MoDex: Planning High-Dimensional Dexterous Control via Learning Neural Hand Models

Tong Wu[1*], Shoujie Li[1*], Chuqiao Lyu[1], Kit-Wa Sou[1], Wang-Sing Chan[1], Wenbo Ding[1†]

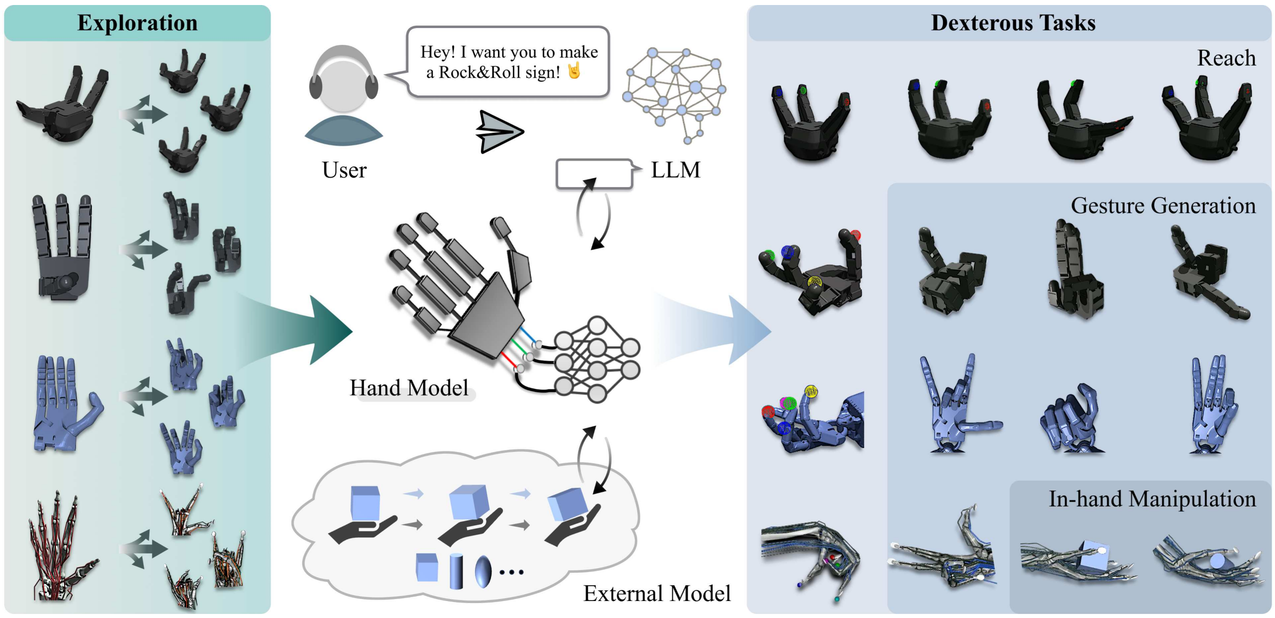

Fig. 1: Here we propose **MoDex**, a framework which learns neural hand models to represent various dexterous hands. We enable precise control in high-dimensional action space with a well-trained hand model, the generation of diverse gestures by linking the hand model with the LLM, and data-efficient in-hand manipulation via learning hierarchical dynamics model.

***Abstract*— Controlling hands in the high-dimensional action space has been a longstanding challenge, yet humans naturally perform dexterous tasks with ease. In this paper, we draw inspiration from the human embodied cognition and reconsider dexterous hands as learnable systems. Specifically, we introduce MoDex, a framework which employs a neural hand model to capture the dynamical characteristics of hand movements. Based on the model, a bidirectional planning method is developed, which demonstrates efficiency in both training and inference. The method is further integrated with a large language model to generate various gestures such as "Scissorshand" and "Rock&Roll." Moreover, we show that decomposing the system dynamics into a pretrained hand model and an external model improves data efficiency, as supported by both theoretical analysis and empirical experiments. Additional visualization results are available at https://tongwu19.github.io/MoDex.**

*Authors contributed equally. †Corresponding author.

[1]Shenzhen Ubiquitous Data Enabling Key Lab, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China. Emails: {wu-t23, lsj20}@mails.tsinghua.edu.cn, lyuchuqiao@163.com, {sujh21, chs22}@mails.tsinghua.edu.cn, and ding.wenbo@sz.tsinghua.edu.cn

This work was supported by Shenzhen Key Laboratory of Ubiquitous Data Enabling (No. ZDSYS20220527171406015), Shenzhen Science and Technology Program (JCYJ20220530143013030), Guangdong Innovative and Entrepreneurial Research Team Program (2021ZT09L197), Tsinghua Shenzhen International Graduate School-Shenzhen Pengrui Young Faculty Program of Shenzhen Pengrui Foundation (No. SZPR2023005), Shenzhen Higher Education Stable Support Program (WDZC20231129093657002), and Meituan.

I. Introduction

Driven by the need for more advanced robotic capabilities, research for dexterous hands has expanded, focusing on areas such as dexterous grasping [1]–[3], in-hand manipulation [4]–[7] and musculoskeletal manipulation [8], [9]. Yet, controlling hands in a high-dimensional action space remains a significant challenge, primary due to the large amount of data needed to train Deep Reinforcement Learning (DRL) policies for such manipulations tasks. This data demand arises from the vast exploration space created by the high action dimensions, coupled with the inefficient data utilization in current DRL methods, resulting in an unreasonbly large dataset needed to learn effective control strategies.

Reducing the dimensionality of the action space is one approach to mitigating this issue. Berg *et al.* [10] demonstrated this by identifying synergies which elucidate correlations among different dimensions, achieved through retaining the principal components. However, their method requires a residual component to refine performance, which paradoxically increases the dimensionality. Another solution is to optimize the use of collected data. As a result, Model-Based Planning (MBP) has been introduced to train dynamics models using online [11] or offline [12] algorithms for more efficient policy learning. Nevertheless, these well-

Manuscript 1480 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2024.

TABLE I: We investigate simulation models of four different dexterous hands, which cover a range of driving mechanisms (joint-driven and tendon-driven), actuation types (fully-actuated, under-actuated, and over-actuated), and structural complexities, reflected in their number of fingers (NoFs), degrees of freedom (DoFs), and action dimensions.

Name	Driven	Actuation	NoFs	DoFs	Dimension
Robotiq [16]	Joint-driven	Fully-actuated	3	11	11
Allegro [17]	Joint-driven	Fully-actuated	4	16	16
Shadowhand [17]	Joint-driven	Under-actuated	5	24	20
Myohand [18]	Tendon-driven	Over-actuated	5	23	39

trained models tend to characterize the dynamics of the entire specific manipulation systems, which greatly limits their transferability to different tasks.

In this paper, we reframed the dexterous hand as an independent system and propose that the hand itself should be studied through the lens of embodied cognition. This perspective is inspired by the remarkable ability of humans and animals to utilize *internal models* to control bodies with high Degrees of Freedom (DoFs) [13]. Specifically, we present a model-based approach to characterize the dexterous hand, utilizing a forward model for state prediction and an inverse model for decision-making. To accelerate the planning process, we bidirectionally integrate the hand model with a Cross-Entropy Method [14] (CEM) planner. The control performance of this approach was then validated on hand-tips reach tasks using simulation models of four well-known dexterous hands (as shown in Table I). Results show that our method is more data-efficient compared to traditional reinforcement learning methods. Additionally, the successful experiments on four different hands also highlight the high representational capacity of the neural hand model.

Furthermore, we showcase the versatility of our hand model by applying it to gesture generation and in-hand manipulation task. First, we combine the hand model with a Large Language Model (LLM) for text-conditioned gesture generation. This is achieved by prompting the LLM to produce a cost function of the hand state, which is then used to plan a corresponding action. Our approach successfully generated a variety of common gestures, such as “Scissors-hand” and “Rock&Roll” across multiple hands. For in-hand manipulation, we leverage the embodied cognition of the dexterous hand by decomposing the system dynamics into a hand model and an external dynamics model. In addition, an online algorithm [15] is applied, which iteratively collects data, learns the dynamics model and updates planning. Results from three different object reorientation tasks show that this hierarchical approach enables data-efficient learning. Our key contributions are as follows:

- We propose modelling dexterous hands with Neural Networks (NNs). The hand model consists of a forward model that depicts the forward dynamics and an inverse model that generates decision proposals. Our NN-based approach offers several advantages, such as high representational capacity and transferability.
- We further introduce a bidirectional framework for efficient control by integrating the hand model with CEM planning. The accuracy of our method is demonstrated through hand-tips reach tasks, evaluated on four dexterous hands in simulation.
- We combine a well-trained hand model with a large language model to generate gestures. These two modules are linked via prompting the LLM to yield cost functions for planning. Our experiments successfully generated a variety of gestures.
- To achieve data-efficient in-hand manipulation, we propose learning decomposed system dynamics models. Our method was tested on object reorientation tasks, showing superior data efficiency while maintaining high performance.

II. Related Works

A. Dexterous Manipulation

Dexterous manipulation has garnered significant attention, due to the remarkable performance demonstrated by model-free RL. Among the various manipulation tasks, prior works mainly focused on dexterous grasping, in-hand manipulation, and the emerging area of musculoskeletal manipulation. For instance, UniDexGrasp++ [19] introduces a multi-stage curriculum learning method for dexterous grasping, employing iterative geometrical curriculum learning learning and generalist-specialist learning. This method demonstrates outstanding performance across more than 3000 object instances. In contrast, DexVIP [20] utilizes human videos to generate rewards for functional grasping by estimating human pose priors. While controlling a dexterous hand for grasping is already challenging, in-hand manipulation further increases the difficulty by requiring the rotation or reorientation of objects within in the hand. Chen *et al.* [5] leverage observed point clouds to reorient novel objects using a three finger hand. Yin *et al.* [6] instead interpret sparse contact information as tactile observation to realize touch-based in-hand rotation. Yuan *et al.* [21] combine visual sensing with tactile sensing for defter in-hand manipulation. Building on these, musculoskeletal manipulation aims to mimic human-like dexterity by introducing musculoskeletal models, which further increases the action dimension. To reduce the complexity of the action space, Berg *et al.* [10] propose using principal component analysis and independent component analysis to identify the synergies in dexterous manipulation. To enhance generalizability in high-dimensional tasks, Caggiano *et al.* [8] follow a multi-task learning procedure to grasp diverse objects with a musculoskeletal hand.

Although model-free RL has demonstrated superior performance on dexterous manipulation, alternative models are being considered to tackle these tasks with less data. Instead of directly training an RL policy for grasping in an end-to-end manner, Unidexgrasp [2] learns a grasp proposal model and leverages the grasp proposals as the input of the RL policy. They employ DexGraspNet [1] to generate a large number of high-quality grasps with accelerated DFC [22]. Other methods use a contact model to improve generalizability. GenDexGrasp [16] and ContactGrasp [23] decompose dexterous grasping into two stages: they utilize a contact model to provide contact map and a hand model for grasping optimization. However, these methods assume a known hand model. On the contrary, Nagabandi *et al.* [11] propose to learn a neural network model for the entire manipulation system with a random shooting MPC controller. Furthermore, Nagabandi *et al.* [15] improve the model learning algorithm and apply it to dexterous manipulation. While these two methods bypass the use of predefined hand models, the learned dynamics model is limited to specific hands and objects. Rather than relying on a predefined hand model or modelling the entire system dynamics, we concentrate on learning a general internal model to represent the hand's behaviors, which can then be applied to downstream dexterous tasks.

B. Internal Model

The internal model was first introduced by Francis *et al.* [24] to explicitly model the control system. The human body, as a highly dexterous physiological system, exhibits the signs of maintaining an internal model [13], [25]. Kawato *et al.* [26] suggest that an accurate internal model constructed from previous perception is the primary information for motor planning, since observations from environment are not always available and reliable. For example, occlusions may occur when humans rely on vision for observation. As a support, Wolpert *et al.* [27] found that even without vision, humans can still estimate the location of hands after movements. The internal model consists of a forward model for predicting the results of control signals and an inverse model for generating the control proposals of given targets. Motor adaptation experiments [28] provide evidences for both models: when humans are perturbed by external force, they can still control their hands to follow the target trajectories by vision supervision. In this work, we propose to learn neural networks as internal models to plan high-dimensional dexterous control, mimicking the physiological processes in human body.

III. Method

A. Learning Neural Hand Models

Controlling a hand with high degrees of freedom is an exceptionally complex task, yet humans manage it effortlessly, indicating the use of internal model planning [29]. This phenomenon highlights the importance of modeling the body system. Accordingly, we propose to learn a neural hand model for high-dimensional dexterous control. The neural hand model includes:

1) A Forward Model for Hand State Prediction: The forward model predicts the next hand state based on the command of actuators and the current hand state, expressed as follows:

$$\hat{\boldsymbol{s}}_{t+1} = f_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t), \tag{1}$$

where $\boldsymbol{a_t} \in \mathbb{R}^K$ represents the action generated by actuators with a dimension of K, $\boldsymbol{s_t} \in \mathbb{R}^H$ denotes the current hand state and $\hat{\boldsymbol{s}}_{t+1} \in \mathbb{R}^H$ denotes the predicted next hand state.

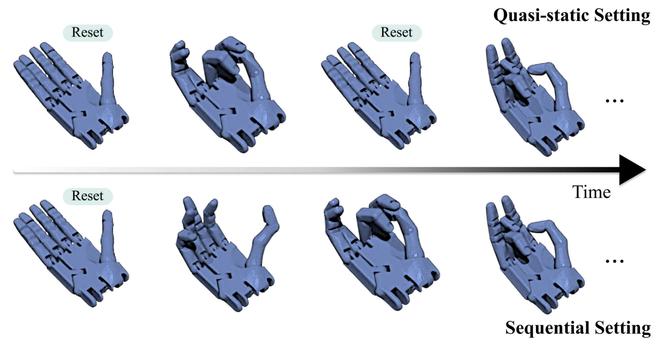

Fig. 2: **Quasi-static v.s. Sequential.** We distinguish dexterous control tasks into two settings. **Quasi-static setting** focuses solely on the final outcome and is formulated as a single-step task. In contrast, **sequential setting** considers intermediate states and is formulated as a multi-step task.

The forward function, $f_\theta(\cdot)$, is defined by learnable parameters θ. While the formula is straightforward, it assumes that the mapping from inputs to the next hand state can be expressed as a function. This assumption is both intuitive and has been shown to be practical in our experiments for robotic hands as well as bionic hands, such as Myohand [18].

In the field of dexterous control, sequential tasks like in-hand manipulation are a major area of focus. However, non-sequential tasks, where the static result is more important than the dynamic process, also play a crucial role. Examples include gesture generation and dexterous grasping. To address these quasi-static dexterous control tasks, we also propose a quasi-static forward model which disregards the current hand state $\boldsymbol{s_t}$ and utilizes only the action to predict the subsequent hand state. Figure 2 illustrates both settings.

2) An Inverse Model for Action Proposal: The inverse model works in reverse to the forward model, providing approximate action proposals to reach a target state. This process is formulated as:

$$\boldsymbol{a} \sim g_\phi(\cdot|\boldsymbol{s}_t, \boldsymbol{s}_T), \tag{2}$$

where g denotes the inverse probability distribution with learnable parameters ϕ and $\boldsymbol{s}_T \in \mathbb{R}^H$ denotes the target hand state. Predicting accurate actions is highly variable, as the accuracy depends on the information available in the hand state. Additionally, the mapping from the input to the control signal might not be a function, resulting in multiple possible action solutions for a single target. To address this, we introduce a rough inverse model which represents action proposals as distributions rather than deterministic values. In implementation, we use the output of the inverse model as the mean, and the average absolute error as the standard variance, to form a Gaussian distribution. This distribution serves as the initial set of action proposals, which are subsequently refined as elaborated in the following section.

Similarly, to address quasi-static situations, we generate action proposals directly from the target hand state, bypassing the consideration of the current hand state.

3) Learning via exploration: To learn a neural hand model, we collect data through exploration. While teleopera-

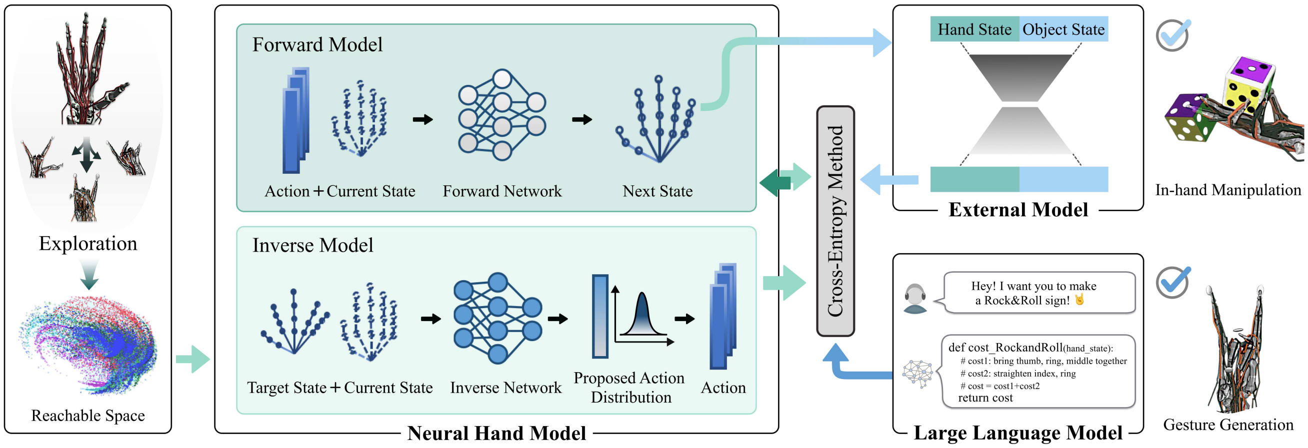

Fig. 3: **Framework. Firstly**, we explore the action space and collect dynamics data. Afterwards, a forward model is trained to predict the next state based on the current state and the proposed action, while the inverse model provides a distribution of action proposals to achieve the target state. Using the two models, we apply CEM to plan the optimal action. **Secondly**, to generate gestures from an action perspective, we utilize LLM to provide a cost function according to the textual inputs. The cost function is used to optimize actions which produce the desired gesture. **Thirdly**, we decompose the system dynamics model into a hand model and an external model, which improves the learning efficiency in in-hand manipulation tasks.

tion and pre-coding are viable alternatives, random or policy-based exploration implicitly captures the hand's reachability within task-specific settings. In quasi-static scenarios, we reset the hand to its default state and record the subsequent state following the executed action. In sequential scenarios, we reset the hand and gather trajectories of transitions. During the training stage, we use MSE loss for the quasi-static setting. In the sequential setting, we employ multi-step MSE loss for the forward model and MSE loss for the inverse model.

B. Bidirectional Planning Strategy

Planning in the high-dimensional action space is time-consuming for iteration-based methods due to the vast size of the action space. To efficiently control a high-DoF hand at high frequency, we draw inspiration from neurobiology, where the central nervous system (CNS) decomposes a complex movement into an initial movement and small sub-movements [30], [31]. The initial movement may be inaccurate but must be generated quickly, while the sub-movements adjust for any errors in the initial movement.

Our bidirectional planning strategy follows a similar approach. Initially, we utilize the learned inverse model to generate a distribution which provides rapid, albeit imprecise, action proposals. In the quasi-static setting the distribution serves as the initial sampling distribution for the CEM planner, while in the sequential setting we utilize the mean as the first inaccurate action. During refinement, CEM planner continuously interacts with the forward model to attain better action samples. We also employ MPC [11] for the sequential setting which updates planning based on current observations. Although the actions generated by inverse model may be inaccurate, the distribution still provides sufficient information to accelerate planning, as demonstrated in Section IV-A.

C. Language-Based Gesture Generation

With a well-trained hand model in place, we could apply it to various tasks. One particularly intriguing yet under-explored task is gesture formation based on language inputs from an action perspective. In this section, we formulate this as a quasi-static problem and propose to combine the hand model with a large language model to achieve language-based gesture generation.

To generate a gesture, we first present the LLM with a linguistic gesture request and obtain a heuristic cost function of the hand state, where the cost decreases as the hand state becomes more similar to the target gesture. In this work, we use fingertip positions as the primary representation of the hand state as they offer a straightforward yet effective means of producing diverse gestures. For instance, to generate an "OK" gesture with a five finger hand, the cost function can be defined as follows:

$$\mathcal{J}(\boldsymbol{s}) = -\|\boldsymbol{s}_0 - \boldsymbol{s}_1\|_2 + \sum_{i>2} (\boldsymbol{s}_i \cdot \hat{\boldsymbol{n}}_i), \tag{3}$$

where $\boldsymbol{s}_i$ represents the position of ith fingertip, and $\hat{\boldsymbol{n}}_i$ denotes the direction for straightening each finger. This formulation forces the thumb and the index finger to come close together while encouraging the middle finger, the ring finger, and the little finger to remain straight. To enable the LLM to generate such cost functions, we provide several examples in system prompts, serving as in-context learning [32]. Subsequently, the learned hand model is employed to optimize the action using our proposed planning strategy, with the cost function as the objective.

D. Decomposed Dynamics Learning

While quasi-static setting can represent a part of dexterous control tasks, sequential setting is more prevalent in practice. In most cases, traditional methods consider the hand and the external environment as a whole system, which we found

TABLE II: We evaluate our method on hand-tips reach tasks in quasi-static setting for 100 episodes. **S.R.**: Success Rate(%) ↑; **R.E.**: Reach distance Error per finger(cm) ↓; **P.S.**: Planning Samples ↓. The successful thresholds for Robotiq&Allegro&Shadowhand and Myohand are 0.6cm and 1.25cm respectively. We note that RL fails in this task since it is suitable for sequential tasks.

Methods	Robotiq			Allegro			Shadowhand			Myohand		
	S.R.	R.E.	P.S.	S.R.	R.E.	P.S.	S.R.	R.E.	P.S.	S.R.	R.E.	P.S.
RS	53	0.58	–	2	1.12	–	0	1.44	–	15	1.97	–
SAC	16	0.83	–	0	2.01	–	0	2.12	–	3	3.41	–
FM+RS	**100**	0.25	10k	57	0.61	10k	20	0.72	10k	33	1.77	10k
FM+BGD	**100**	0.24	3.2k	**79**	**0.50**	3.2k	41	0.63	3.2k	37	1.69	3.2k
FM+CEM	**100**	0.24	2.4k	73	0.52	2.4k	47	0.60	2.4k	74	1.03	2.4k
Ours	**100**	**0.22**	**1.6k**	76	0.51	**1.6k**	**49**	**0.59**	**1.6k**	**80**	**0.93**	**1.6k**

inefficient since the dexterous hand and the external environment are highly independent. Therefore, in this section, we propose a hierarchical method, which decomposes the system dynamics into a hand model and an external dynamics model.

When executing an action, we first apply the hand model to predict the hand state transition, formulated in Equation 1. The hand model is pretrained since humans can obtain it from past experiences. Next, we take as input the predicted hand state and the current external state to predict the next hand state and the next external state using an external dynamics model:

$$\boldsymbol{s}_{t+1}, \boldsymbol{x}_{t+1} = f_\psi(\hat{\boldsymbol{s}}_{t+1}, \boldsymbol{x}_t), \tag{4}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ denote the current external state and the next external state, while $\hat{\boldsymbol{s}}_{t+1}$ and $\boldsymbol{s}_{t+1}$ denote the predicted hand state by hand model and the predicted hand state after interacting with external world. f_ψ is a learnable model which depicts the intermediate dynamics. If we express the system dynamics as one whole model, the input-output dimension is $(H + O + K) \times (H + O)$, where H, O and K are the dimensions of hand state, object state and action respectively. By adopting the hierarchical approach, the input-output dimension is reduced to $(H+O) \times (H+O)$, which directly eliminates the action dimension of the input. However, since pretrained hand models may lack accuracy, we opt to train the hierarchical model in an end-to-end manner. To learn the dynamics model, we employ an online algorithm [15] that iteratively rolls out, collects dynamics data and updates the model. A multi-step loss function is used to facilitate long-term prediction.

IV. Experiments

A. Hand-Tips Reach

In this stage, we evaluated our method for controlling four different dexterous hands with high degrees of freedom in both quasi-static and sequential settings. The evaluation is based on the hand-tips reach tasks, which require the fingertips to reach target positions. We compare our method with four baselines: a model-free method, Soft Actor-Critic [33] (**SAC**) trained with 1M data points, and three model-based methods—Random Shooting (**FM+RS**), Batch Gradient Descent (**FM+BGD**), and Cross-Entropy Method (**FM+CEM**)—each trained on 10K/100K data points (quasi-static/sequential). Additionally, we examine the **RS** method, which randomly interacts with the environment 1K/10K times, to highlight the challenge of each experiment. As shown in Table II and Table III, our method outperforms the baselines in most cases across different dexterous hands and settings. Although our method exhibits slightly lower accuracy in two cases, it reduces planning samples by over 30%, indicating significant acceleration during inference. Meanwhile, our method utilizes only 1%/10% of the data required by model-free methods, demonstrating much higher data efficiency.

TABLE III: We evaluate our method on hand-tips reach tasks in sequential setting for 100 episodes. **S.R.**: Success Rate(%) ↑; **R.E.**: Reach distance Error per finger(cm) ↓; **P.S.**: Planning Samples per step ↓. The successful thresholds for Robotiq&Allegro&Shadowhand and Myohand are 0.8cm and 1.25cm respectively.

Methods	Robotiq			Allegro			Shadowhand			Myohand		
	S.R.	R.E.	P.S.	S.R.	R.E.	P.S.	S.R.	R.E.	P.S.	S.R.	R.E.	P.S.
RS	89	0.53	–	5	1.43	–	0	1.71	–	17	2.67	–
SAC	**100**	0.53	–	47	1.03	–	33	1.11	–	62	1.52	–
FM+RS	95	0.52	50k	8	1.28	50k	0	1.53	50k	22	2.25	50k
FM+BGD	**100**	0.57	3.2k	36	1.16	3.2k	25	1.18	3.2k	68	1.49	3.2k
FM+CEM	**100**	**0.51**	2.4k	53	0.94	2.4k	32	1.11	2.4k	71	1.40	2.4k
Ours	97	**0.51**	**1.6k**	**61**	**0.89**	**1.6k**	**44**	**1.02**	**1.6k**	**78**	**1.29**	**1.6k**

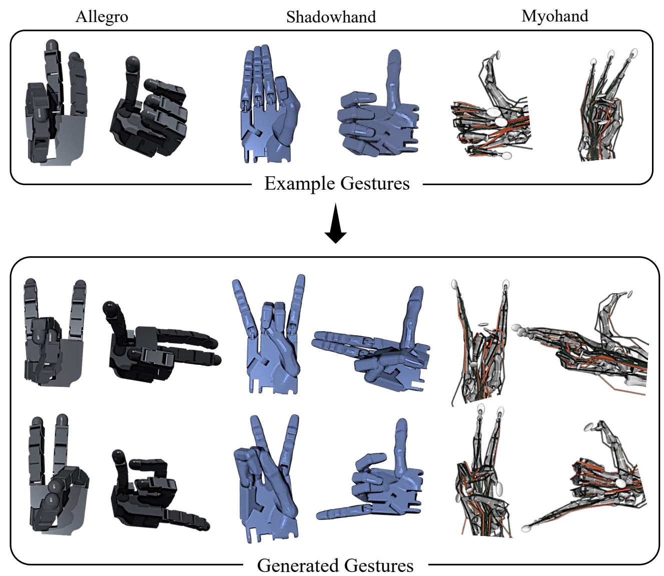

Fig. 4: **Gesture Generation Experiment.** We generate common gestures from action perspective. **Top:** gestures provided in LLM prompts; **Bottom:** generated gestures.

B. Gesture Generation

In this section, we employed GPT-4 as the core model for generating cost functions and conducted a study on gesture generation. We directly applied a hand model trained on 10K data points from random exploration, to bidirectional

Manuscript 1480 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2024.

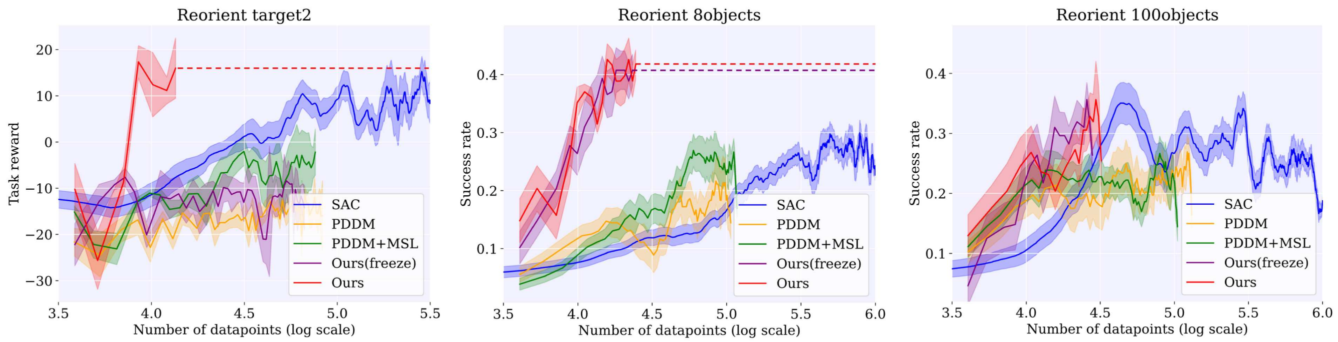

Fig. 5: **In-Hand Manipulation Experiments. Reorient Target2** sets 2 poses ($\pm\frac{\pi}{2}$ in z axis) as targets; **Reorient 8objects** reorients 8 random objects to random poses within a given range. **Reorient 100objects** reorients 100 random objects to random poses within a given range. The task reward for the first task and the success rates for the last two tasks are reported.

planning for attaining target actions. The generated gestures are illustrated in Figure 4. Despite the intuitive nature of the generation process, integrating the large language model with the proposed hand model enables the creation of diverse gestures, such as "Rock&Roll," "Scissorhands," "Finger Gun," and "Calling." Our method demonstrates its representational capabilities by being applicable to various types of robotic hands. Notably, the successful generation of gestures via employing a well-trained hand model suggests that the hand model can be transferable across different dexterous manipulation tasks.

C. *In-Hand Manipulation*

In this experiment, we apply decomposed dynamics learning to in-hand manipulation tasks using a hand model pretrained with 100K data points. For comparison, we select three baselines: Soft Actor-Critic (**SAC**) to represent model-free reinforcement learning; **PDDM** [15] which learns a dynamics model for the whole manipulation system; **PDDM+MSL** which enhances PDDM with a multi-step loss for longer-term prediction. We evaluate our method in two settings: one where the hand model is frozen during training, and another where it was trained in an end-to-end manner. As shown in Figure 5, our method (unfrozen) outperforms all baselines across three tasks while utilizing significantly less data than model-free RL. Although other model-based methods are more data-efficient than model-free RL, they do not achieve the same performance as ours. Additionally, we found that freezing the hand model introduces large prediction errors, leading to the failure in the first task.

D. *Ablation: Impact of Action Dimension and Data Volume*

In previous experiments, we observed that the inverse model primarily impacts planning efficiency, whereas the forward model is crucial for the accuracy of dexterous control. Therefore, this section focuses on investigating two factors affecting the forward model's performance: action dimension and data volume. The results, shown in Figure 6, indicate that increasing data volume improves the forward model's prediction accuracy. However, higher action dimensions tend to result in higher prediction errors. These findings suggest that augmenting dataset might mitigate the adverse effects associated with higher action dimensions, thereby enhancing the accuracy.

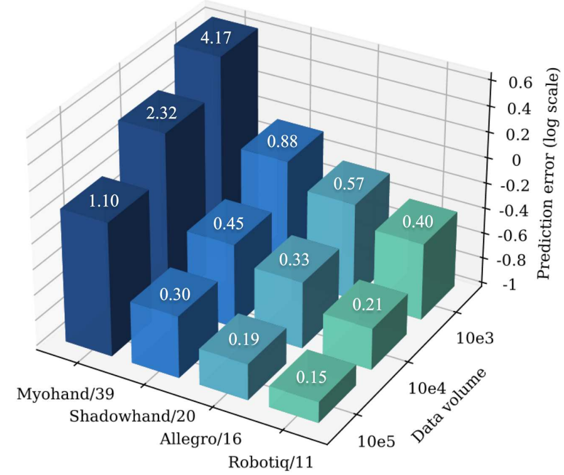

Fig. 6: **Ablation Study.** We investigate the behavior of prediction errors in quasi-static setting across varying action dimensions and data volumes. Absolute values are stick on top of each bar for direct comparison and log10 values are reported as the heights of the bars for better visualization.

V. Conclusion

In this work, we propose a neural network-based internal model for controlling a dexterous hand in high-dimensional action space. Integrated with a CEM planner, our approach achieves high-accuracy control across four different dexterous hands in hand-tips reach tasks. Furthermore, we combine the hand model with an LLM for gesture generation and with an external model for data-efficient in-hand manipulation. While our method successfully accomplishes several dexterous tasks in simulation, representing fingertip positions as hand state may complicate sim-to-real transfer. To facilitate real-world deployment of our method, future work should focus on more pragmatic settings and utilize raw observation, such as point clouds, to represent the hand state.

References

[1] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, "DexGraspNet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 359–11 366.

[2] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen, T. Liu, L. Yi, and H. Wang, "UniDexGrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 4737–4746.

[3] P. Mandikal and K. Grauman, "Learning dexterous grasping with object-centric visual affordances," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6169–6176.

[4] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam *et al.*, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5977–5984.

[5] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, "Visual dexterity: In-hand reorientation of novel and complex object shapes," *Science Robotics*, vol. 8, no. 84, p. eadc9244, 2023.

[6] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang, "Rotating without seeing: Towards in-hand dexterity through touch," *Robotics: Science and Systems (RSS)*, 2023.

[7] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," in *Conference on Robot Learning (CoRL)*, 2023, pp. 2549–2564.

[8] V. Caggiano, S. Dasari, and V. Kumar, "Myodex: a generalizable prior for dexterous manipulation," in *International Conference on Machine Learning (ICML)*, 2023, pp. 3327–3346.

[9] V. Caggiano, H. Wang, G. Durandau, M. Sartori, and V. Kumar, "Myosuite: A contact-rich simulation suite for musculoskeletal motor control," in *Learning for Dynamics and Control Conference (L4DC)*, 2022, pp. 492–507.

[10] C. Berg, V. Caggiano, and V. Kumar, "Sar: Generalization of physiological agility and dexterity via synergistic action representation," *Robotics: Science and Systems (RSS)*, 2023.

[11] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7559–7566.

[12] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu, "Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 533–549, 2024.

[13] S. W. Egger, E. D. Remington, C.-J. Chang, and M. Jazayeri, "Internal models of sensorimotor integration regulate cortical dynamics," *Nature Neuroscience*, vol. 22, no. 11, pp. 1871–1882, 2019.

[14] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. Springer, 2004, vol. 133.

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

[16] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang, "Gendexgrasp: Generalizable dexterous grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8068–8074.

[17] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac Gym: High performance GPU based physics simulation for robot learning," in *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*, 2021.

[18] H. Wang, V. Caggiano, G. Durandau, M. Sartori, and V. Kumar, "Myosim: Fast and physiologically realistic mujoco models for musculoskeletal and exoskeletal studies," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8104–8111.

[19] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, and H. Wang, "UniDexGrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning," in *International Conference on Computer Vision (ICCV)*, 2023, pp. 3891–3902.

[20] P. Mandikal and K. Grauman, "Dexvip: Learning dexterous grasping with human hand pose priors from video," in *Conference on Robot Learning (CoRL)*, 2021, pp. 651–661.

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

[22] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, "Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 470–477, 2021.

[23] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, "Contactgrasp: Functional multi-finger grasp synthesis from contact," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2386–2393.

[24] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[25] D. E. Angelaki, A. G. Shaikh, A. M. Green, and J. D. Dickman, "Neurons compute internal models of the physical laws of motion," *Nature*, vol. 430, no. 6999, pp. 560–564, 2004.

[26] M. Kawato, "Internal models for motor control and trajectory planning," *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 718–727, 1999.

[27] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan, "An internal model for sensorimotor integration," *Science*, vol. 269, no. 5232, pp. 1880–1882, 1995.

[28] R. Shadmehr and F. A. Mussa-Ivaldi, "Adaptive representation of dynamics during learning of a motor task," *Journal of Neuroscience*, vol. 14, no. 5, 1994.

[29] M. Ito, "Control of mental activities by internal models in the cerebellum," *Nature Reviews Neuroscience*, vol. 9, no. 4, pp. 304–313, 2008.

[30] R. S. Woodworth, "Accuracy of voluntary movement." *The Psychological Review: Monograph Supplements*, vol. 3, no. 3, p. i, 1899.

[31] D. Elliott, W. F. Helsen, and R. Chua, "A century later: Woodworth's (1899) two-component model of goal-directed aiming." *Psychological bulletin*, vol. 127, no. 3, p. 342, 2001.

[32] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," in *Conference on Robot Learning (CoRL)*, 2023, pp. 540–562.

[33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning (ICML)*, 2018, pp. 1861–1870.

Manuscript 1480 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2024.