TypeTele: Releasing Dexterity in Teleoperation by Dexterous Manipulation Types

Yuhao Lin*[1], Yi-Lin Wei*[1], Haoran Liao [1], Mu Lin [1], Chengyi Xing [2], Hao Li [2]
Dandan Zhang [3], Mark Cutkosky [2], Wei-Shi Zheng†[1,4]
[1] Sun Yat-sen University, China, [2] Stanford University, USA, [3] Imperial College London, UK
[4] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
{linyh96, weiylin5}@mail2.sysu.edu.cn wszheng@ieee.org

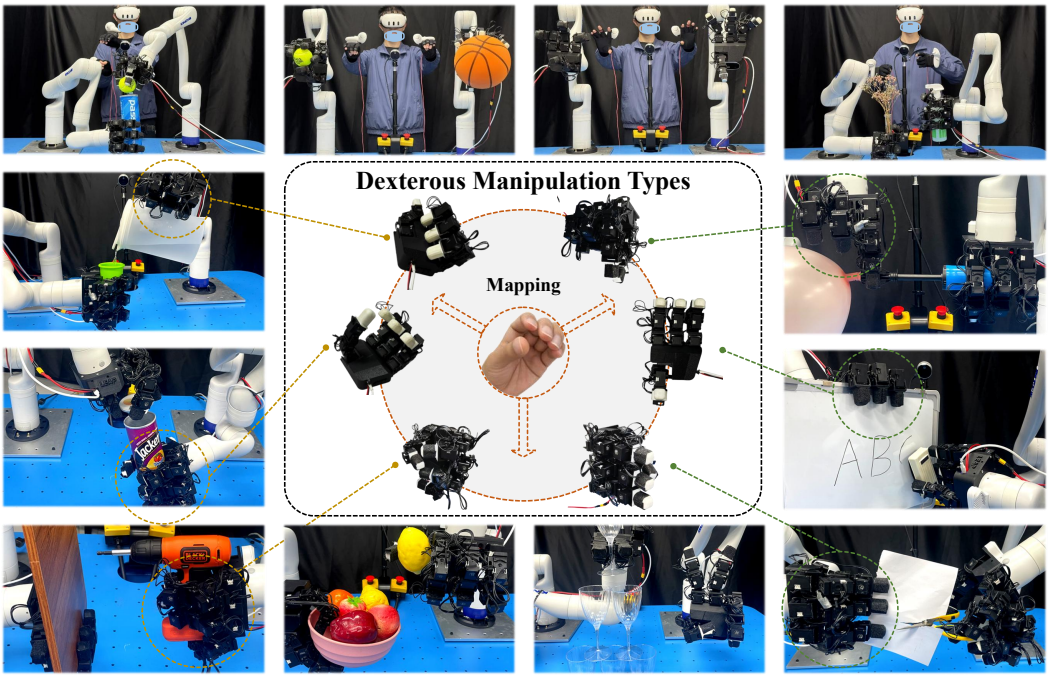

Figure 1: **TypeTele**, an effective dexterous teleoperation system, enables operators to complete various manipulation tasks by corresponding human hands with different types of robotic hands.

Abstract: Dexterous teleoperation plays a crucial role in robotic manipulation for real-world data collection and remote robot control. Previous dexterous teleoperation mostly relies on hand retargeting to closely mimic human hand postures. However, these approaches may fail to fully leverage the inherent dexterity of dexterous hands, which can execute unique actions through their structural advantages compared to human hands. To address this limitation, we propose **TypeTele**, a type-guided dexterous teleoperation system, which enables dexterous hands to perform actions that are not constrained by human motion patterns. To support this system, we build an extensible dexterous manipulation type library to cover comprehensive dexterous postures used in manipulation tasks. During teleoperation, we employ a MLLM-assisted type retrieval module to identify the most suitable manipulation type based on the specific task and operator commands. Extensive experiments of real-world teleoperation and imitation learning demonstrate that the incorporation of manipulation types significantly takes full advantage of the dexterous robot's ability to perform diverse and complex tasks with higher success rates. The project page is https://isee-laboratory.github.io/TypeTele.

Keywords: Dexterous, Teleoperation, Manipulation

*Equal contribution. †Corresponding author.

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

1 Introduction

With the development of learning-based methods and large-scale robotic datasets, dexterous robots have become increasingly capable of performing diverse and delicate tasks [1, 2]. Teleoperation plays a critical role in collecting real-world data, as it enables the acquisition of high-quality robotic demonstrations under realistic observations and physically executable actions [3, 4, 5].

Previous dexterous teleoperation methods aim to control the dexterous hand by imitating human hand postures, typically achieved by first capturing the human hand poses and retargeting them to the dexterous hand [6, 7, 8, 5]. Most hand retargeting approaches employ optimization or inverse dynamics techniques to preserve the spatial consistency of vectors between the wrist and predefined keypoints (such as fingertips) in both the human and robotic hands [5, 7, 9]. However, the retargeting paradigm is unable to fully utilize this dexterity of the dexterous hand, leading to difficulty in performing both basic and complex tasks.

However, two challenges hinder the effectiveness of teleoperation in existing methods. **First, the retargeting paradigm restricts the dexterous hand to motions feasible for human hands**, as it enforces consistency between human and robotic hand postures. The fully actuated dexterous hand can perform poses that humans cannot, but are more suitable to complete specific manipulation tasks, as shown in the left of Fig. 2. **Second, morphological differences between human and robotic hands may lead to the unreasonable retargeting poses.** Existing methods typically align corresponding vectors between the two hands and solve for a pose in the full joint space. However, differences in kinematics often lead to unstable postures, self-collisions, or undesirable contact directions in the robotic hand [10, 11, 12], as shown in Fig. 2.

To overcome these problems, we propose **TypeTele**, a type-guided dexterous teleoperation system, which allows operators to employ appropriate dexterous manipulation types to manipulate different objects and complete different tasks. The introduction of types offers two benefits, which address the two aforementioned bottlenecks: **1)** Introducing dexterous manipulation types enables robots to perform actions that the human hand cannot perform. **2)** Dividing dexterous actions into discrete types improves the effectiveness and rationality of the dexterous hand postures.

To support our system, we construct a dexterous manipulation type library organized using a hierarchical taxonomy that covers typical actions required in manipulation tasks. Each manipulation type is annotated with corresponding stretching and contracting postures of the robotic hand, which determines the range of executable actions within this type. Based on this library, our teleoperation framework operates in two stages: type retrieval and action execution. For type retrieval, we propose an MLLM (Multi-modality Large Language Model)-assisted type retrieval module that identifies the most appropriate manipulation type based on the current task. For action execution, we design an interpolation mapping strategy that maps the natural human hand action to the specific dexterous manipulation type, enabling intuitive control of the robotic hand through human motion.

The experimental results demonstrate the effectiveness of our teleoperation system: 1) Our system enables the successful execution of tasks that are unachievable using retargeting-based teleoperation system. 2) Our system significantly improves the data collection efficiency. 3) The data collected by our system shows higher quality, which benefits subsequent imitation learning and enhances the performance and robustness of autonomous policies. 4) The key insight of introducing type into teleoperation shows strong applicability to various tasks and can be applied to different systems.

2 Related Works

2.1 Dexterous Teleoperation

Teleoperation is a fundamental task for robotics[13, 14], as it not only enables remote operation of robots but also facilitates data collection for imitation learning [15, 6, 16]. Research on teleoperation for two-finger gripper robots primarily focuses on arm control, such as master-slave systems [17, 4] and VR devices [18], achieving impressive performance. Compared to grippers, dexterous hands

Figure 2: The challenges of previous retargeting-based dexterous teleoperation systems. Unachievable Grasping shows poses that are physically infeasible for human hands. Unstable Grasp leads to object dropping due to weak contact. Self-Collision indicates finger interference during motion. Undesired Contact refers to insufficient contact between the tactile sensor surfaces and the object.

offer greater dexterity for fine-grained manipulation tasks, but they also introduce challenges in hand pose mapping due to morphological differences between human and robotic hands [19, 20, 21, 12]. Previous methods focus on human hand pose capture and pose mapping to closely mimic human hand postures, typically involving different hardware setups for motion capture [7, 22, 3, 8, 5] while sharing fundamentally similar retargeting algorithms [20, 9]. These methods face two key limitations: (1) they are limited to actions feasible for human hands and (2) pose mapping remains challenging due to morphological differences, hindering fine manipulation tasks. In this paper, we propose a type-guided teleoperation system to address these limitations, enabling more complex manipulation tasks despite morphological mismatches.

2.2 Dexterous Manipulation

Achieving autonomous and generalizable dexterous manipulation is a long-term goal for robotics community[23, 1]. With the development of deep learning, imitation learning methods have shown great promise to achieve this goal [24, 2, 11], with transformer-based [15, 25], diffusion-based[26, 27, 28] or Vision-Language-Action based architecture [29, 1]. However, the effectiveness of these methods largely depends on the quality and scale of expert demonstration data [30, 31, 32]. To address this, our system achieves higher data quality and collection efficiency, which facilitate more effective imitation learning and improve the performance of autonomous policies.

3 Type-guided Teleoperation System

3.1 System Overview

Our system's overview is shown in Figure 3. First, we construct a dexterous manipulation type library, covering types required for various manipulation tasks. Then, we propose a MLLM-assisted type retrieval module to select the most appropriate type based on the current task. We also design a type adjustment strategy to improve versatility. Finally, during teleoperation, we use an interpolation mapping strategy to control the dexterous action of specific type by human hand motion.

3.2 Dexterous Manipulation Type Library

Inspired by existing human grasp taxonomies that classify hand postures into distinct types to encompass most human manipulations [33], we design a **Dexterous Manipulation Type Library**, comprising diverse dexterous types to guide dexterous postures across a wide range of teleoperation tasks, as shown in Figure 4. The library is built upon recent taxonomies [34, 33, 35, 2, 36] and augmented with postures specially designed for dexterous hands, which are extracted from a variety of dexterous manipulation tasks.

To effectively cover the dexterous manipulation action space and better organize the library, we classify the dexterous manipulation types into two primary categories: single-hand types and bimanual collaborative types, as shown in Figure 4. Single-hand types are further subdivided into grasp types and non-grasp types. Grasp types further include two subcategories: robot-exclusive grasp types, which support manipulation tasks that exceed human hand capabilities, and general grasp types,

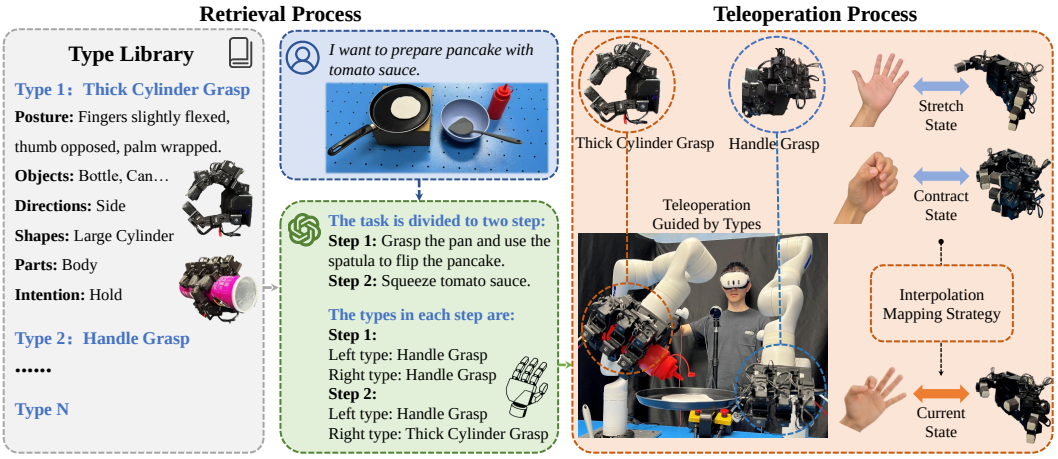

Figure 3: TypeTele includes a retrieval process using a MLLM to select manipulation types from the library, and a teleoperation process that applies them with an interpolation-based mapping strategy.

which are derived from established human grasp taxonomies. Bimanual collaborative types are subdivided into symmetric and asymmetric types based on the relative positions and functional roles of the two hands during manipulation. Our library is composed of 4 sub-categories and 30 types.

Specifically, each dexterous type is annotated with stretching and contracting postures, which correspond to the natural stretching and contracting postures of the human hand. [37] Additionally, to facilitate the autonomous retrieval of type, each dexterous type is annotated with object-centric and posture-centric information to describe: (1) what kinds of objects and task this posture is suitable for manipulating; (2) what the posture specifically looks like. These details are organized into manipulation attributes belonging to each dexterous type, as shown in the left of Figure 3, facilitating type retrieval in teleoperation. More details can be found in supplementary materials.

3.3 MLLM-assisted Type Retrieval

To facilitate teleoperation, we propose an MLLM-assisted type retrieval framework that autonomously selects the most appropriate manipulation type for a given task. In this framework, all types from the library, annotated with attribute descriptions, are converted into language prompts for an MLLM like GPT-4o [38]. We then prompt the MLLM to sequentially reason through two sub-questions: (1) *How many steps are required to complete the task?* and (2) *Which type of manipulation should be assigned to each hand per step?* The MLLM is first guided to decompose the task into a series of steps, infer which objects are involved in each step, and determine the interaction method. Based on this reasoning, the MLLM infers the desired attributes of the manipulation type for each hand, which are then used to retrieve the most suitable type from the library.

We develop a voice control program to enable hands-free interaction with the retrieval module, freeing operators' hands for robotic control tasks. This system utilizes Whisper [39] to transcribe operator speech into text. The real-time captured image from the camera and the text prompt are subsequently passed to the GPT-4o API to generate type recommendations, which are then processed to automatically switch or adjust the corresponding control mode or joint configuration.

3.4 Type Adjustment Strategy

Our system supports type adjustment to further enhance its versatility, while our type library can already cover most common tasks, and each type generalizes well across objects with similar geometric characteristics. To enable such adjustment, the system allows users to explicitly apply offsets to the position or orientation of specific fingertips.

Specifically, the system first obtains the initial fingertip position and orientation of the origin type through forward kinematics. And the desired adjustment can be specified either by capturing the 6-

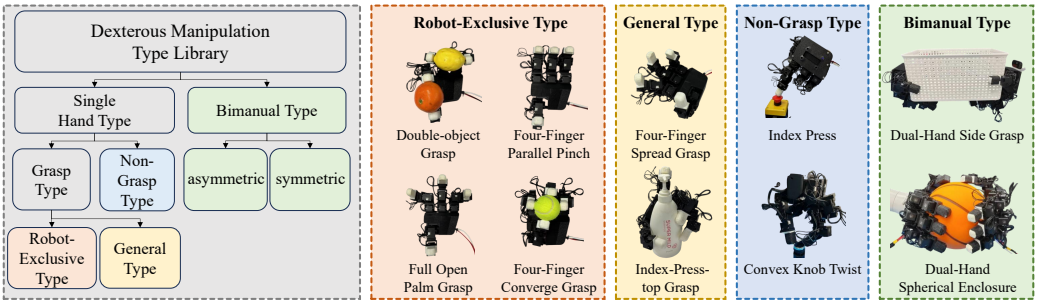

Figure 4: The illustration of the dexterous manipulation library. The left side presents the hierarchical taxonomy of the library, while the right side displays examples from each category.

DOF motion of the user's fingertip or by providing transformation values obtained through manual input. The system then applies the offset to the fingertip pose and uses the resulting position and orientation to compute the adjusted joint angles via inverse kinematics, formulated as:

$$q' = IK\left(FK(q) \cdot T_{\triangle}\right) \tag{1}$$

where q denotes the initial joint angles, $FK(\cdot)$ represents the forward kinematics function, $T_{\triangle}$ is the desired transformation applied to the end-effector pose, $IK(\cdot)$ denotes the inverse kinematics function; and q' is the resulting new joint configuration. To ensure that the adjusted type closely matches the original pose, the system initializes the inverse kinematics solver with the joint angles of the origin type, thereby avoiding unintended deviations or discontinuities in the resulting posture.

3.5 Interpolation Mapping Strategy

We design an interpolation mapping strategy to intuitively control robotic dexterous hands using human hand motions. Specifically, we first associate the stretched and contracted postures of the human hand with the corresponding postures of the robotic hand. Given the current human hand posture, we compute a normalized projection ratio for each fingertip position along the 3D vector defined by the stretched and contracted positions:

$$p_{\text{ratio}} = \text{clip}\left(\frac{(\mathbf{p}_{\text{current}} - \mathbf{p}_{\text{stretch}}) \cdot (\mathbf{p}_{\text{contract}} - \mathbf{p}_{\text{stretch}})}{\|\mathbf{p}_{\text{contract}} - \mathbf{p}_{\text{stretch}}\|^2},\ 0,\ 1\right), \tag{2}$$

where $\mathbf{p} \in \mathbb{R}^3$ denotes the 3D fingertip position, $\cdot$ denotes the dot product, and clip constrains the output within the range $[0, 1]$. The resulting scalar p_{ratio} is then used to linearly interpolate between the stretched and contracted joint angles of the robotic hand:

$$\theta_{current} = p_{ratio} \cdot (\theta_{contract} - \theta_{stretch}) + \theta_{stretch}, \tag{3}$$

where $\theta_{contract}$ and $\theta_{stretch}$ represent the joint angles corresponding to the fully contracted and stretched states, respectively.

3.6 Hardware and Robot Control

The hardware of our system involves the hand motion capture device, robot arm, dexterous hand and camera. For motion capture, we use Rokoko Gloves to capture each finger's 3 DOF position and the controller of Meta Quest 3 for the wrist's 6 DOF pose, following [8]. The robotic system includes two Kinova arms (6 DOF and 7 DOF, respectively) and two LEAP Hand (16 DOF each) [40]. For vision, a Realsense L515 LiDAR Camera captures single-view RGB-D observations of the scene.

For dexterous hand control, we use joint position PD control, where the target position is obtained through interpolation in the mapping. For arm control, we utilize the high-frequency Cartesian velocity control [41] interface provided by Kinova. The arm's motion is predefined as a uniform acceleration and deceleration motion for smoothness. The maximum translational velocity is fixed at 20 cm/s. The rotational velocity is dynamically adjusted: it increases when the orientation error is large, with an upper bound enforced to ensure safety. Additionally, Kalman filtering [42] is applied to smooth the estimated velocity signals and further enhance the continuity of motion. More details of robot control can be found in supplementary materials.

(a) Hardware System Setup (b) Objects

Figure 5: The illustration of hardware system setup and the objects used in experiments.

4 Experiment

4.1 Experiment Setting and Evaluation Metrics

Tasks. We design a diverse set of tasks to evaluate the effectiveness of both our teleoperation system and the imitation learning policy. For better comparison, we referenced prior work in task design and adopted several existing tasks [6, 3, 8, 22]. Additionally, we introduce more challenging tasks that are difficult to complete using previous systems, in order to demonstrate the superior performance of our framework. Details of the tasks are provided in the supplementary materials.

Teleoperation Setting and Metrics. We compare our type-guided teleoperation system with the retargeting-based baseline, where human hand postures are directly mapped to the robot [7, 8, 3]. Both systems share identical hardware and robot control algorithms to ensure a fair comparison. We collected demonstration data from 10 participants with a range of experience in robotics and teleoperation, gathering 20 successful demonstrations for each task. We record the success rate Suc, the total time spent of completing the data collection of one task T_{all} including the time for failure cases, and the average demonstration duration T_{single} for each successful demonstration. Higher success rates and shorter durations indicate better performance [3].

Imitation Setting and Metrics. We adopt the state-of-the-art diffusion-based policy, iDP3 [27], as our imitation learning algorithm. The policy takes a single-view 3D observation and current robot proprioception as conditional inputs, and outputs the desired Cartesian position of the robot arm's end-effector and the joint angles of the dexterous hand. Observation and action horizon vary from 3–8 and 13-8 respectively, depending on task length. To evaluate the impact of teleoperation quality, we train separate policies on datasets collected from the retargeting-based and our systems, using the same number of demonstrations and policy hyper-parameters. Higher task success rates achieved by a policy indicate higher-quality demonstrations and thus more effective data collection.

4.2 Comparison Results

TypeTele significantly improves the efficiency of data collection. The teleoperation results in Table 1 show that our system achieves a shorter overall collection time, a higher task success rate, and reduced lengths of averaged demonstration trajectory lengths. This indicates that leveraging dexterous manipulation types enables more convenient object manipulation and more stable grasping, which is highly beneficial for improving teleoperation performance.

TypeTele enables the successful execution of tasks that are unachievable using retargeting-based teleoperation system, as shown in the results for challenging tasks (the final 5 tasks) presented in Table 1. Specifically, when using scissors and spray bottles, the dexterous hand requires accurate manipulation while maintaining a stable grasp. For using a heavy kettle, a firm grip is necessary to counteract the object's weight. When opening a large box or grasping two objects, the hand must open widely to securely grasp and lift the lid or open to specific postures. These tasks are particularly challenging for retargeting-based teleoperation systems, due to issues of unstable

grasping and undesired contacting, as mentioned in Figure 2. These results highlight the enhanced capability of our system in handling complex and dexterous manipulation tasks.

Task	Description	System	Suc	T_{all}	T_{single}
Pick and Place	Pick up the tennis ball and place it into the basket.	Baseline	95.2%	579.6	8.28
		Ours	100%	536.9	7.67
Collect and Store	Collect objects on the table and store them into a basket.	Baseline	60.6%	1231.6	37.32
		Ours	95.2%	616.8	29.37
Handover	Transfer the object from the left hand to the right hand.	Baseline	80.0%	459.5	18.38
		Ours	95.2%	244.4	11.64
Pouring from Pan	Grasp the handle of pan and pour its contents into the basket.	Baseline	14.2%	1149.4	16.42
		Ours	83.0%	174.9	14.57
Use Scissors	Use scissors to cut the paper strip into two pieces.	Baseline	0	-	-
		Ours	91.1%	161.1	5.37
Spray Water	Use the spray bottle to spray water toward the target direction.	Baseline	0	-	-
		Ours	86.9%	167.4	7.28
Use a **Heavy** Kettle	Grasp the kettle, lift it and pour water into the container.	Baseline	0	-	-
		Ours	85.0%	369.4	18.47
Open a **Large** Box	Open the lid of the large box, then pick up the items inside.	Baseline	0	-	-
		Ours	95.2%	398.6	18.98
Grasp Two Objects	Grasp two medium-sized objects by one hand simultaneously.	Baseline	0	-	-
		Ours	69.6%	488.3	21.23

Table 1: The teleoperation results compared with retargeting-based teleoperation system (baseline).

	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9
Baseline	10/10	3/10	1/10	1/10	-	-	-	-	-
Ours	10/10	10/10	6/10	9/10	9/10	9/10	9/10	9/10	8/10

Table 2: Comparison with the imitation policy trained using data collected by different teleoperation.

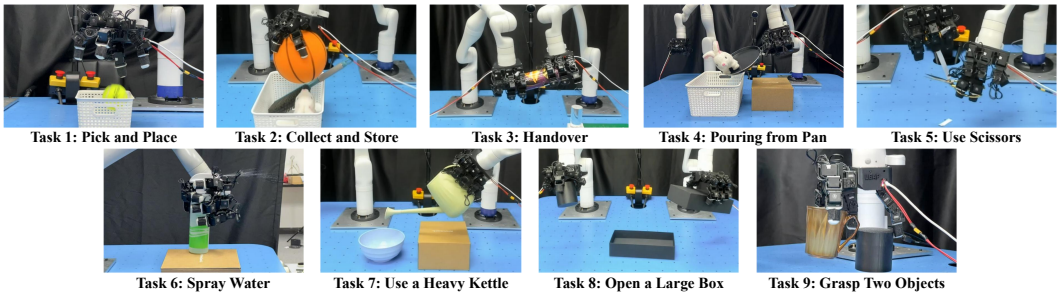

Figure 6: The visualization of autonomous policy execution process.

Imitation learning results demonstrate the higher quality of data collected by TypeTele. Since demonstration quality impacts imitation learning, we compare policies trained on an equal number of successful demonstrations from different systems. As shown in Table 2, the task order is consistent with Table 1, and entries marked "-" denote tasks the baseline failed to complete. Evaluated 10 attempts, the policy trained on our system's data achieves a higher success rate. This confirms the importance of high-quality demonstrations and our system's superior data collection capability.

4.3 Applicability of the TypeTele System

One dexterous manipulation type can be applied to various objects with similar geometric structures or functional properties. As shown in Figure 7, the type designed for objects like trigger sprayers and lotion pumps can generalize across different instances. And the type designed for square-shaped and objects with thin handle can adapt to objects of varying sizes. These results demonstrate the broad applicability of our dexterous manipulation type design.

TypeTele handles complex, long-horizon manipulation tasks involving multiple objects and stages, as shown in the bottom of Figure 7. Its ability to select and execute the appropriate type at each stage demonstrates strong adaptability and generalization for diverse interactions.

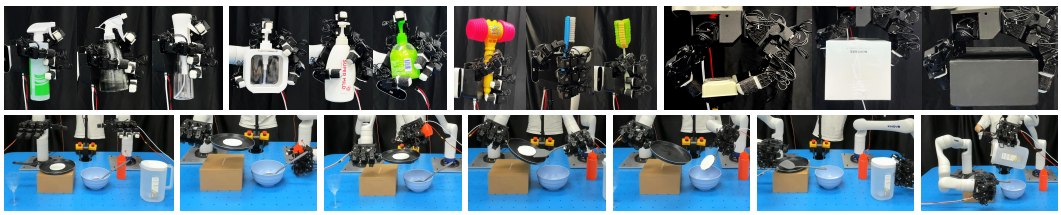

Figure 7: Top: Visualization that one type can apply to various objects with similar structures or functions. Bottom: Visualization of long horizon task involves mutlpile steps and objects.

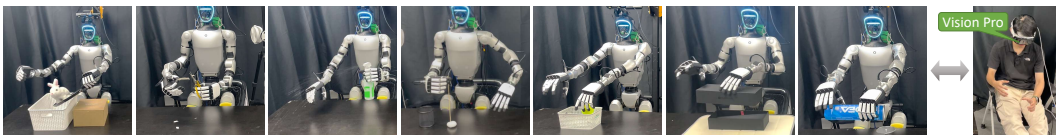

Figure 8: Left: Visualization of the experiments for Inspire Hand.

TypeTele is applicable to various dexterous robotic hands, as illustrated in Figure 8. We conduct real-world manipulation experiments using the Inspire Hand, demonstrating the adaptability of our method to diverse hardware. the detailed results are presented in Table 3 in the Appendix.

4.4 Efficiency of TypeTele System

During teleoperation, the system records data at 15 FPS using a Windows 10 PC with an Intel Core i7-14700 CPU. Inference with the imitation policy runs at 11 FPS on an NVIDIA GeForce RTX 3090 GPU. An independent control thread for the robotic arm consistently maintains a frame rate of 25 FPS during both teleoperation and inference. For the MLLM-assisted retrieval module, the average query time is 4.8 seconds. While the retrieval step is relatively time-consuming, it occurs only once per task, after which the teleoperator can switch among the retrieved manipulation types with minimal latency using voice commands, keyboard input, or a foot pedal. This approach minimizing the impact on the system's real-time performance. Furthermore, with contextual image caching, query latency can be reduced to under 2.4 seconds, mitigating potential bottlenecks for long-horizon or sudden incidents. These results demonstrate TypeTele's practical, system-wide efficiency.

4.5 Effectiveness of MLLM-assisted Retrieval Module

Experiments are conducted to evaluate the effectiveness of MLLM-assisted retrieval module. We construct 50 test environments, including 40 single-object tasks and 10 multi-object long-horizon tasks. A retrieval is considered successful if the retrieved manipulation type is suitable for the current task. The success rates are 91.89% for single-object tasks and 92.00% for multi-object tasks. This high accuracy confirms that our retrieval module can reliably identify appropriate manipulation types for diverse tasks. The operator can also interact with the MLLM to improve performance or fix errors. In 30 retrieval attempts conducted within severely cluttered multi-object scenes, 27 retrievals returned the optimal type. For other 3 sub-optimal but feasible types, a single corrective prompt led to the successful identification of the optimal grasp, confirming module's high efficacy.

5 Conclusion

We believe that achieving effective teleoperation for the data collection of delicate dexterous manipulation task is important in the robotic learning communities. In this paper, we propose Typetele, a novel dexterous teleoperation system with the insight that introducing types into teleoperation. To support this system, we build a dexterous manipulation library, comprising various types required for common dexterous tasks. During the teleoperation, a MLLM-assisted type retrieval module is proposed to select the suitable type for current task. And a interpolation mapping is used to control the dexterous hand by human hand motion. The extensive experiments show that our system not only enables tasks previously unachievable by teleoperation, but also greatly improves data collection efficiency and quality, thereby enhancing imitation learning and autonomous policy performance.

6 Limitations

The primary limitation of our system is its difficulty with highly dexterous in-hand manipulations, such as pen spinning, which require a high level of finger-level dexterity. This could potentially be addressed by incorporating learning-based methods. Integrating learning-based approaches with our dexterous manipulation types presents a promising avenue for future research. Furthermore, in specific situations requiring the adjustment module, its use introduces a temporal overhead to the task execution. This could be mitigated by simplifying the adjustment interaction mechanism or by developing an automated fine-tuning process based on visual feedback. Lastly, the Dexterous Manipulation Type Library will need to be adapted for different dexterous hands and expanded to include more types.

7 Acknowledgment

This work was supported partially by NSFC(92470202, U21A20471), Guangdong NSF Project (No. 2023B1515040025), Guangdong Key Research and Development Program(No.2024B0101040004). The authors would like to thank Zhizhao Liang for his generous contributions to this work, and Zesen Lin for his assistance with the experiments.

References

[1] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[2] H.-S. Fang, H. Yan, Z. Tang, H. Fang, C. Wang, and C. Lu. Anydexgrasp: General dexterous grasping for different hands with human-level learning efficiency. *arXiv preprint arXiv:2502.16420*, 2025.

[3] 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.

[4] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Proceedings of The 8th Conference on Robot Learning*, pages 4066–4083. PMLR, 2025.

[5] K. Shaw, Y. Li, J. Yang, M. K. Srirama, R. Liu, H. Xiong, R. Mendonca, and D. Pathak. In *8th Annual Conference on Robot Learning*.

[6] 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.

[7] 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. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[8] 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.

[9] 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*, pages 570–587. Springer, 2022.

[10] Y.-L. Wei, J.-J. Jiang, C. Xing, X. Tan, X.-M. Wu, H. Li, M. Cutkosky, and W.-S. Zheng. Grasp as you say: Language-guided dexterous grasp generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[11] K. Li, P. Li, T. Liu, Y. Li, and S. Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. *arXiv preprint arXiv:2503.21860*, 2025.

[12] 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.

[13] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci. Teleoperation of humanoid robots: A survey. *IEEE Transactions on Robotics*, 39(3):1706–1727, 2023.

[14] P. F. Hokayem and M. W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42 (12):2035–2057, 2006.

[15] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha: A low-cost open-source hardware system for bimanual teleoperation. *arXiv preprint arXiv:2309.13126*, 2023.

[16] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[17] J. Aldaco, T. Armstrong, R. Baruch, J. Bingham, S. Chan, K. Draper, D. Dwibedi, C. Finn, P. Florence, S. Goodrich, et al. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024.

[18] A. Erkhov, A. Bazhenov, S. Satsevich, D. Belov, F. Khabibullin, S. Egorov, M. Gromakov, M. A. Cabrera, and D. Tsetserukou. Viewvr: Visual feedback modes to achieve quality of vr-based telemanipulation. *arXiv preprint arXiv:2501.07299*, 2025.

[19] Z.-H. Yin, C. Wang, L. Pineda, F. Hogan, K. Bodduluri, A. Sharma, P. Lancaster, I. Prasad, M. Kalakrishnan, J. Malik, et al. Dexteritygen: Foundation controller for unprecedented dexterity. *arXiv preprint arXiv:2502.04307*, 2025.

[20] Y. Qin, H. Su, and X. Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[21] 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.

[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. In *Conference on Robot Learning (CoRL)*, 2024.

[23] A. M. Turing. *Computing machinery and intelligence*. Springer, 2009.

[24] 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. In *Robotics: Science and Systems (RSS)*, 2023.

[25] W. Wang, F. Wei, L. Zhou, X. Chen, L. Luo, X. Yi, Y. Zhang, Y. Liang, C. Xu, Y. Lu, et al. Unigrasptransformer: Simplified policy distillation for scalable dexterous robotic grasping. *arXiv preprint arXiv:2412.02699*, 2024.

[26] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[27] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024.

[28] Z. Xue, S. Deng, Z. Chen, Y. Wang, Z. Yuan, and H. Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

[29] Y. Zhong, X. Huang, R. Li, C. Zhang, Y. Liang, Y. Yang, and Y. Chen. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. *arXiv preprint arXiv:2502.20900*, 2025.

[30] H. Li, Y. Cui, and D. Sadigh. How to train your robots? the impact of demonstration modality on imitation learning. *arXiv preprint arXiv:2503.07017*, 2025.

[31] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[32] Y. Mu, T. Chen, S. Peng, Z. Chen, Z. Gao, Y. Zou, L. Lin, Z. Xie, and P. Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). *arXiv preprint arXiv:2409.02920*, 2024.

[33] M. R. Cutkosky et al. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on robotics and automation*, 5(3):269–279, 1989.

[34] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on human-machine systems*, 46(1):66–77, 2015.

[35] F. Krebs and T. Asfour. A bimanual manipulation taxonomy. *IEEE Robotics and Automation Letters*, 7(4):11031–11038, 2022.

[36] J. Chen, Y. Chen, J. Zhang, and H. Wang. Task-oriented dexterous hand pose synthesis using differentiable grasp wrench boundary estimator. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5281–5288. IEEE, 2024.

[37] A. Bicchi, M. Gabiccini, and M. Santello. Modelling natural and artificial hands with synergies. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1581): 3153–3161, 2011.

[38] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[39] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[40] K. Shaw, A. Agarwal, and D. Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

[41] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Transactions of the American society of mechanical engineers*, 64(8):759–765, 1942.

[42] R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.

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

Supplementary Materials

8 TypeTele System Details

8.1 Dexterous Manipulation Type Library

Visualization of Type Library

We construct a dexterous manipulation type library using the taxonomy, based on prior grasp type work [34, 33, 35, 2, 36], and extending it based on the structure of dexterous hands [40] and manipulation tasks. The visualization of the library is illustrated in Figure 9.

Figure 9: Visualization of Dexterous Manipulation Type Library.

Annotation Information of Dexterous Manipulation Types

Each manipulation type is annotated with descriptive information to characterize its posture and functionality, which facilitates retrieval. The annotated attributes include: *hand posture*, *manipulable object categories*, *contact parts on the object*, *the geometry of these parts*, *grasp direction*, and *intended manipulation purpose*. Examples of the annotated information are shown in Figure 10.

8.2 MLLM-assisted Type Retrieval Module

We employ GPT-4o [38] to retrieve the most suitable dexterous type for current task. The details of prompts are as following:

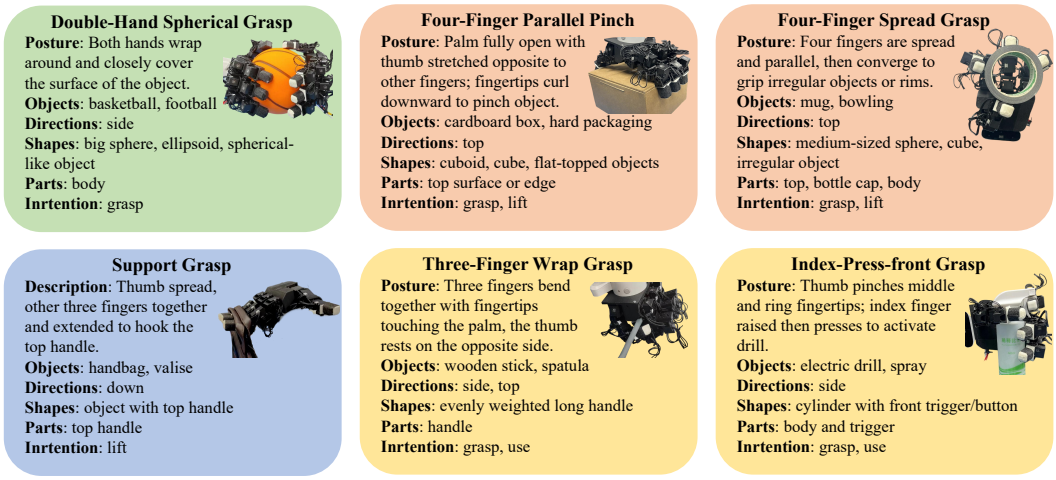

Figure 10: Visualization of the annotations examples of types in the library.

```
System Prompt:
You are a robotic manipulation expert. When given a user goal and an image
    of tools or ingredients, your job is to:
(1) Decompose the task into clear manipulation steps.
(2) Assign a suitable grasping type for each hand (left/right) in every
    step based on the provided grasp type library.
(3) Format your response in this structured way:
The task is divided into N steps:
Step 1: [describe the subtask]
Step 2: [describe the subtask]
...
The types in each step are:
Step 1: Left type: [grasp type name] Right type: [grasp type name]
Step 2: Left type: [grasp type name] Right type: [grasp type name]
...
User Command: I want to [describe the whole task].
```

Example of Output

User Command

```
I want to prepare a pancake with tomato sauce and a glass of water for
    breakfast.
```

GPT-4o

```
The task is divided into 3 steps:
Step 1: Pick up the pan with the pancake using the right hand, and hold and
     squeeze the tomato sauce bottle using the left hand to apply sauce.
Step 2: Keep holding the pan with the right hand, and use a spatula with
    the left hand to transfer the pancake into the bowl.
Step 3: Hold the water pitcher with the left hand and the cup with the
    right hand, then pour water into the cup.
The types in each step are:
Step 1:
Left type: Thick Cylinder Grasp (for squeezing the sauce bottle)
Right type: Three-Finger Load-Bearing Wrap Grasp (for lifting the pan by
    the handle)
Step 2:
```

```
Left type: Three-Finger Wrap Grasp (for using the spatula)
Right type: Three-Finger Load-Bearing Wrap Grasp (continue holding the pan)
Step 3:
Left type: Curved Handle Grasp (for holding the pitcher)
Right type: Thick Cylinder Grasp (for holding the glass)
```

8.3 Kinesthetic Teach Module

Although the types provided in our type library can handle the most of everyday applications, we offer a teaching mode for the creation of new type for special cases and unique user needs. This allows users without robotics expertise to intuitively and conveniently create dexterous types.

We implemented the teaching mode using admittance control and motor backdrivability. The formula for admittance control is as follows:

$$M\ddot{x}(t) + B\dot{x}(t) + Kx(t) = F_{\text{ext}}(t) \tag{4}$$

In this equation, $x(t)$ denotes the position of the control output, $\dot{x}(t)$ and $\ddot{x}(t)$ represent the velocity and acceleration respectively, and $F_{\text{ext}}(t)$ is the external force applied to the robot's end-effector. The parameters M, B, and K correspond to the virtual mass, damping, and stiffness, respectively.

Here, we estimate the external force using the current magnitude and positional deviation of the dexterous hand's motors. We also incorporate the motor's velocity information to give the motion a certain degree of inertia, making the teaching process smoother.

8.4 Robot Control

To enhance the control fidelity and operational fluidity of the robotic arms during both teleoperation and imitation learning inference, we implemented several key methodological improvements:

(1) Multi-threaded Device Communication: We have adopted a multi-threaded approach for device communication. Each distinct data stream – including RGB-D imagery, cartesian poses of the two end-effectors, and joint angles of the two robotic hands – is managed by an independent thread. This architecture ensures that when the main thread requires specific information, it can be provided instantaneously, thereby circumventing delays typically associated with data acquisition operations.

(2) Uniformly Accelerated Motion for Velocity Control: For both translational and rotational velocity control, we have applied uniformly accelerated motion profiles. This strategy guarantees that velocity changes are smooth and devoid of abrupt transitions. Consequently, the robotic arm's movements are exceptionally fluid, and this approach also mitigates jitter stemming from natural human hand tremors or sensor inaccuracies.

(3) Dynamic Speed Control for Rotation: A dynamic speed control scheme has been implemented for rotational movements. When the current orientation is significantly distant from the target orientation, the rotational speed will be increased, enabling the robotic arm to rapidly converge towards the desired direction. Conversely, as the current orientation approaches the target, the rotational speed will be reduced. This allows the operator to perform precise, fine-grained rotational adjustments.

(4) Dedicated Asynchronous Robot Control Thread: The robotic arms are controlled by a dedicated, separate thread, ensuring asynchronous operation. The main thread focuses solely on transmitting the target pose to this robot control thread. Subsequently, the robot control thread governs the robotic arms at a consistent control frequency. This approach guarantees stable and smooth robotic arm control, even when the main thread's frame rate fluctuates or varies, such as during transitions between teleoperation and imitation learning inference with their differing frame rates.

8.5 Details of Imitation Learning

We adopt a diffusion-based imitation policy to learn from expert demonstration data, following [27]. The observation input consists of single-view point clouds $x_i \in \mathbb{R}^{N\times 3}$ and robot proprioceptive inputs $x_p \in \mathbb{R}^p$. Specifically, we randomly downsample $N = 4096$ points from the raw depth maps. The proprioceptive input ($p = 44$) includes the Cartesian poses of both robot arms and the joint angles of the two dexterous hands.

The point clouds are encoded using a pyramid convolutional encoder [27], while the proprioceptive inputs are processed via a multilayer perceptron (MLP). We define the observation horizon as t_o and the action horizon as t_a. In our setup, we adopt a fixed total horizon length of $t_o + t_a - 1 = 15$. For tasks requiring longer-term reasoning or delayed consequences, we use longer observation horizons (e.g., $t_o = 6$ or 8) and shorter action horizons (e.g., $t_a = 10$ or 8), while for reactive or short-horizon tasks, we opt for shorter observations (e.g., $t_o = 3$ or 4) and correspondingly longer action horizons. This enables a flexible temporal encoding of task-relevant information, tailored to the nature of each behavior. All features are used as conditional inputs to predict the noise associated with the robot action $a \in \mathbb{R}^{k_a}$, where k_a denotes the action dimension specific to the task. The training objective minimizes the denoising score matching loss, formulated as:

$$\mathcal{L} = \mathbb{E}_{\mathbf{a}_0, \boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\left\|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{a}_t, t)\right\|^2\right], \tag{5}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ is the Gaussian noise. The network ϵ_θ is trained to predict the added noise given the noisy action $\mathbf{a}_t$ and the timestep t. We employ DDIM [43] for inference sampling.

$$\mathbf{a}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{a}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{a}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\cdot\epsilon_\theta(\mathbf{a}_t, t) \tag{6}$$

where $\bar{\alpha}_{t-1}$ and $\bar{\alpha}_t$ are the cumulative noise schedule coefficients at time steps $t-1$ and t, respectively.

9 Experiments Details

9.1 Details of Tasks

Task 1: Pick and Place. Task 1 is a fundamental task that requires picking up a tennis ball on the table and placing it into a basket.

Task 2: Collect and Store. Task 2 focuses on the integrated capabilities of the system. Task 2 requires collecting three objects from the table and placing them into the basket in the following order: doll, broom, and basketball. For TypeTele, the operator can use voice commands to switch types for objects with different geometric shape during teleoperation.

Task 3: Handover. Task 3 evaluates the system's bimanual coordination capabilities and grasp robustness. In this task, the left hand is required to pick up a can from a stand and then hand it over to the right hand.

Task 4: Pouring from Pan. Task 4 requires stably grasping a pan and then pouring the contents of the pan into the basket. The difficulty of this task lies in the need for the hand to firmly grip the pan's handle to prevent tilting or dropping during the pouring process.

Task 5: Use Scissors. In Task 5, the right hand is required to hold a strip of paper while the left hand uses a pair of scissors to cut through it. The task is considered successful if the lower part of the paper strip is completely severed in a single cut.

Task 6: Spray Water. Task 6 requires grasping a spray bottle and then pressing the trigger to spray water. The task is considered successful if a stream of water is sprayed out.

Task 7: Use a Heavy Kettle. Task 7 evaluates the ability to operate under extreme weight. This task requires firmly gripping the handle of a watering kettle, lifting it, and then pouring water into a

bowl. A large water balloon is placed inside the kettle, bringing the total weight to over 2kg, which poses a significant challenge for maintaining a stable grasp.

Task 8: Opening a Large Box. Task 8 evaluates the ability to manipulate objects of significant size. In this task, the left hand is used to open a large box, followed by the right hand retrieving the object contained within. The short side of the box measures 15cm, demanding both a wide grasp span and precise force control.

Task 9: Grasp Two Objects. Task 9 aims to fully leverage the dexterity of the dexterous hand. This task requires using one hand to grasp two objects, first a water cup and then a cylinder. Both objects have a minimum diameter of 8cm, making it highly challenging to maintain a stable, simultaneous grasp with a single hand.

9.2 Additional Experiments

To further substantiate the capabilities and versatility of our proposed system, we conducted a series of supplementary teleoperation experiments. For a more comprehensive comparison within our original experimental setup, we introduced an additional retargeting-based baseline, AnyTeleop [7], which utilizes an optimization-based approach, evaluating it alongside the existing baseline ARCap [8], which employs an Inverse Kinematics (IK) methodology. Both are representative and prominent open-source baselines. To ensure a fair and rigorous evaluation, we strictly adhered to the official configurations of these baselines, employing their prescribed optimization and control parameters.

To assess the adaptability of our method to different hardware, we established a new experimental environment. This setup featured an Inspire Hand integrated with a Unitree G1 robot as the hardware platform, while using an Apple Vision Pro for motion capture. In this new context, we benchmarked our method against Open-television [6], a prominent retargeting framework developed for the Vision Pro, using its prescribed configuration to maintain comparative integrity.

Task	**LEAP Hand**						**Inspire Hand**			
	Success (%) ↑			T_{single} (s) ↓			**Success (%)**↑		T_{single} **(s)** ↓	
	TypeTele	[8]	[7]	TypeTele	[8]	[7]	TypeTele	[6]	TypeTele	[6]
Pick and Place	100	95.2	30	7.67	8.28	12.1	100	100	7.78	7.93
Collect and Store	95.2	60.6	–	29.4	37.3	–	–	–	–	–
Handover	95.2	80.0	30	11.6	18.4	21.1	100	100	16.9	17.3
Pouring from Pan	83.0	14.2	–	14.6	16.4	–	100	90	9.23	9.50
Use Scissors	91.1	–	–	5.37	–	–	100	80	9.56	12.5
Spray Water	86.9	–	–	7.28	–	–	100	90	6.25	6.33
Use a Heavy Kettle	85.0	–	–	18.5	–	–	–	–	–	–
Open a Large Box	95.2	–	–	19.0	–	–	100	60	7.04	9.42
Grasp Two Objects	69.6	–	–	21.2	–	–	–	–	–	–
Insert Stick	100	30	40	14.7	20.1	29.7	100	50	17.3	20.7
Press the Pump	60	–	–	6.87	–	–	–	–	–	–
Lift Blackboard	90	30	30	9.38	10.3	10.2	100	90	6.79	6.85
Error Recover	80	50		42.1	69.2					
Package Sorting	100	40		37.2	41.1					

Table 3: Performance evaluation on additional teleoperation experiments involving new tasks, additional baselines, and a different hardware setup.

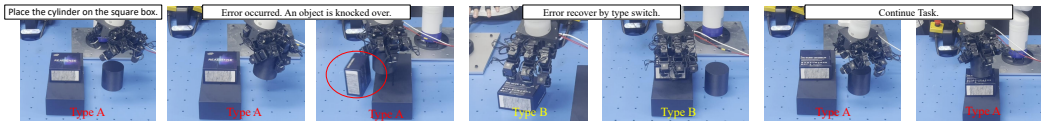

Figure 11: Recover form error by switching type dynamically.

Furthermore, to demonstrate the generality of our system, we expanded the scope of our evaluation to include five additional and distinct teleoperation tasks. **Insert Stick:** This task requires picking up a thin stick from a pen holder and inserting it firmly into a piece of plasticine nearby. The difficulty

of this task lies in the slender nature of the stick, which makes it challenging to grasp stably. Furthermore, the stick is prone to being dropped during insertion due to the resistance from the plasticine. **Press the Pump:** This task requires pressing a pump bottle with a single hand. The task is considered successful if the pump dispenses liquid. **Lift Blackboard:** This task requires lifting a small whiteboard standing on the table with one hand. The task is deemed successful if the whiteboard is held for five seconds without being dropped. **Error Recovery:** This task involves an object-stacking scenario, requiring the placement of a cylinder on top of two stacked cuboids. During the process of placing the cylinder, the top cuboid is intentionally knocked over. The recovery sequence requires the system to restore the fallen cuboid to its original position, and finally re-attempt the placement of the cylinder, as shown in Figure 11. This task presents a significant challenge to the response time of TypeTele, as the operator must issue a new voice command to the MLLM for type switching after the error occurs, which consumes additional time. **Package Sorting:** In this task, a total of six cubes and cylinders are presented to the teleoperator in a random sequence. The objective is to promptly sort these objects into their designated bins. The task is considered successful if all objects are sorted correctly without any being dropped during the process. This task evaluates the type switching speed of the TypeTele system.

The experimental results are summarized in Table 3. For each additional experiment, a total of ten independent trials were conducted in order to measure both the task success rate and the average demonstration duration T_{single} for each successful attempt. The symbol "–" denotes that the task could not be completed. Specifically, a task was considered unachievable if, after multiple teleoperators attempted it for 30 minutes, no successful trials were observed or all successful trials significantly exceeded the predefined time constraints. For short-duration tasks, the time limit was set to 3 minutes, while for long-duration tasks, the limit was 8 minutes. In such cases, the corresponding table entries are denoted by the symbol "–", indicating that the task could not be completed.

The experimental results unequivocally demonstrate the superior performance of our proposed method compared to all baselines, achieving consistently higher success rates while reducing task completion times. A more detailed analysis comparing the two dexterous manipulators offers further insight into our method's advantages. The LEAP Hand, possessing a greater number of Degrees of Freedom (DoF) than the Inspire Hand, inherently presents a more significant control challenge. This complexity proved detrimental to the baseline retargeting approach, under which the lower-DoF Inspire Hand successfully completed 8 tasks, whereas the more dexterous LEAP Hand completed only 6. This finding suggests that the baseline framework struggles to harness the capabilities of high-DoF systems. In stark contrast, our method effectively leverages the Leap Hand's advanced dexterity, enabling it to successfully complete a total of 14 tasks. This significant performance delta underscores our method's ability to unlock the full potential of high-DoF manipulators, translating their complex kinematic capabilities into tangible task success where conventional methods falter.

The reasons for task failure are diverse, stemming from both software and hardware limitations. In **Collect and Store**, the Inspire Hand's size proved inadequate for grasping a basketball with a single hand, as its physical dimensions and grasping envelope were insufficient to encompass the basketball. In **Pouring from Pan**, the baseline system failed to utilize the finger base for effective gripping of the pan's handle. This inadequacy led to slippage and rotation of the pan during the pouring motion, ultimately resulting in the dropping of either its contents or the pan itself. In **Use Scissors**, task demanded significant closure, which was a challenge for the baseline as it exhibited poor force modulation. This difficulty, combined with unstable manipulation control, resulted in frequent dropping of the scissors, thereby preventing task success. In **Spray Water**, the baseline encountered difficulties in reaching the spray bottle's trigger and lacked the precision required to apply force to press it, resulting in multiple failed attempts despite repeated trials. In **Use a Heavy Kettle**, the Inspire Hand successfully gripped the kettle but was unable to lift it due to the excessive weight, beyond the payload capacity of the Unitree G1 robot's arm. For the LEAP Hand, the baseline could not maintain a firm grip on the kettle's handle, causing it to slip or remain immobile, whereas the TypeTele system performed successfully by ensuring a firm hold, even under maximum load. In **Open a Large Box**, the Inspire Hand was pushed to its physical limits, with the baseline struggling

to apply sufficient force, leading to damage to the hand's linkages in some cases. The LEAP Hand, constrained by retargeting paradigm, was unable to extend its fingers sufficiently to grasp both sides of the box, leading to task failure. The **Grasp Two Objects** task was not achievable with the Inspire Hand due to its limited DoF and size constraints, while the LEAP Hand baseline struggled to grasp two relatively large objects, a task that is also non-trivial for the human hand. Despite attempts at wrapping and pinching, the baseline failed to successfully manipulate the objects. In **Insert Stick**, the primary failure arose from the stick's tendency to drop or not be securely inserted into plasticine. In **Press the Pump**, the Inspire Hand's low DoF prevented simultaneous grasping and actuation, while the LEAP Hand baseline could not leverage lateral DoF, leading to failed attempts and dropped objects. In **Lift Blackboard**, the baseline failed because the fingers could not achieve a sufficient surface contact with the whiteboard, resulting in slippage during the lifting or holding process. Detailed videos are available on our project webpage: https://isee-laboratory.github.io/TypeTele.

The last two tasks were designed to validate the efficiency and responsiveness of TypeTele. In **Error Recover**, while our approach necessitates an extra voice command post-error, incurring a few seconds of overhead, this is compensated for by the overall improvement in task execution efficiency. As a result, our method's final completion time was significantly shorter than the baseline's. In **Package Sorting**, the operator performs an initial one-time Type retrieval, allowing all subsequent Type switching to occur with minimal delay. Consequently, the introduction of Type did not increase the task completion time but did significantly boost the success rate.

9.3 User Study

We conducted additional experiments to further evaluate our teleoperation system through a user study involving five participants with varying levels of prior teleoperation experience. Each participant was instructed to complete an identical task (grasping the handle of a frying pan) using both the TypeTele system and a retargeting-based baseline. In order to mitigate the influence of learning effects, three participants used the TypeTele first, while the remaining two began with the baseline, and none were informed which system was the TypeTele and which was the baseline. Each system was tested in five trials per participant, during which we recorded Success Rate and Average Time per Success. The Average Time per Success is calculated by dividing the total time spent across all trials by the number of successful trials. This metric reflects the average amount of time required to obtain a single successful execution, capturing both task efficiency and failure overhead.

After completing all the tests, each participant completed a questionnaire that evaluated both systems in four dimensions: accuracy, responsiveness, ease of use, and user confidence. Each dimension was rated on a scale of 0-10. We then computed the average score for each dimension across all participants for both systems. The results are as follows:

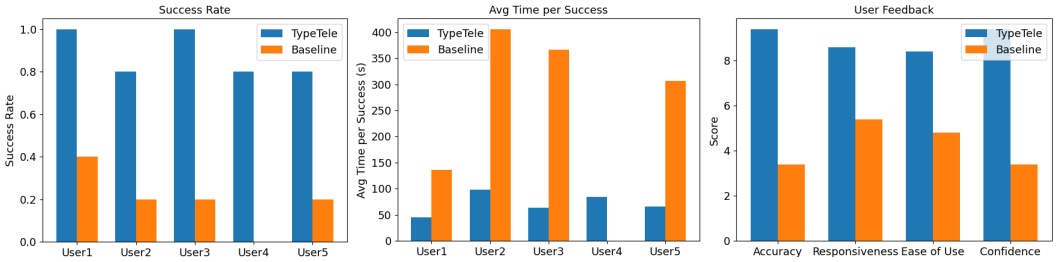

Figure 12: Results of User Study.

Experimental results demonstrate that the TypeTele system significantly outperforms the retargeting-based baseline across both objective and subjective measures. On average, TypeTele achieved a task success rate of 88%, compared to only 20% for the baseline. Participants also completed tasks faster using TypeTele. Subjective ratings further support these findings: TypeTele received higher scores across all four dimensions—accuracy (9.4 vs 3.4), responsiveness (8.6 vs 5.4), ease of use (8.4 vs

4.8), and user confidence (9.4 vs 3.4). These results indicate that TypeTele not only improves task performance but also delivers a more satisfying and trustworthy user experience.