone, "Feeling good: Validation of bilateral tactile telemanipulation for a dexterous robot," in *Annual Conference Towards Autonomous Robotic Systems*. Springer, 2023, pp. 443–454.

[11] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto, "Holo-dex: Teaching dexterity with immersive mixed reality," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5962–5969.

[12] A. Iyer, Z. Peng, Y. Dai, I. Guzey, S. Haldar, S. Chintala, and L. Pinto, "Open teach: A versatile teleoperation system for robotic manipulation," *arXiv preprint arXiv:2403.07870*, 2024.

[13] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," in *Robotics: Science and Systems (RSS)*, 2024.

[14] K. Shaw, Y. Li, J. Yang, M. K. Srirama, R. Liu, H. Xiong, R. Mendonca, and D. Pathak, "Bimanual dexterity for complex tasks," *arXiv preprint arXiv:2411.13677*, 2024.

[15] H. Zhang, S. Hu, Z. Yuan, and H. Xu, "Doglove: Dexterous manipulation with a low-cost open-source haptic force feedback glove," *arXiv preprint arXiv:2502.07730*, 2025.

[16] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto, "Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation," in *2023 ieee international conference on robotics and automation (icra)*, 2023, pp. 5954–5961.

[17] S. Chen, C. Wang, K. Nguyen, L. Fei-Fei, and C. K. Liu, "Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback," *arXiv preprint arXiv:2410.08464*, 2024.

[18] D. Antotsiou, G. Garcia-Hernando, and T.-K. Kim, "Task-oriented hand motion retargeting for dexterous manipulation imitation," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[19] 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*. Springer, 2022, pp. 570–587.

[20] 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.

[21] R. Ding, Y. Qin, J. Zhu, C. Jia, S. Yang, R. Yang, X. Qi, and X. Wang, "Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning," *arXiv preprint arXiv:2407.03162*, 2024.

[22] S. Yang, M. Liu, Y. Qin, R. Ding, J. Li, X. Cheng, R. Yang, S. Yi, and X. Wang, "Ace: A cross-platform visual-exoskeletons system for low-cost dexterous teleoperation," *arXiv preprint arXiv:2408.11805*, 2024.

[23] S. Huang, Z. Zhang, M. Chen, Z. Wu, Q. Li, and Z. Ming, "Designing of a dexterous hand and performance evaluation based on teleoperation," in *2024 International Conference on Intelligent Robotics and Automatic Control (IRAC)*, 2024, pp. 169–172.

[24] K. Shaw, S. Bahl, A. Sivakumar, A. Kannan, and D. Pathak, "Learning dexterity from human hand motion in internet videos," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 513–532, 2024.

[25] H. Yuan, B. Zhou, Y. Fu, and Z. Lu, "Cross-embodiment dexterous grasping with reinforcement learning," *arXiv preprint arXiv:2410.02479*, 2024.

[26] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," *arXiv preprint arXiv:2407.01512*, 2024.

[27] B. Romero, H.-S. Fang, P. Agrawal, and E. Adelson, "Eyesight hand: Design of a fully-actuated dexterous robot hand with integrated vision-based tactile sensors and compliant actuation," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 1853–1860.

[28] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox, "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9164–9170.

[29] A. Sivakumar, K. Shaw, and D. Pathak, "Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube," in *Robotics: Science and Systems*, 2022.

[30] Wikipedia contributors, "Phalanx bone — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Phalanx_bone&oldid=1275529475, 2025, [Online; accessed 1-April-2025].

[31] S. Li, X. Ma, H. Liang, M. Görner, P. Ruppel, B. Fang, F. Sun, and J. Zhang, "Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 416–422.

[32] R. Meattini, R. Suarez, G. Palli, and C. Melchiorri, "Human to robot hand motion mapping methods: Review and classification," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 842–861, 2022.

[33] K. Shaw, A. Agarwal, and D. Pathak, "LEAP Hand: Low-cost, efficient, and anthropomorphic hand for robot learning," *Robotics: Science and Systems (RSS)*, 2023.

[34] Y. Park and P. Agrawal, "Using apple vision pro to train and control robots," 2024. [Online]. Available: https://github.com/Improbable-AI/VisionProTeleop

[35] S. G. Johnson, "The NLopt nonlinear-optimization package," https://github.com/stevengj/nlopt, 2007.

APPENDIX

ADDITIONAL DETAILS OF OBJECTIVE FORMULATION

Relative position among fingertips: The pinch term is formulated as:

$$\mathcal{L}_{\text{pinch}} = \sum_{i=1}^{N-1} s(d_i) \left\| \boldsymbol{\gamma}_i^r - l(d_i)\hat{\boldsymbol{\gamma}}_i^h \right\|^2, \tag{7}$$

where $\boldsymbol{\gamma}_i$ is the vector from the thumb fingertip to the fingertip of the i^{th} primary finger, $d_i = \left\|\boldsymbol{\gamma}_i^h\right\|$ and $\hat{\boldsymbol{\gamma}}_i^h = \frac{\boldsymbol{\gamma}_i^h}{d_i}$. Instead of using a discrete weight function as DexPilot, we use a continuous weight function

$$s(d_i) = \text{sigmoid}(d_i, \epsilon_1, 10),$$

where $\text{sigmoid}(\cdot)$ is the sigmoid function defined as follows:

$$\text{sigmoid}(\text{x}, \text{c}, \text{w}) = \frac{1}{1 + e^{w(x-c)}}.$$

Our distance rescaling function is defined as follows:

$$l(d_i) = \begin{cases} 0, & d_i < \epsilon_2 \\ \frac{\epsilon_1}{\epsilon_1 - \epsilon_2}(d_i - \epsilon_2), & \epsilon_2 \le d_i \le \epsilon_1 \\ d_i, & d_i > \epsilon_1, \end{cases} \tag{8}$$

where fingertip distance within pinching range $[\epsilon_2, \epsilon_1]$ is linearly rescaled into $[0, \epsilon_1]$. This ensures a continuous transition in the pinching range and avoids sudden changes around the threshold ϵ_1. In practice we set $\epsilon_1 = 1 \times 10^{-1}$ m and $\epsilon_2 = 1 \times 10^{-2}$ m.

Overall hand shape: To balance fingertip positions relative to the wrist and the thumb, we also set a switching weight for the fingertip position term $\mathcal{L}_{\text{fingertip\_pos}}$:

$$\mathcal{L}_{\text{fingertip\_pos}} = \sum_{i=1}^{N} \tilde{s}(d_i) \left\| \boldsymbol{v}_i^r - \boldsymbol{v}_i^h \right\|^2, \tag{9}$$

where

$$\tilde{s}(d_i) = \text{sigmoid}(d_i, \epsilon_1, -10),$$

so that the sum of $s(d_i)$ and $\tilde{s}(d_i)$ will be a fixed number. In ablation studies where the pinch term is removed (A1, A6 and A8), we set $\tilde{s}(d_i)$ to be a constant 1.0 as in (2).

ADDITIONAL KINEMATIC POSTURE RETARGETING RESULTS

Three sets of additional quantitative results of kinematic posture retargeting on another trajectory involving finger crossing motions and on the Shadow Hand are shown in Fig. 5. Similar conclusions to the main text can be derived.

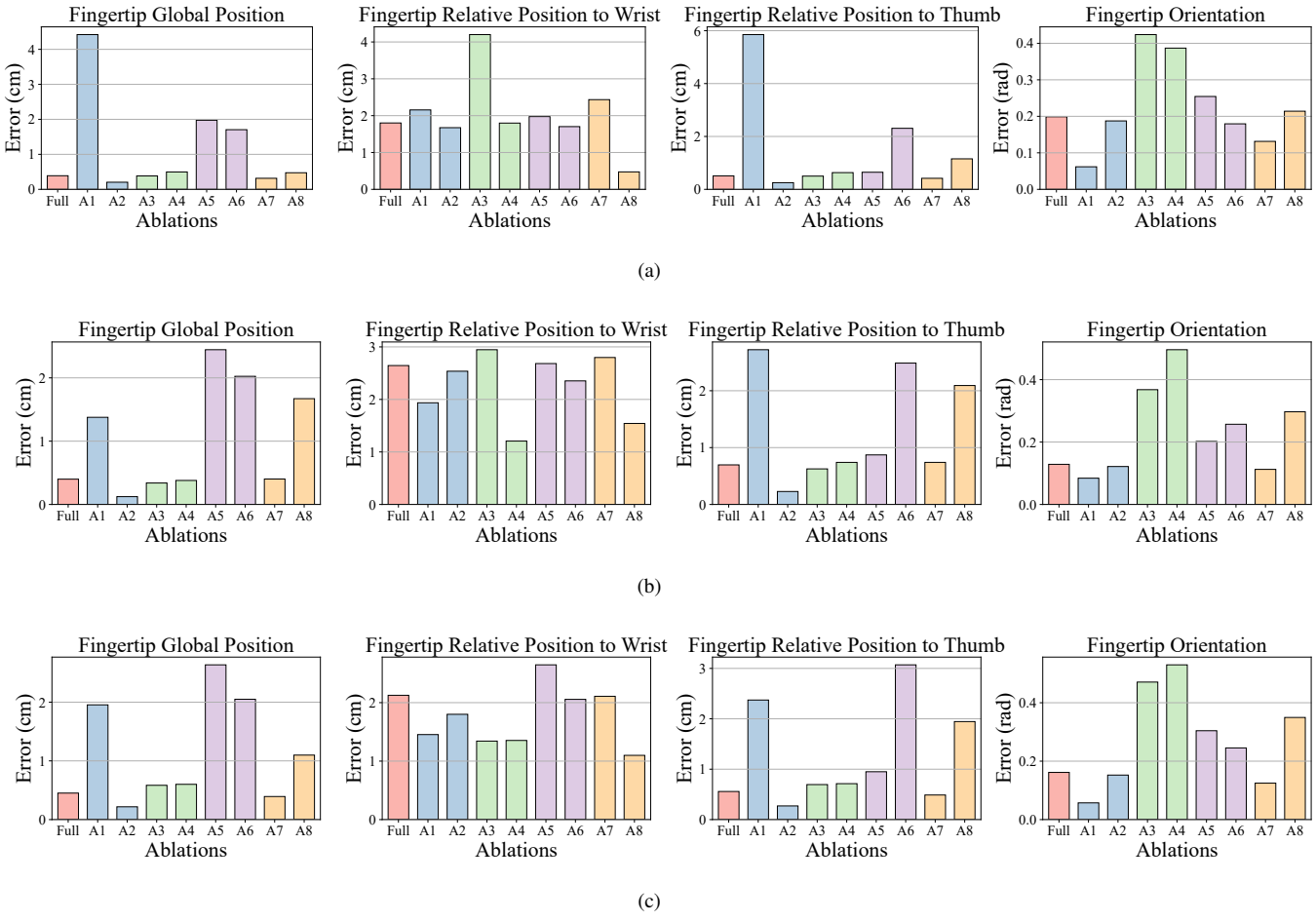


Fig. 5. Additional results on kinematic posture Retargeting. (a) Kinematic posture retargeting results using Leap hand on another trajectory involving finger crossing motion. (b) Kinematic posture retargeting results using Shadow hand on the same pinch motion trajectory as in the main text. (c) Kinematic posture retargeting results using Shadow hand on the trajectory involving finger crossing motion.

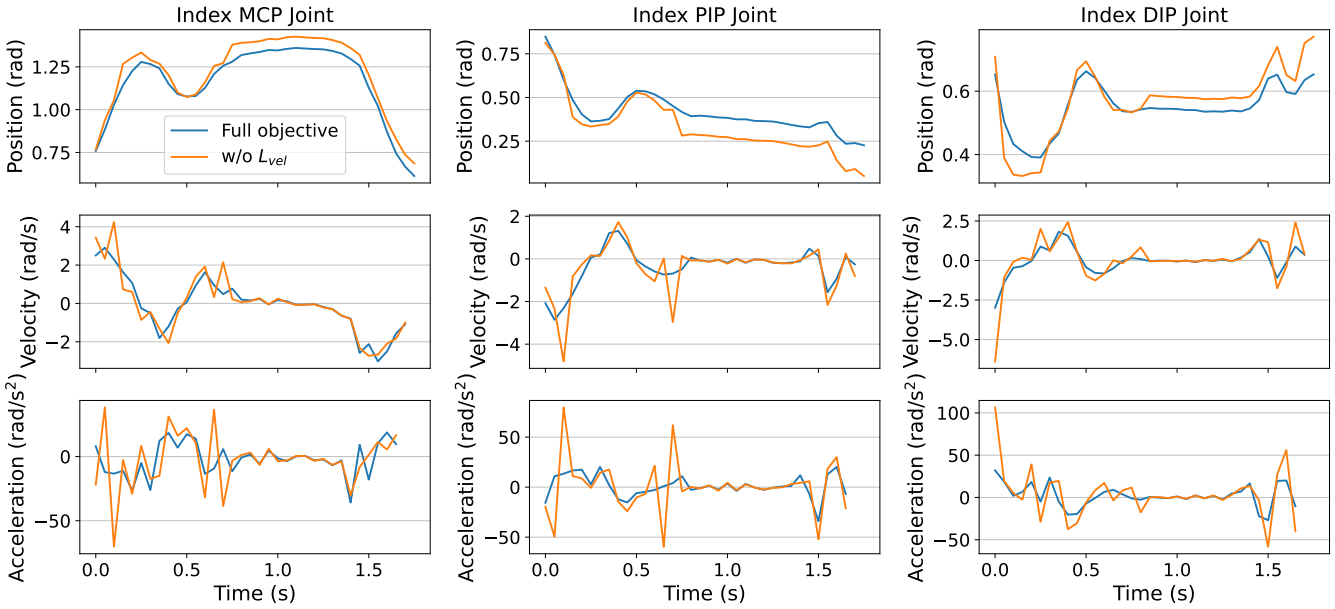


Fig. 6. Index MCP/PIP/DIP joint position/velocity/acceleration profiles with and without $\mathcal{L}_{\text{vel}}$. For all three joints, the position trajectories are smooth and free of oscillations in both cases. However, removing $\mathcal{L}_{\text{vel}}$ consistently leads to larger peaks in the velocity and especially in the acceleration profiles, indicating sharper changes in joint motion during the finger-crossing motion.

Explanation of Joint Velocity Regularization Term

As introduced in Sec. II, the joint-velocity regularization term

$$\mathcal{L}_{\text{vel}} = \sum_{j=1}^{m} w_j^{\text{vel}} \left\| q_j - q_j^{\text{prev}} \right\|_2^2$$

penalizes large changes in joint positions between consecutive timesteps and is intended to encourage smooth retargeted joint trajectories.

To illustrate the effect of this term, we compare the full objective in (6) with an ablation that removes $\mathcal{L}_{\text{vel}}$. We use a real-world teleoperation trajectory that involves finger-crossing motions with the index finger and generate the corresponding retargeted joint trajectories under both settings. The trajectories are evaluated on the three joints of the index finger (MCP, PIP, and DIP) on the LEAP Hand.

Figure 6 shows the joint position, velocity, and acceleration profiles for the index MCP, PIP, and DIP joints, respectively, under the full objective (blue line) and the variant without $\mathcal{L}_{\text{vel}}$ (orange line). These results empirically support the importance of $\mathcal{L}_{\text{vel}}$ in suppressing high-frequency variations and reducing acceleration spikes, thereby improving the overall trajectory smoothness.

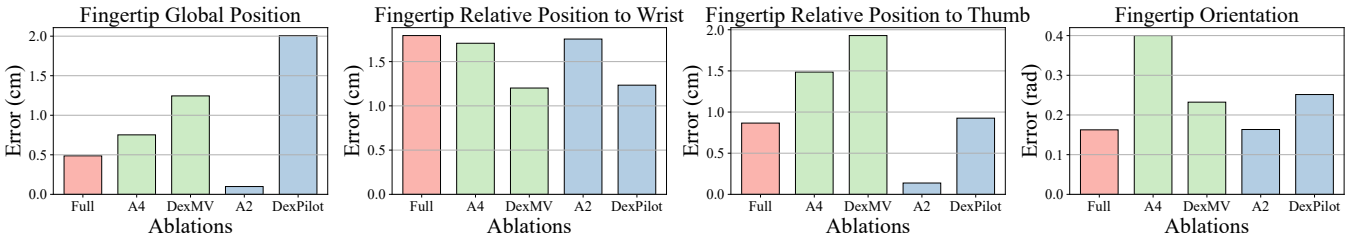
Comparison with Existing Retargeting Approaches

In addition to the ablation studies, we also compare our full proposed method with two representative retargeting approaches, DexMV and DexPilot [19], [28]. In our ablation setting, A2 and A4 are designed to closely mimic the key design choices of DexPilot and DexMV respectively. Specifically, A2 uses the raw fingertip pinch distance in the pinch term; unlike DexPilot, it does not include the fingertip distance rescaling function to ensure minimal distance between the thumb and a primary finger and force minimum separation distance between two primary fingers. A4 replaces the vectors from DIP joints to fingertips with vectors from the wrist to DIP joints, which is similar to the fingertip orientation formulation in DexMV (we use DIP joints instead of the PIP joints used in the original DexMV).

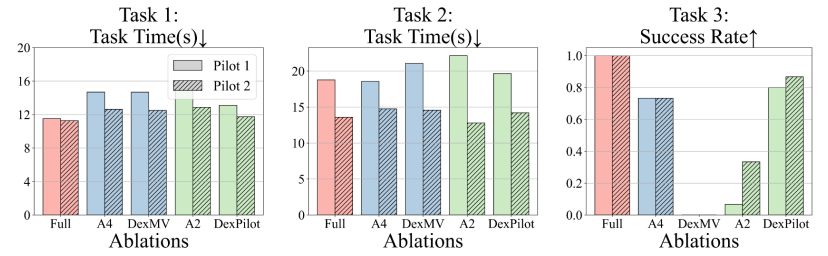
In contrast, the pinch term in our full objective differs from DexPilot in two aspects. First, we use a continuous switching weight $s(d_i)$ based on a sigmoid function, instead of the discrete weights in DexPilot. Second, our distance rescaling function $l(d_i)$ varies continuously on $[\epsilon_2, \epsilon_1]$, while DexPilot directly clamps fingertip distances below a fixed threshold. As a result, in our objective, both the rescaled distance and its weight change smoothly as the human pinch distance crosses threshold ϵ_2 and ϵ_1, which leads to smoother and more stable motions during precise pinching in real-world teleoperation.

To further ensure a fair comparison beyond these approximate variants, we additionally implement DexMV and DexPilot by directly following their objective functions and hyper-parameter settings. For DexPilot, we follow the code in the dex-retargeting repository [20]. Since their formulations are defined only for hand retargeting, we integrate them into our arm-hand retargeting pipeline by first applying their hand retargeting objective in the wrist frame and then separately solving the arm IK to track the resulting wrist pose.

We evaluate the full objective, the two ablations (A2 and A4), DexMV, and DexPilot on the LEAP-hand kinematic retargeting trajectories on finger pinch motions described in Sec. III-A, and compute the same four metrics as in Fig. 3: fingertip global position error, fingertip relative position to the wrist, fingertip relative position to the thumb, and fingertip



(a)



(b)

Fig. 7. (a) Results of kinematic posture retargeting on finger pinch trajectories. (b) Results of the real-world manipulations. Task 1 and 2 are assessed by task time, while Task 3 is evaluated by success rate. The bars represent the full objective, two ablations (A2, A4), DexMV, and DexPilot.

orientation error. We also evaluate them on the same three real-world manipulation teleoperation tasks as Sec. III-A.

Figure 7 summarizes the quantitative kinematic posture retargeting and real-world manipulation results. The results show that, on the kinematic retargeting trajectories (Fig. 7(a)), the full objective consistently achieves lower fingertip global position error, fingertip relative position to the thumb, and fingertip orientation error compared to DexMV and DexPilot.

The real-world teleoperation experiments in Fig. 7(b) further validate our design choice. Across all three manipulation tasks, the full objective achieves the shortest or comparable task times and the highest success rates for both human pilots. DexMV and DexPilot, by contrast, generally lead to slower executions and more failures, especially on Task 3 which requires precise fingertip pinch. In particular, DexMV fails in every trial of Task 3 because it does not include the fingertip pinch term and therefore cannot close the gap between thumb and index fingertips, which is similar to the results of ablation A1 where we remove the pinch term. These results demonstrate that the proposed objective not only improves kinematic retargeting quality but also leads to more efficient and reliable performance in real-world dexterous teleoperation compared to existing retargeting approaches.

IMPLEMENTATION

Hyper-parameter Setting

The hyper-parameters used in the retargeting objective are listed in Table III.

The weights λ_1–λ_5 reflect the relative importance of different retargeting objectives. We assign larger weights of 10 to the thumb-tip position, fingertip orientation, and pinch terms (λ_1, λ_4, λ_5), as these factors are crucial for dexterous manipulation and contact-rich pinching tasks. The weight of 0.1 for the wrist orientation term (λ_2) is relatively smaller as the objective aims to emphasize fingertip tracking accuracy rather than wrist pose accuracy. The weight for the fingertip position term (λ_3) is set to 1 so that its contribution is of the same order of magnitude as the other terms when evaluated in typical retargeting scenarios.

TABLE III
HYPER-PARAMETERS

Hyper Parameter	**Value**
λ_1	10
λ_2	0.1
λ_3	1
λ_4	10
λ_5	10
w_j^{pos} (Leap hand)	0.5, $j = 7, 11, 15, 18$ 0.1, $j = 20$ 0, else
w_j^{pos} (Shadow Hand)	0.5, $j = 9, 13, 17, 22$ 0.1, $j = 26$ 0, else
w_j^{vel} (Leap hand)	0.1, $j = 0 \sim 6$ 0.01, $j = 7 \sim 22$
w_j^{vel} (Shadow Hand)	0.1, $j = 0 \sim 6$ 0.01, $j = 7 \sim 30$

For the weights of joint position and velocity regularization terms w_j^{joint} and w_j^{vel}, index j from 0 to 6 corresponds to the joints of the Panda arm, while indices $j = 7$ to 22 and $j = 7$ to 30 correspond to the joints of the Leap hand and the Shadow Hand respectively. Note that here we assume all DoFs of the Shadow Hand are actuated.

The joint position regularization weights w_j^{pos} are determined according to the physical roles of different joints.

For Leap hand, $j = 7, 11, 15$ correspond to the abduction/adduction joints of index, middle and ring, $j = 18$ corresponds to the DIP joint of the ring, and $j = 20$ corresponds to the rotation of the thumb. For Shadow Hand, $j = 9, 13, 17, 22$ correspond to the finger movements of index, middle, ring and little finger in the palm plane, while $j = 26$ corresponds to the rotation of the thumb. Non-zero weights are assigned to these joints whose extreme values tend to produce unnatural or mechanically unfavorable configurations. For joints with non-zero position regularization, the pre-defined joint configurations are set to $\bar{q}_j = 0$.

The joint velocity regularization weights w_j^{vel} are chosen to penalize large changes in joint positions compared to previous timestep to encourage trajectory smoothness. In practice, we use larger w_j^{vel} for the arm joints and slightly smaller values for the finger joints, as arm motions are more prone to sudden changes. As shown in Fig. 5, this choice effectively reduces acceleration spikes and leads to smoother joint trajectories.

In our implementation, we rescale the size of the human hand by a factor of 1.5 for the Leap hand and 1.0 for the Shadow Hand to address the size difference between human and robot hand. In the real-world experiments, the retargeting control frequency is 20 Hz, and we use an exponential moving average with $\alpha_{\text{ema}} = 0.3$ to further smoothen the joint movements:

$$\boldsymbol{q}_t = \alpha_{\text{ema}} \cdot \boldsymbol{q}_t + (1 - \alpha_{\text{ema}}) \cdot \boldsymbol{q}_{t-1} \tag{10}$$

Control Approach

In the real-world experiments, we use joint-space position control for both the Panda arm and the robot hand. At each retargeting step (20 Hz), the retargeting optimizer outputs a desired joint configuration $\boldsymbol{q}_t$ for all actuated joints. This configuration is first upsampled to a 100 Hz command stream via interpolation and then sent to the hardware. The LEAP Hand executes the control command via a built-in current-based PD controller. The Franka arm runs an impedance controller at a higher 1000 Hz with strict acceleration and jerk limits, so the 100 Hz joint targets are further interpolated to match the 1000 Hz control frequency before being executed by the arm controller.

Discussion on Runtime, Latency, and Robustness

a) Runtime and latency: We profiled the computational cost of our retargeting pipeline on the LEAP-hand kinematic retargeting experiments. The full teleoperation loop runs at 20 Hz, corresponding to a frame time of 50 ms. Within each frame, the nonlinear optimizer for the retargeting objective takes on average 33 ms, which is lower than the frame time and is suitable for real-time control.

Overall, latency in the system mainly comes from three stages: (i) hand pose estimation and streaming from the Vision Pro headset; (ii) retargeting optimization; (iii) communication latency between our teleoperation node and the robot; and (iv) the low-level control latency on the robot side. The Vision Pro perception module runs in a separate thread and continuously receives hand pose estimations, while our 20 Hz teleoperation loop always uses the latest available hand pose at the beginning of each control cycle. After that, the optimizer runs within about 33 ms and the resulting joint commands are immediately sent to the low-level joint controllers. The latency introduced by our retargeting module is below 50 ms, and the Vision Pro streaming introduces an additional latency of about 50 ms [34]. Overall, the end-to-end latency of the system is approximately 0.15–0.25 s, which we empirically found sufficient for ensuring real-time teleoperation in all our real-world manipulation tasks.

b) Robustness to tracking noise and occlusions: Accurate detection and tracking of human hand pose is key to dexterous teleoperation. We experimented with both a single RGB camera and the Vision Pro headset for hand pose detection. Compared to the RGB camera, Vision Pro provides a more flexible field of view and more accurate hand pose tracking, but it still inevitably exhibits some tracking noise, especially in precise motions such as pinching. In practice, we observed that when the human thumb and index fingertip are already in contact, the estimated fingertip distance from Vision Pro can remain slightly positive and exhibit small jitter over time. We specially design the pinch term in our objective to be more robust to tracking error to enable stable pinching: we rescale the estimated fingertip distances in a small range $[\epsilon_2, \epsilon_1]$ to $[0, \epsilon_1]$ and clamp values below ϵ_2 to zero (Sec. III-A), so that sufficiently small estimated distances are treated as a closed pinch.

With respect to occlusions, Vision Pro can estimate a consistent hand pose under slight self-occlusion conditions, such as when a finger is partially occluded by the palm, as long as the operator keeps the hand within the headset's field of view. This was sufficient for all three teleoperation tasks in our experiments, but for scenarios that require robustness under severe occlusions, Vision Pro may produce inaccurate hand pose estimation. Glove-based hand tracking systems will offer better robustness and be more occlusion-resistant in such cases [13], [15].