IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.
Cite as T-ASE paper.

Language-Guided Dexterous Functional Grasping by LLM Generated Grasp Functionality and Synergy for Humanoid Manipulation

Zhuo Li, Junjia Liu, Zhihao Li, Zhipeng Dong, Tao Teng, Yongsheng Ou, *Senior Member, IEEE*, Darwin Caldwell, *Fellow, IEEE* and Fei Chen, *Senior Member, IEEE*

***Abstract*—Dexterous Functional Grasping (DFG) is the crucial first step for humanoid robots to perform generalized manipulation tasks. However, enabling robots to learn language-guided DFG skills in real-world environments presents several challenges, including comprehending the complex relationship between task instructions and grasp functionality, generating feasible functional grasps of dexterous hands, and handling generalization for novel functional concepts. To tackle these challenges, we introduce SayFuncGrasp, a Large Language Model (LLM) based DFG framework that can synthesize versatile dexterous functional grasps from language instructions and achieve generalization on novel functional concepts. SayFuncGrasp first harnesses the open-ended manipulation knowledge from an LLM to infer grasp functionality based on language instructions. Subsequently, it employs the inferred grasp functionality to synthesize plausible DFG actions characterized by hand synergies. Simulation experiments show that SayFuncGrasp significantly outperforms the baseline method in open-set grasp functionality generalization. Real robot experiments demonstrate the effectiveness and generalizability of SayFuncGrasp for interactive humanoid manipulation tasks, achieving an overall grasp success rate of 64.66% and a manipulation success rate of 70.41%.**

***Note to Practitioners*—This research was motivated by the practical challenge of enabling humanoid robots with high-DoF dexterous hands to perform functional grasping based on verbal instructions. In industrial settings, such capabilities can significantly enhance the versatility and adaptability of humanoid assistants, allowing them to perform complex manipulations simply by being told what to do, thereby reducing programming complexity and increasing flexibility. Current dexterous functional grasping methods rely solely on visual input, without the ability to process language instructions. Furthermore, they are restricted to pre-defined functional concepts and cannot be generalized to novel object classes and manipulation tasks within natural language. Our newly proposed language-guided dexterous functional grasping system takes advantage of open-ended manipulation knowledge from LLMs to produce generalized functional grasps of dexterous robot hands according to verbal commands. Our experiment results demonstrate improved versatility and generalizability compared to the state-of-the-art.**

***Index Terms*—Dexterous functional grasping, Language-guided robot manipulation, large language models, hand synergies.**

*This work is supported in part by the Research Grants Council of the Government of the Hong Kong SAR via the Grant 24209021, 14213324, C7100-22GF and in part by the InnoHK of the Government of the Hong Kong SAR via the Hong Kong Centre for Logistics Robotics. *(Corresponding author: Fei Chen.)*

Zhuo Li, Junjia Liu, Zhihao Li, Zhipeng Dong, Tao Teng and Fei Chen are with the Department of Mechanical and Automation Engineering, T-Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong (e-mail: zli@mae.cuhk.edu.hk; jjliu@mae.cuhk.edu.hk; zhihaoli@mae.cuhk.edu.hk; zhipengdongneu@gmail.com; tao.teng@ieee.org; f.chen@ieee.org).

Yongsheng Ou is with the Department of Control Science and Engineering, Dalian University of Technology, Dalian, China (e-mail: yoo2023@dlut.edu.cn).

Darwin Caldwell is with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy (e-mail: darwin.caldwell@iit.it).

I. Introduction

DEVELOPING general-purpose humanoid robots capable of understanding and physically executing natural language instructions is a longstanding vision of robotics and artificial intelligence [1]. With this capability, one could verbally instruct humanoid assistants to handle diverse manipulation tasks, such as "Use the spray to clean a dish" and "Unscrew the bottle cap." The crucial first step in completing these tasks is to grasp the intended object with the dexterous hand in a functional manner under the guidance of language instructions, i.e., language-guided DFG skill. While humans demonstrate a remarkable proficiency in this skill, endowing robots with a comparable capability presents three main challenges, including (a) understanding the intricate interconnection between language instructions and grasp functionality, (b) synthesizing functional grasps for high-DoF dexterous hands, (c) generalizing the learned skill to novel functional concepts.

Learning language-guided grasping skills has garnered significant interest due to its potential to enhance the adaptability and flexibility of robot systems in interactive operating scenarios [2]–[5]. Ahn et al. [2] integrated high-level task commands with the affordance functions of pre-trained skills to enable language-guided mobile grasping. Namasivayam et al. [3] proposed to map language commands into disentangled robot actions via a neuro-symbolic manipulation model and demonstrate the generalizability in long-horizon tasks. However, existing methods mainly target performing basic grasping tasks (e.g., pick and place) with low-DoF parallel grippers, overlooking the intricate challenge of dexterous functional grasping that involves humanoid manipulation.

Current DFG methods focus on generating functional grasps with annotated grasp functionality, which can be categorized into contact-based methods [6]–[8] and interaction-based methods [9]–[12]. Contact-based methods characterize grasp functionality through object contact maps extracted from human demonstration, so as to reveal a specific mapping between object shapes and functional grasps. Interaction-based methods encode detailed interactive information between objects

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

and robot hands for grasp functionality representation, such as knuckle-level hand-object representation [9] and semantic touch codes [12]. Despite the achievements, these methods have limitations in achieving language-guided DFG skills. One major limitation is that they are confined to pre-defined grasp functionality (e.g., semantic touch codes), limiting their generalizability to novel functional concepts within natural language. Additionally, these methods focus on object-centric functional grasp synthesis (i.e., generating grasp actions based solely on the inherent functional regions of the object shape) while ignoring the task designations beyond objects. As a result, they are incapable of producing versatile functional grasps for the intended object according to different task instructions. Moreover, these methods typically explore feasible grasp configurations in full-dimensional joint space, which leads to considerable learning complexity owing to the high-DoF nature of dexterous hands.

Recent advances in LLMs have brought new insights to address these limitations. These LLMs are trained on massive text corpora that encapsulate extensive human commonsense, thereby exhibiting superior language comprehension and knowledge generalization abilities in a variety of robot manipulation tasks [13]–[16]. Meanwhile, we observe two pivotal traits central to the excellence of human functional grasping behavior. First, humans intuit grasp functionality based on their prior manipulation experience by resolving two fundamental queries: where (i.e., grasp affordance) and how (i.e., grasp type) to grasp, rather than endeavoring to delineate intricate contact maps (see figure 1). Second, they perform functional grasps in a sequential manner by first approaching a task-oriented palm pose indicated by grasp affordance, then synergistically adapting the hand configuration to form the desired grasp type.

Inspired by the above-mentioned traits, we present SayFuncGrasp, a language-guided DFG framework that can synthesize versatile dexterous functional grasps with hand synergies based on task instructions and generalize to novel functional concepts. SayFuncGrasp first utilizes a pre-trained LLM to provide prior manipulation knowledge for inferring grasp affordance and type from the instructions. Subsequently, it introduces a task-oriented grasp pose generation module to predict functional palm poses based on the inferred grasp affordance. With the predicted palm pose and the inferred grasp type, a synergy-based functional grasping policy is proposed to generate fine DFG actions by exploring grasp configurations in low-dimensional hand synergy space. An overview of SayFuncGrasp is presented in figure 2.

Compared with existing methods, SayFuncGrasp enables language-guided DFG skill learning and attains generalization toward novel functional concepts by leveraging the open-end manipulation knowledge from LLMs. In addition, it incorporates task designation into the grasp inference process by characterizing grasp functionality as object-centric grasp affordance and task-centric grasp type, thus enabling the synthesis of versatile grasp actions adapted to different manipulation tasks. Furthermore, SayFuncGrasp generates functional grasp actions in a synergistic manner rather than optimizing each joint individually, which improves learning efficiency and grasping robustness. In summary, the contributions of this paper are as follows.

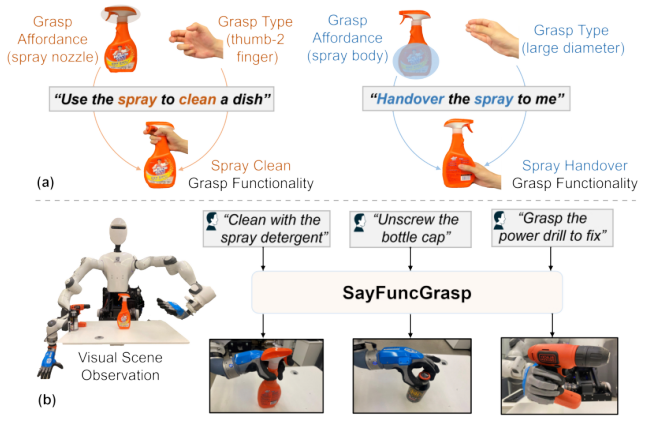

Fig. 1. (a) Examples of language-guided DFG skills exhibited in human daily manipulation behavior. (b) Given language instructions and visual observation, SayFuncGrasp can efficiently synthesize versatile dexterous functional grasps for diverse humanoid manipulation tasks.

- A language-guided DFG framework SayFuncGrasp is proposed, which exploits the open-ended manipulation knowledge from LLMs to synthesize versatile functional grasps of dexterous hands from natural language instructions and generalize to novel functional concepts.
- A synergy-based functional grasping policy is proposed to efficiently generate fine DFG actions by exploring grasp configurations in the low-dimensional hand synergy space.
- Real robot experiments demonstrate the effectiveness and generalizability of SayFuncGrasp in performing various interactive humanoid manipulation tasks, with an overall grasp success rate of 64.66% and a manipulation success rate of 70.41%.

II. Related Work

A. Dexterous Functional Grasping

Dexterous functional grasping refers to the problem of generating grasp configurations for dexterous hands that are not only stable, but also facilitate post-grasp manipulation. Early DFG methods try to solve this by analyzing the contact information between objects and human hands [6]–[8]. ContactDB [6] was the first to analyze object contact maps during human functional grasping. They proposed a large-scale dataset containing extensive object contact maps that show the functional areas where the human hand tends to touch. Further, ContactGrasp [7] presented a sample-and-rank functional grasp synthesis framework based on these contact maps. Since the contact map is derived from human grasp, it cannot provide precise guidance for robot hands, resulting in misalignment between the hand segment and the object. To address this problem, recent studies aim to enhance the contact map with hand-object interaction information [9]–[12]. Wei et al. [9] proposed a knuckle-level hand-object contact representation that associates each knuckle of a robotic hand with a corresponding object functional region. Zhu et al.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

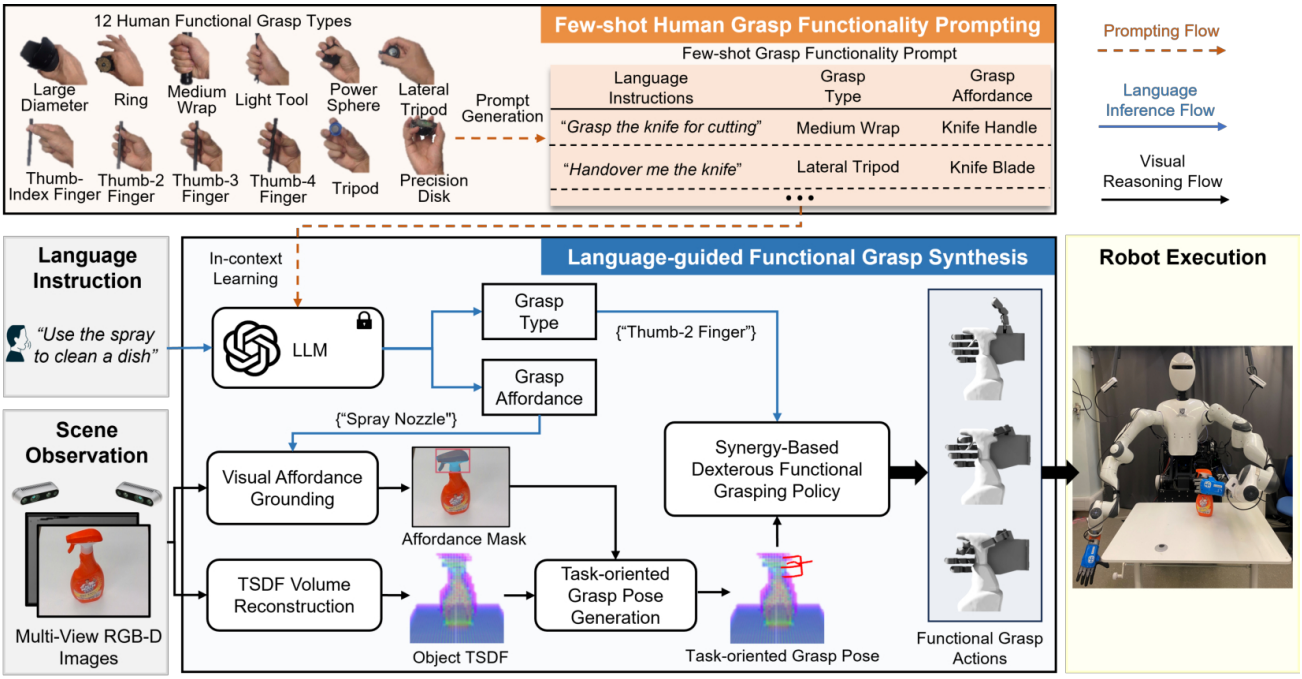

Fig. 2. An Overview of SayFuncGrasp framework. In the few-shot prompting stage, a grasp functionality prompt is generated based on human grasping knowledge to learn an LLM in context. In the grasp synthesis stage, SayFuncGrasp first leverages the prompted LLM to infer grasp functionality from the language instruction, encompassing both grasp type and affordance. Subsequently, it employs the inferred grasp functionality to generate synergy-based DFG actions suitable for task execution.

[12] developed a semantic code for grasp functionality representation that explicitly defines the touch relations between robot hand segments and functional areas on the object point cloud. SayFuncGrasp distinguishes itself from previous DFG methods by fully exploring the potential of language guidance in the DFG learning process, which enhances the versatility and adaptability of the robot system in interactive operating environments.

B. Hand Synergies Based Robot Manipulation

The concept of hand synergies reveals that the human hand's complex motions can be effectively decomposed into a smaller set of coordinated movements [17], known as synergies. Drawn to this synergistic pattern, researchers have explored a variety of synergy-based methods for robot manipulation. Ciocarlie et al. [18] first proposed to reduce the dimensionality of the search space for dexterous grasp planning with hand synergies. The reduced computational burden allowed by a low-dimensional hand posture space enabled online grasp synthesis at a rate compatible with effective user interaction [19], and the reduction of the control variables [20]. Starke et al. [21] employed a deep autoencoder to capture the synergy characteristics involved in the human grasping process, subsequently applying these characteristics to predict the grasp actions of anthropomorphic robot hands. Furthermore, He et al. [22] discovered the advantage of synergy-based reinforcement learning (RL) policy in solving dexterous manipulation tasks. They demonstrated that refining the action space through hand synergies not only improved the convergence of the RL learning process but also enhanced the robustness of manipulation. In this work, we follow the same spirit and propose a synergy-based functional grasping policy to generate feasible DFG actions.

III. Problem Formulation

This work studies the problem of synthesizing functional grasps for dexterous robot hands based on natural language instructions. Mathematically, the proposed SayFuncGrasp framework $\mathcal{M}$ takes a language instruction L and a visual scene observation Ω as input and generates dexterous functional grasp G feasible for task execution:

$$G = \mathcal{M}(L, \Omega) \tag{1}$$

Here, L specifies an object o and a manipulation task τ. $\Omega = [\omega_r, \omega_l] \in \mathbb{R}^{H \times W \times 4 \times 2}$ includes two RGB-D images with the size of $H \times W \times 4$, which are acquired from the right and left sides using the fixed camera system illustrated in figure 2. A dexterous functional grasp $g \in G$ is characterized by the grasp functionality g_f and the grasp configuration g_c:

$$g = [g_f, g_c] \tag{2}$$

where grasp functionality g_f consists of grasp affordance g_f^a and grasp type g_f^t. g_f^a denotes a specific part of the target object that humans tend to grasp when performing a certain manipulation task. g_f^t describes the typical hand grasp shape within the human grasp taxonomy [23]. Grasp configuration g_c is represented by the hand preshape G_c (i.e., $g_c \in G_c$), which is defined as:

$$G_c = \{[p, j], p \in SE(3), j \in \mathbb{R}^n\} \tag{3}$$

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

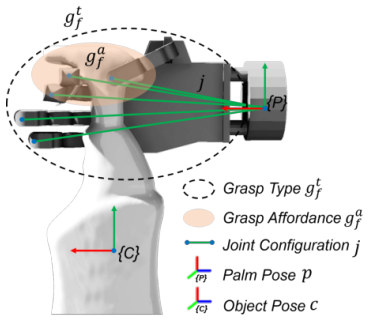

Fig. 3. Illustration of a dexterous functional grasp, showing the grasp type g_f^t, grasp affordance g_f^a, joint configuration j and palm pose p.

where p denotes the hand palm pose consisting of the displacement vector $d = [d_x, d_y, d_z]$ and the quaternion orientation vector $r = [r_w, r_x, r_y, r_z]$. j represents the joint configuration and n denotes the actual DoF of the hand. For the PISA/IIT SoftHand2 used in our work, $j \in \mathbb{R}^{19}$. The illustration of a dexterous functional grasp g is shown in figure 3.

IV. Method

A. Overview

SayFuncGrasp comprises two stages: few-shot human grasp functionality prompting and language-guided functional grasp synthesis. In the prompting stage, a grasp functionality prompt is generated based on human grasping knowledge, followed by in-context learning with an LLM to infer human-level grasp functionality. In the grasp synthesis stage, SayFuncGrasp first exploits the LLM to infer grasp functionality by determining grasp type and affordance from a given language instruction. A visual affordance grounding module is then employed to ground the text-based grasp affordance to mask-based visual affordance. Meanwhile, the Truncated Signed Distance Functional (TSDF) algorithm [24] is utilized to extract a dense volumetric representation of the object. Subsequently, a task-oriented grasp pose generation module takes both the visual affordance and the object TSDF as input and predicts functional palm poses. Finally, a synergy-based functional grasping policy synthesizes fine DFG actions using the predicted palm pose and the inferred grasp type.

B. Grasp Functionality Inference with LLM

Inferring appropriate grasp functionality g_f required for task execution is significantly challenging due to the complex interconnections among object categories, task designations, and functional grasps within language instructions. To achieve this, we propose an LLM-based grasp functionality inference method. The key idea behind this method is to leverage the open-ended manipulation knowledge in LLMs, combined with in-context human functional grasping examples and Chain-of-Thought (CoT) [25] reasoning, to infer grasp affordance g_f^a and grasp type g_f^t for object o under task τ.

A few-shot grasp functionality prompt is first generated including *Environment Information*, *Grasp Functionality Definition*, and *In-context Human DFG Examples*, as shown in figure 4. To define the grasp functionality, we investigate the typical grasp type used in human grasping and select 12 common ones for functional use (see figure 2):

Few-shot Grasp Functionality Prompt

Environment Information

***You are** a humanoid robot with anthropomorphic dexterous hands. **Your role** is to infer grasp functionality based on human instructions. Keep in mind that each answer contains only one set of the most plausible grasp type and grasp affordance.*

Grasp Functionality Definition

***The definition of grasp affordance that you can refer to**: The specific region of an object that humans tend to grasp when performing certain manipulation tasks.*
The grasp type of the dexterous hand that you can choose from include:
***Large Diameter:** The hand wraps around a large cylindrical object, with fingers and thumb encircling it;*
***Ring:** The fingers, excluding the thumb, loop through a ring-like object while the thumb supports from the outside;*
***Medium Wrap:** The hand firmly wraps around a medium-sized object, similar to the large diameter but for smaller objects;* ....

In-Context Human DFG Examples

Here are some examples you can learn from:
***User instruction**: "Grasp the knife to cut"*
***Answer:** Let's think step by step. The manipulation task the user wants me to perform is cutting, and the object they want me to grasp is a knife. Since humans most commonly use the "Knife Handle" when cutting, I should choose the grasp affordance of "Knife Handle." Additionally, as the knife handle is a medium-sized part and requires a firm grasp for cutting tasks, I should select the grasp type of "Medium Wrap" for the dexterous hand.*
Overall, the inference result is as follows:
Grasp Affordance: Knife Handle
Grasp Type: Medium Wrap
....

Fig. 4. The snippet of the few-shot grasp functionality prompt. Some of the grasp type definitions and in-context examples are omitted here due to content limitations. The CoT reasoning process is highlighted in red.

$$g_f^t = \{g_f^{t_1}, g_f^{t_2}, \ldots, g_f^{t_{12}}\} \subseteq \Phi \tag{4}$$

where $\Phi = \{g_1, g_2, g_3, \ldots, g_{33}\}$ is the Cutkosky's grasp taxonomy [23]. The definition of each functional grasp type and grasp affordance are then listed in *Grasp Functionality Definition* part. We then compose in-context human DFG examples by combining task instructions with the corresponding grasping functionalities that humans commonly use. To further improve the accuracy of the LLM inference process, we incorporate the intermediate reasoning steps of grasp functionality into the *In-context Human DFG Examples* through CoT. With the generated prompt, we harness the powerful In-Context Learning (ICL) ability [26] of LLMs to understand human functional grasping behaviors and infer appropriate grasp functionality when presenting novel instructions.

C. Visual Affordance Grounding

To ground the inferred text-based grasp affordance into mask-based visual affordance for guiding subsequent grasp pose generation, a visual affordance grounding module is proposed. Considering the extensive object categories that may be contained in natural language instructions, the inferred grasp affordance is open-ended. Traditional visual affordance detection methods [27], [28], which are trained on closed-world sets of semantic concepts, are inefficient for this task. Inspired by the impressive semantic understanding capability of visual foundation models (VFMs), we employ the open-vocabulary part detector (VLpart) [29] and the Segment Anything Model (SAM) [30] for visual affordance grounding. The

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

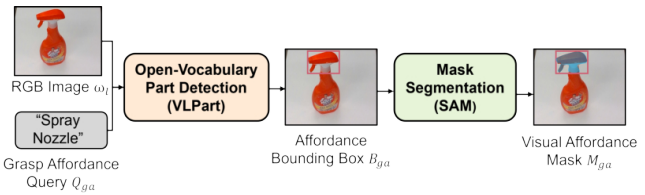

Fig. 5. The pipeline of visual affordance grounding module.

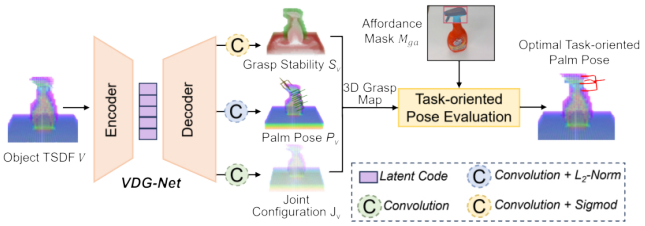

Fig. 6. The pipeline of task-oriented grasp pose generation based on VDG-Net.

overall pipeline is shown in figure 5. Given a grasp affordance query Q_{ga} and the RGB image of visual observation ω_l, the VLpart is first utilized to predict the optimal affordance bounding box B_{ga} for the target object. Subsequently, the bounding box B_{ga} is fed into Segment Anything as a prompt to generate the corresponding visual affordance mask $M_{ga} = [m_{ij}]$:

$$m_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \omega_l^f \\ 0 & \text{if } (i,j) \notin \omega_l^f \end{cases} \tag{5}$$

where (i, j) is the pixel index, ω_l^f is the object functional region within the image ω_l.

Nevertheless, directly applying pre-trained VFMs to downstream tasks in a zero-shot paradigm often causes incompleteness or incorrectness [16]. To overcome these problems, we introduce intermediary processing modules between the VFMs and the downstream tasks. To reduce the impact of shadows, we utilize a gray threshold filter and then perform a morphological closing operation to fill in small gaps before the image is processed by the VLpart. Following the segmentation process of SAM, a morphological opening technique is applied to remove small or disconnected spaces. Additionally, we discard masks with implausible sizes and utilize Non-Maximum Suppression (NMS) to minimize the production of superfluous masks.

D. Task-oriented Grasp Pose Generation

The task-oriented grasp pose generation module consists of two sequential steps: stable grasp pose prediction and task-oriented grasp pose evaluation. The overall network pipeline is illustrated in figure 6.

A Volumetric Dexterous Grasping Network (VDG-Net) is first proposed to predict stable grasp poses of dexterous hands. VDG-Net adopts a multi-branch 3D Fully Convolutional Network (FCN) featuring an asymmetric architecture for its encoder and decoder components. VDG-Net aims to approximate the mapping function f defined in Eq. (6). Here, V represents the input object TSDF volume with a size of 1×40^3. S_V denotes the predicted grasping stability volume with a size of 1×40^3. Each voxel within this volume contains a value in the range of (0,1), which signifies the stability of a potential grasp centered at that voxel. P_V is the predicted palm pose volume with a size of 4×40^3 which provides quaternion orientation r' for each grasp candidate. J_V denotes the joint configuration volume, where each voxel is associated with the predicted angles for the hand's joints.

$$f : V \rightarrow S_V, P_V, J_V \tag{6}$$

Given that the palm orientation r' is predicted within the TSDF coordinate frame, the palm pose p to the robot base coordinate frame is determined as follows:

$$\begin{cases} d = {}^B_V T(vl) \\ r = {}^B_V T(r') \end{cases} \tag{7}$$

where v is the voxel index corresponding to the location where the potential grasp is specified and l denotes the voxel size. The transformation matrix is represented as ${}^B_V T(v)$, which contains the transformation from the TSDF coordinate to the robot base coordinate. During training, VDG-Net is updated in an end-to-end manner with the following loss functions:

1) $L_s(s'_i, s_i)$ represents the binary cross-entropy loss, quantifying the divergence between the predicted stability labels s' and the ground-truth labels s.

2) $L_r(r'_i, r_i)$ is the inner product error for measuring the distance between the predicted quaternion orientations r' and target quaternion orientations r:

$$L_r(r', r) = 1 - |r' - r| \tag{8}$$

3) $L_j(j'_i, j_i)$ calculates the mean-squared errors to evaluate the differences between the predicted joint angles j' and the desired target joint angles j.

4) $L_{collision}(p)$ serves as the collision loss function, which imposes penalties for any penetrations between the palm and the object, ensuring the collision-free palm pose p is generated:

$$L_{collision}(p) = \sum_{i=1}^{K} \min^2(o_t(T(k_i, p)), 0) \tag{9}$$

where o_t is the truncated distance function of the object o, $k_i = 1, ..., K$ are a series of uniformly selected contact points on the hand palm. T is the forward kinematics associated with the dexterous hand, which transforms the contact point k_i into its global coordinate representation. Finally, the overall loss for training is formulated as follows:

$$L = L_s(s'_i, s_i) + s_i(L_r(r'_i, r_i) + L_j(j'_i, j_i) + L_{collision}(p_i)) \tag{10}$$

We selected 285 object meshes from KIT [31], YCB [32], and BigBIRD [33] and generated one million dexterous grasps and 25,000 object TSDFs in simulation to train our network. The training dataset is automatically generated in a self-supervised manner, consisting of three steps: palm pose sampling, grasp evaluation, and grasp storage. To address the simulation-to-reality gap, we also incorporated a variety of

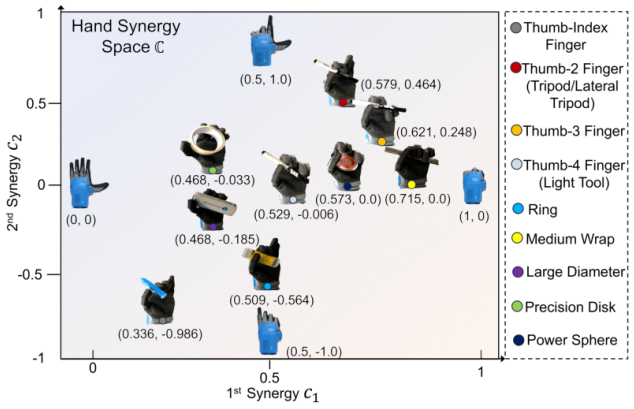

Fig. 7. Functional grasp type representation in the hand synergy space. The first synergy c_1 captures the dominant hand movement pattern during grasping, while the secondary synergy c_2 captures additional variations that refine the grasp.

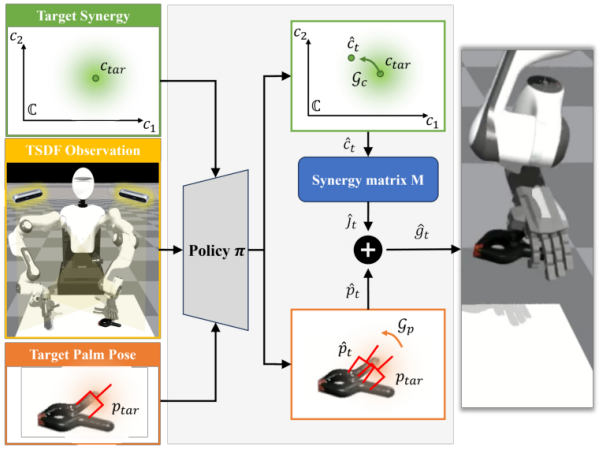

Fig. 8. The pipeline of synergy-based dexterous functional grasping policy.

physical parameter randomizations during the synthetic data generation process. Details of the data collection process can be found in our previous work [34].

To further generate the optimal task-oriented palm pose from the predicted stable candidates, we introduce a grasp pose evaluation mechanism. It utilizes the center of the visual affordance mask M_{ga} as the task key point and then projects the point into the aligned depth image to obtain its corresponding z value. By using the 3D coordinates of the task key point, the Euclidean distances between the key point and all the stable candidates are evaluated. Finally, the candidate with the minimum distance between the key point is selected as the optimal task-oriented palm pose.

E. Synergy-based Functional Grasp Synthesis

Once the dexterous hand is positioned in a task-oriented palm pose, the robot needs to synthesize fine-grained functional grasp actions for the object based on the inferred grasp type. Previous DFG methods typically learn grasp actions in full-dimensional joint space by mapping specific grasp types to finger joint angles [11]. While this approach facilitates an intuitive understanding of human grasp topology, it limits learning efficiency since humans grasp synergistically rather than controlling each finger joint individually. Drawing inspiration from the concept of *hand synergy* [17], we introduce a synergy-based functional grasping policy to efficiently learn grasp actions in low-dimensional hand synergy space.

1) Hand Synergy Space Construction: Hand synergy is defined as an ordered basis of the finger joint configuration space, identified by analyzing the covariance matrix obtained from experimental data on typical human hand movements through Principal Component Analysis [35]. To construct the low-dimensional hand synergy space $\mathbb{C} = span\{\vec{c_1}, \ldots, \vec{c_m}\} \subseteq \mathbb{R}^m$ from the finger joint configuration space $\mathbb{J} = span\{\vec{j_1}, \ldots, \vec{j_n}\} \subseteq \mathbb{R}^n$, the following transformation $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^m$ is introduced:

$$\mathbb{C} = \mathcal{T}(\mathbb{J}) = \mathbf{M}^{\dagger}\mathbb{J} \tag{11}$$

where $n \times m$ matrix $\mathbf{M}$ represents the synergy matrix formed by the eigenvectors of the covariance matrix.

According to [19], more than 80% of the variance information pertinent to hand grasping is encompassed by the first two principal components, indicating that a majority of habitual grasp configurations can be reconstructed from merely a pair of these principal component vectors. Therefore, we form the synergy matrix $\mathbf{M}$ for SoftHand2 with the first two principal components, i.e., $m = 2$. On this basis, the grasp posture defined by a group of finger joint angles $\boldsymbol{j}$ could be represented completely in hand synergy space with two synergy coordinates $[c_1, c_2]$:

$$\begin{bmatrix} j_1 & \ldots & j_{15} \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} c_1 & c_2 \end{bmatrix} \tag{12}$$

where $\mathbf{M}$ was elaborated in [35] as

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -2 & -2 & -2 \end{bmatrix}^{\top} \tag{13}$$

In this work, the range of synergy coordinates c_1 and c_2 are set to $[0, 1]$ and $[-1, 1]$ according to [35]. To encourage the robot hand to explore the hand synergy space associated with the functional grasp type instead of the entire joint configuration space, we then represent the hand posture of all 12 functional grasp types with synergy coordinates. The constructed synergy space and the represented functional grasp types are illustrated in figure 7.

2) Synergy-based Grasp Policy Learning: Since the grasp size may vary with the object's shape and size, a synergy-based reinforcement learning policy is designed to synthesize adaptive functional grasp actions for different objects. The core idea is that the policy exploits low-dimensional hand synergy space $\mathbb{C}$ to generate grasp actions for synergies instead of joints. As the number of synergies is much fewer than joints, the learning complexity is reduced. Given the inferred grasp type from LLMs, we first map it to the corresponding target synergy $\boldsymbol{c}_{tar} = [c_1, c_2]$ predefined in the figure 7. Subsequently, the proposed synergy-based grasping policy refines the target synergies to adapt to the specific geometry of the object. This refinement achieves a stable and functional grasp that respects the inferred grasp type and conforms

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

to the object's shape. Specifically, we use Proximal Policy Optimization (PPO) [36] to implement the policy. The pipeline of synergy-based grasping policy is shown in figure 8.

Observation: The observation of the policy contains target synergy $\boldsymbol{c}_{tar} = [c_1, c_2]$, object TSDF observation o_t and task-oriented palm pose p_{tar}.

Action: The policy outputs a sequence of synergy-based functional grasp actions, which consist of synergy coordinates $\hat{\boldsymbol{c}}_t$ and palm pose $\hat{p}_t$. These synergy-based actions are then combined linearly via the synergy matrix $\mathbf{M}$ to produce final joint space grasp actions $\hat{\mathbf{g}} = (\hat{g_1}, \ldots, \hat{g_T})$:

$$\hat{g}_t \leftarrow \pi\left(o_t | \boldsymbol{c}_{tar}, p_{tar}\right), t \in T \tag{14}$$

Reward: The policy is tasked with exploring suitable synergy $\hat{\boldsymbol{c}}_t$ and palm pose $\hat{p}_t$ values around the target coordinates $\boldsymbol{c}_{tar}, p_{tar}$ to form the grasp action $\hat{g}_t$. To facilitate this exploration, two normal distributions are defined: $\mathcal{G}_c(\boldsymbol{c}_{tar}, \sigma_c), \mathcal{G}_p(p_{tar}, \sigma_p)$, where the variances σ_c, σ_p determine the extent of exploration around the target values:

$$r_t = \alpha_{lift} r_{lift} + \alpha_c \mathcal{G}_c(\hat{\boldsymbol{c}}_t) + \alpha_p \mathcal{G}_p(\hat{p}_t) \tag{15}$$

where $\alpha_{lift}, \alpha_c, \alpha_p$ are weight parameters, r_{lift} is the height at which objects are lifted.

V. Experiments

A. Setup

1) Implementation Details: The GPT-4 model, specifically the gpt-4-0314 version, is chosen as the LLM with a temperature parameter set to 0 and a max token set to 512. For visual affordance grounding, we select the pre-trained SwinTransformer-Base model and VisionTransformer-Base model provided by Hugging Face as the backbone of VLpart and SAM respectively. Furthermore, the entire VDG-Net network is implemented in PyTorch and trained with the Adam optimization algorithm, utilizing batch sizes of 64 across 60 training epochs, while maintaining a constant learning rate of 3×10^{-4}. Open3D [37] is used to reconstruct object TSDF from depth images. The simulation environments are constructed based on Isaac Gym [38], a GPU-based parallel physics simulator designed for robot learning. Both simulation and real robot experiments are conducted on a graphics workstation with a single Nvidia RTX 4090 GPU. Additionally, typed instructions are used to convey grasping commands to the robot during the experiments. SayFuncGrasp can also accept voice instructions, simply by adding an additional speech-to-text module such as OpenAI Whisper [39].

2) Evaluation Metrics: The following quantitative metrics are used to evaluate our approach:

- **ε-quality** [40] assesses the quality of the generated grasps by calculating the maximum ball radius that can fit within the convex hull of the wrench space origin.
- **Penetration Depth and Volume** evaluate the collision of the generated grasps by quantifying the depth and volume of penetration between the robot hand and the object. We use the same metric settings as in [9] and voxelise the hand and objects with a grid size of 0.25cm.
- **Grasp Functionality Accuracy (GFA)** measures the accuracy of the inferred and executed grasp functionality, which is the number of correct grasp functionality as a proportion of the total number of predictions.
- **Grasp Success Rate (GSR)** quantifies the overall grasping performance of the proposed method. A successful dexterous functional grasp is defined as one in which the hand can accurately grasp the object following the specified task requirements.
- **Grasp Generation Time** assesses the time efficiency of the proposed method for grasp generation.
- **Manipulation Success Rate (MSR)** measures the practicality of the proposed method in achieving the post-grasp manipulation, which corresponds to the percentage of successful manipulations executed by the dexterous hand.

TABLE I
Accuracy of Grasp Functionality Inference on Open-set Human Instructions.

	Grasp Affordance	Grasp Type
Ablated Prompt (w/o *GFD*)	0.21	0.17
Ablated Prompt (w/o *IE*)	0.55	0.45
Ablated Prompt (w/o *COT*)	0.75	0.64
Full Prompt	**0.92**	**0.85**

TABLE II
Results of VDG-Net Simulation Experiments. ↑: Higher the Better, ↓: Lower the Better

		VDG-Net	GraspIt!	PointNetGPD	VGN
Grasp Success Rate (%) ↑		**0.88**	0.47	0.51	0.65
Grasp Generation Time (sec.) ↓		**0.23**	3.74	1.34	0.25
Avg. ε-quality ↑		0.57	**0.62**	0.44	0.50
Penetration	Depth (cm) ↓	**0.81**	1.87	3.32	1.23
	Volume (cm^3) ↓	**1.57**	1.72	6.06	2.50

B. Experiments for Grasp Functionality Inference

We conduct an evaluation experiment to validate the LLM-based grasp functionality inference method. Initially, we construct an open-set human instruction dataset based on the Language Instruction Template [14], which includes 100 randomly composed language instructions across 57 distinct manipulation tasks and 68 different object categories. Human annotators are then engaged to label the ground truth grasp functionality for each task instruction. To verify the correctness of the inferred grasp functionality, we compare the results from our LLM-based module against the human-labeled ground truth. Specifically, we employ the GFA metric to evaluate the inference performance. Furthermore, to examine the efficacy of each component in the designed grasp functionality prompt, three types of prompt are ablated as shown in Table I. The *GFD*, *IE*, and *COT* in the table denote the *Grasp Functionality Definition* component, the *In-context Examples* component, and the *Chain of Thought* component, respectively.

Experiment results demonstrate the effectiveness of the proposed method in inferring human-level grasp functionality, achieving a high GFA on the test dataset (i.e., 0.92 for Grasp Affordance and 0.85 for Grasp Type). Furthermore, the GFA of

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

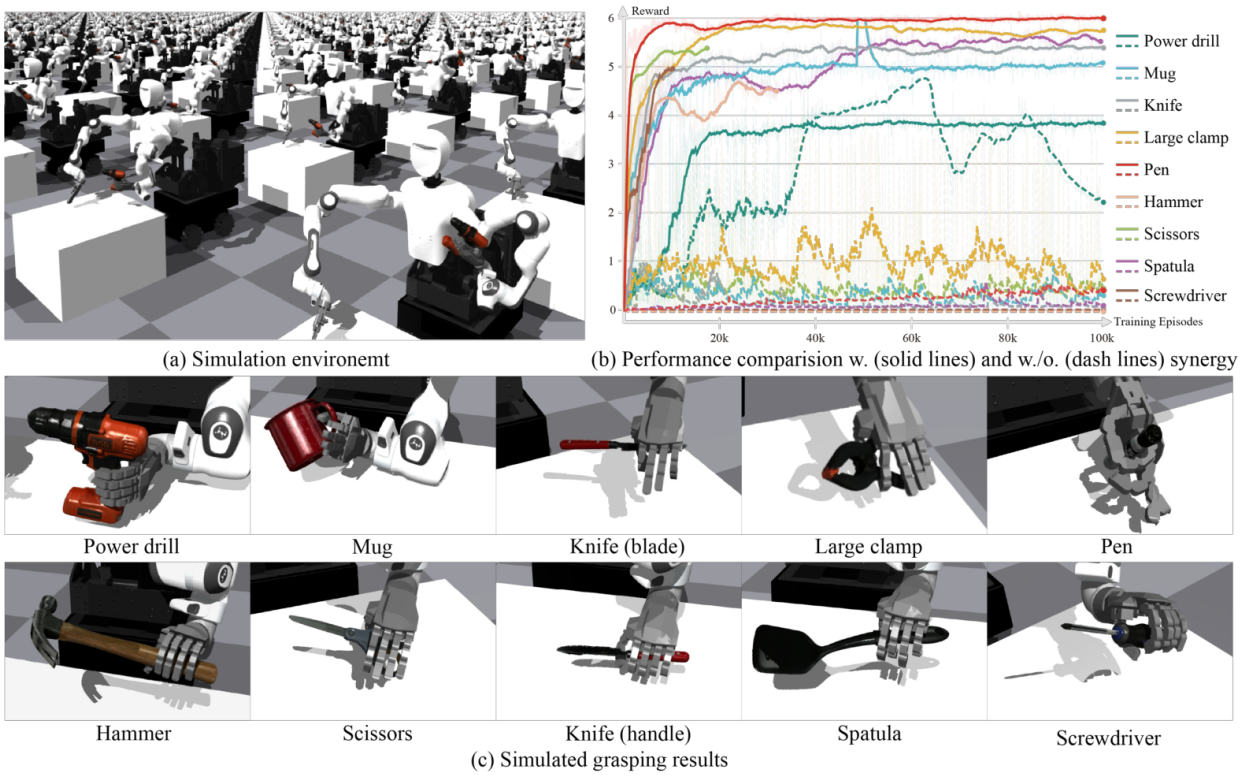

Fig. 9. Results of synergy-based functional grasping policy in simulation.

the full prompt outperforms all three ablations, which validates the efficacy of each component in the designed prompt. The high inference accuracy confirms the capability of our method to understand and learn human functional grasping behaviors, facilitated by the synergistic integration of CoT and ICL within the few-shot grasp functionality prompt. While the results are promising, we find that LLMs can still generate false or inconsistent grasp functionality. Incorporating an evaluation module, similar to that used in [41], to systematically assess and refine the accuracy of the LLM-generated outputs could be a promising direction for further enhancing the reliability of our method.

C. Experiments for VDG-Net

To assess the stable grasp detection capability of the proposed VDG-Net, we conducted 30 rounds of simulated grasping experiments using a SoftHand2 manipulator. For the evaluation, 10 seen objects from the YCB dataset and 10 novel objects are selected. Three state-of-the-art grasp detection methods are chosen as baselines for comparison: GraspIt! [42], PointNetGPD [43], and Volumetric Grasping Network (VGN) [44]. The experimental procedure in each round is as follows: 1) 15 grasp candidates are generated per object using each algorithm; 2) The most stable grasp candidate is selected and executed. For GraspIt!, the ε-metric is adopted to select grasps, while for PointNetGPD, VGN and VDG-Net, the grasp candidate with the highest stability score is selected.

Table II shows the quantitative experimental results of each method. The VDG-Net significantly outperforms the three baseline methods in terms of grasp success rate, indicating a greater consistency in generating high-quality and stable grasp poses. This can be attributed to the benefits of utilizing a large-scale grasp dataset for network training, which improves the generalizability and grasp detection accuracy of VDG-Net. Additionally, the average grasp generation time for VDG-Net is 0.23 seconds, a speed 16 times greater than that of GraspIt!. This efficiency stems from the end-to-end nature of VDG-Net, which prevents the requirement of sampling and evaluating grasp candidates during run-time, thereby diminishing the time required for grasp generation. Furthermore, VDG-Net achieves minimal penetration during grasping compared to the baseline methods, underscoring the effectiveness of the $L_{collision}$ loss functions used in network training, and demonstrating the capability of VDG-Net to predict collision-free grasp poses.

D. Experiments for Synergy-based Functional Grasping

The experimental grasping tasks include nine objects: power drill, hammer, mug, scissors, knife, large clamp, spatula, pen, and screwdriver (see figure 9(c)). All object assets are standard models provided by YCB and generated as URDF files by Rofunc [45]. The model of Collaborative dUal-arm Robot manIpulator (CURI) is introduced in the simulation, and a table with suitable height is set in front of CURI, and objects are placed on it, as shown in figure 9(a). The initial poses of objects are generated uniformly around the table center with a limited range. With the support of GPU-based parallel training ability provided by Isaac Gym, the simulation experiments are conducted with 1024 environments simultaneously. In each environment, the observation of the RL agent is object TSDF provided by two cameras located above the robot's shoulder.

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

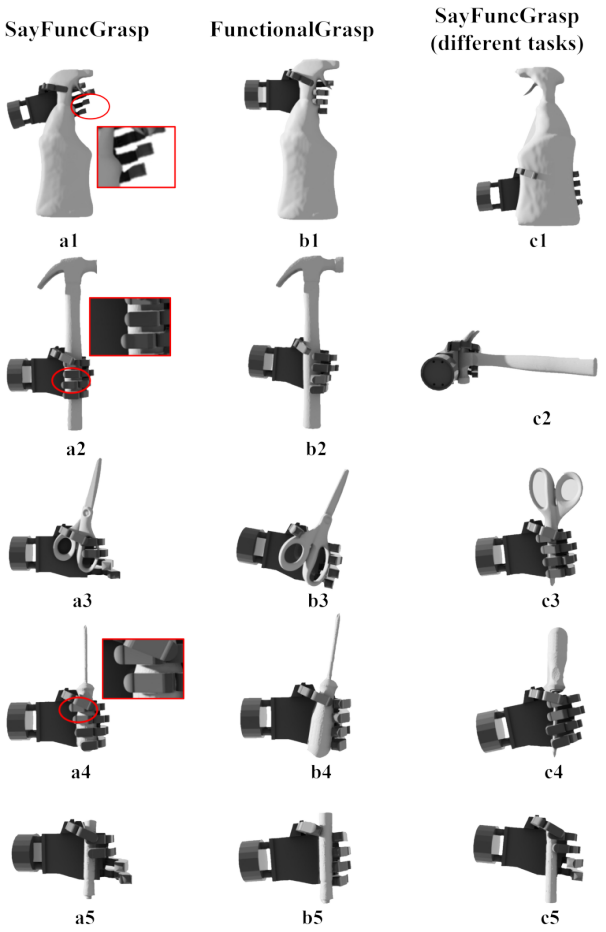

Fig. 10. Qualitative results of comparison experiments. a1-a5 and b1-b5: comparisons between grasps synthesized by SayFuncGrasp and [11], the red circles indicate the finger links with no tight contacts. c1-c5: demonstration of the versatility exhibited by SayFuncGrasp for different task requirements.

As shown in figure 9(c), the RL agent achieved the functional grasping of all objects. We also tried different grasp types on the same object. The agent is required to grasp the blade and handle of a knife after giving target grasp pose and target synergy, and the result in the middle of figure 9(c) shows that the RL agent has the generalization among different synergies. It is worth noting that one of the key features that need to be verified by simulation experiments is the effect of using synergy for dexterous functional grasping. Figure 9(b) compares learning performance between using synergy and controlling each finger joint directly with all nine selected objects. Directly controlling high-dimensional finger joints indeed reflects on the difficulty in grasping learning. Most objects in the simulation experiments cannot be grasped without synergy. This demonstrates the effectiveness and importance of using synergy-based control in dexterous functional grasping tasks.

E. Comparison Experiments

To the best of our knowledge, SayFuncGrasp is the first study on synthesizing dexterous functional grasps from natural language instructions. Therefore, it is difficult to compare the language-guided grasping performance with existing DFG methods. Here we only compare functional grasping performance in simulation with FunctionalGrasp [11], which utilizes the semantic touch code to guide the synthesis of functional grasps and achieves state-of-the-art performance. Given that [11] requires object point clouds as visual input, we first extract the point cloud of each test object from the TSDF. Furthermore, we replace the gripper layer of the mapping module in [11] for Softhand2, and retrain the grasp refinement module using the grasp dataset collected in Sec. IV-D. We conduct comparative experiments focusing on two aspects: closed-set grasp functionality evaluation and open-set grasp functionality generalization. The former evaluates the grasp performance of SayFuncGrasp on pre-defined functional concepts (i.e., known object categories and manipulation tasks within the semantic touch codes). In contrast, the latter assesses the generalization performance of both methods on novel functional concepts outside the semantic touch codes.

TABLE III
COMPARISON EXPERIMENT RESULTS UNDER DIFFERENT EVALUATION SETTINGS. *NO* DENOTES NOVEL OBJECT CATEGORIES, *NT* DENOTES NOVEL MANIPULATION TASKS.

Evaluation Setting	FunctionalGrasp		Ours	
	GSR	GFA	GSR	GFA
Close-set Evaluation	**73.14%**	**0.89**	72.44%	0.83
Open-set Generalization (NO)	58.77%	0.63	**69.52%**	**0.79**
Open-set Generalization (NO&NT)	40.23%	0.44	**68.86%**	**0.77**

Table III presents the overall grasp success rate. In the closed-set evaluation, SayFuncGrasp attains GSR and GFA that are on par with [11], suggesting its effectiveness in generating viable functional grasps. Nonetheless, a notable discrepancy in performance emerges during the open-set generalization test. The performance of [11] experiences a significant decline when applied to unfamiliar tasks, indicating that while pre-defined grasp functionalities may effectively encapsulate the interplay between functional grasps and specific object categories within known tasks, they falter when applied to novel functional concepts. In contrast, SayFuncGrasp consistently demonstrates robust performance, achieving an average grasp success rate of 68.86% and a grasp functionality accuracy of 0.77. This enhanced adaptability can be ascribed to its independence from pre-defined functional concepts. Instead, it capitalizes on the expansive manipulation knowledge sourced from LLMs to deduce grasp functionalities for new object categories and tasks, thereby displaying a more generalized grasping capability. Moreover, the qualitative results shown in figure 10 demonstrate the versatility of SayFuncGrasp, which can synthesize plausible grasps for the designated object tailored to different task requirements, such as *handover the detergent* (as seen in c1) or *take off the pen cap* (as seen in c5). While our method demonstrates enhanced capability in producing generalizable and versatile grasps, it retains a limitation compared to the baseline method. Specifically, as depicted in subplots a1-a5, SayFuncGrasp occasionally fails to ensure tight contact between the finger links and objects, potentially causing slippage. This issue arises because SayFuncGrasp focuses mainly on identifying grasp affordance

Fig. 11. Real robot experiments for language-guided dexterous functional grasping

and type, rather than specifying detailed contact maps used as described in [11].

F. Real-Robot Experiments

We built a physical robot platform (see figure 12) to assess the real-world performance of the SayFuncGrasp framework. The platform comprises a humanoid CURI robot with two Franka Panda arms and SoftHand2 anthropomorphic hands. Two RealSense D435i cameras are fixedly positioned above the platform to record both RGB and depth images. Since we only consider functional grasping with a single hand, the left-hand-arm system of the CURI robot is utilized for the experiment.

1) Language-guided Dexterous Functional Grasping: The robot is required to grasp the target object functionally based on human language instructions. To evaluate the performance of SayFuncGrasp, we compare it against five baseline methods as follows:

- FunctionalGrasp [11]: To enable FunctionalGrasp to process language instructions, we manually converted the instructions into predefined grasp types and touch codes. The generated full-dimensional grasp configurations were then mapped to hand synergy parameters for execution using Equation 12.

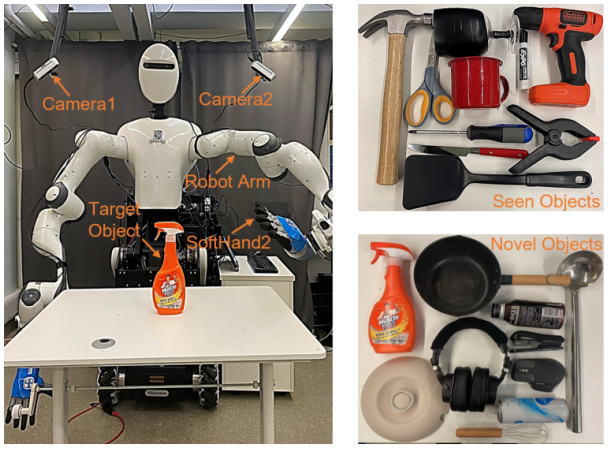

Fig. 12. Real robot experiment settings. (a) CURI humanoid robot grasping platform. (b) Testing objects: the top part shows 10 seen YCB objects, and the bottom part shows 10 novel objects collected in our lab.

- GraspGPT [14]: This foundation model-based grasping approach leverages the semantic knowledge of LLMs to generate task-oriented grasp poses, specifically designed for parallel grippers.
- LAN-Grasp [46]: A novel language-guided grasping framework that utilizes LLMs to infer generalized object

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

TABLE IV
RESULTS OF LANGUAGE-GUIDED DEXTEROUS FUNCTIONAL GRASPING EXPERIMENTS ON SEEN OBJECTS

Method	Seen Objects										Average GSR
	Mug	Power Drill	Screwdriver	Pen	Wine Glass	Scissors	Hammer	Knife	Large Clamp	Spatula	
FunctionalGrasp	8/15	11/15	9/15	5/15	8/15	8/15	12/15	9/15	12/15	13/15	63.33%
GraspGPT	7/15	10/15	8/15	2/15	2/15	4/15	11/15	10/15	11/15	12/15	51.33%
LAN-Grasp	5/15	8/15	4/15	1/15	1/15	3/15	9/15	9/15	10/15	11/15	40.66%
QwenGrasp	4/15	7/15	2/15	0/15	0/15	2/15	8/15	8/15	8/15	9/15	32.00%
ThinkGrasp	6/15	10/15	7/15	2/15	2/15	3/15	11/15	9/15	12/15	11/15	48.66%
Ours	10/15	12/15	10/15	4/15	7/15	7/15	13/15	11/15	13/15	13/15	**66.67**%

TABLE V
RESULTS OF LANGUAGE-GUIDED DEXTEROUS FUNCTIONAL GRASPING EXPERIMENTS ON NOVEL OBJECTS

Method	Novel Objects										Average GSR
	Spoon	Soda Can	Pan	Bottle	Spray Detergent	Headset	Mouse	Stapler	Bowl Lid	Whisk	
FunctionalGrasp	7/15	8/15	9/15	6/15	4/15	7/15	9/15	9/15	7/15	8/15	49.33%
GraspGPT	8/15	11/15	8/15	4/15	3/15	6/15	8/15	7/15	7/15	7/15	46.00%
LAN-Grasp	8/15	9/15	8/15	3/15	2/15	4/15	6/15	6/15	4/15	5/15	36.66%
QwenGrasp	6/15	7/15	6/15	2/15	0/15	4/15	6/15	6/15	0/15	4/15	27.33%
ThinkGrasp	8/15	10/15	9/15	3/15	3/15	5/15	7/15	7/15	6/15	6/15	42.66%
Ours	10/15	11/15	11/15	7/15	7/15	10/15	10/15	9/15	9/15	10/15	**62.66**%

semantics and achieve task-oriented grasping.

- QwenGrasp [47]: A combinational task-oriented grasping method that integrates a Multimodal Large Language Model (MLLM) with a 6-DoF grasp neural network.
- ThinkGrasp [48]: A plug-and-play vision-language grasping system that employs GPT-4o's advanced contextual reasoning to enable strategic part grasping.

We first randomly compose five task instructions for each test object. Then, each method takes the composed instructions and object observation as input and synthesizes grasp actions. To execute the grasp, the robot approaches the target palm pose using MoveIt! and closes the hand based on the synthesized hand synergy parameters. For grasp poses generated by the baseline methods, the closed grasp was mapped to the synergy coordinates (1, 0) to ensure compatibility with the SoftHand2. Each instruction is performed three times, and the final result for each test object is determined by calculating the average grasp success rate across all trials.

Experimental results presented in Tables IV and V demonstrate the effectiveness of our method. SayFuncGrasp achieves the highest average GSR of 66.67% on the seen YCB objects, outperforming all baseline methods. In comparison, GraspGPT (51.33%), LAN-Grasp (40.66%) and ThinkGrasp (48.66%) exhibit lower performance, particularly on objects requiring precise functional grasps (e.g., wine glasses and pens). This disparity arises because the grasp poses generated by these baseline methods do not incorporate specific grasp type information, leading to suboptimal hand configurations. As a result, failure rates increase when these poses are executed on a dexterous hand, underscoring the importance of accurate grasp type inference for functional grasping. For novel objects, SayFuncGrasp maintains its superiority with the highest average GSR of 62.66%. It outperforms FunctionalGrasp (49.33%) by dynamically adapting to novel object categories and tasks, where predefined grasp types are insufficient. This consistency highlights the value of integrating open-ended manipulation knowledge from LLMs to enhance the generalization capabilities of DFG. QwenGrasp, with an average GSR of 27.33%, struggles with thin and precise object parts (e.g., bowl lid and spray nozzel), exposing limitations in its multimodal integration for grasp affordance detection. During the experiments, SayFuncGrasp demonstrated reliable grasping capabilities when confronted with ambiguous or unclear language instructions. We ascribe this reliability to the integration of CoT-based reasoning in the LLM prompting and the visual context provided by VLMs. By merging the semantic information extracted from the LLMs with the visual affordance cues, the system was able to accurately infer the intended grasp functionality, even when the language instructions were vague or incomplete.

Regarding time efficiency, the average inference time for SayFuncGrasp is 8.23 seconds. The majority of this time, approximately 6-7 seconds, is due to the delayed response from the LLMs and VLMs. We consider this a reasonable compromise between the generalization performance and time efficiency that foundation models bring to robotic manipulation frameworks [49]. Qualitative results shown in figure 11 also indicate that SayFuncGrasp can effectively generate versatile DFG actions according to different task instructions. Moreover, figure 13 depicts the frequency distribution of various grasp types observed during the real robot experiments, which confirms the successful execution of all 12 functional grasping types as delineated in our research. The distribution of grasp type frequencies enacted by SayFuncGrasp bears a resemblance to the distribution characteristic of human experts, which suggests that the functional grasping capability of SayFuncGrasp is comparable to human proficiency.

2) Post-grasp Manipulation: To further evaluate the practi-

IEEE Transactions on Automation Science and Engineering (T-ASE) paper, presented at ICRA 2026, Vienna, Austria.

TABLE VI
RESULTS OF POST-GRASP MANIPULATION EXPERIMENTS

SayFuncGrasp	Stir-fry		Handover		Pouring		Hanging		Whisking		Covering		Average MSR
	GSR	MSR	GSR	MSR	GSR	MSR	GSR	MSR	GSR	MSR	GSR	MSR	
	15/20	15/20	17/20	16/20	13/20	12/20	14/20	14/20	14/20	14/20	13/20	12/20	70.41%

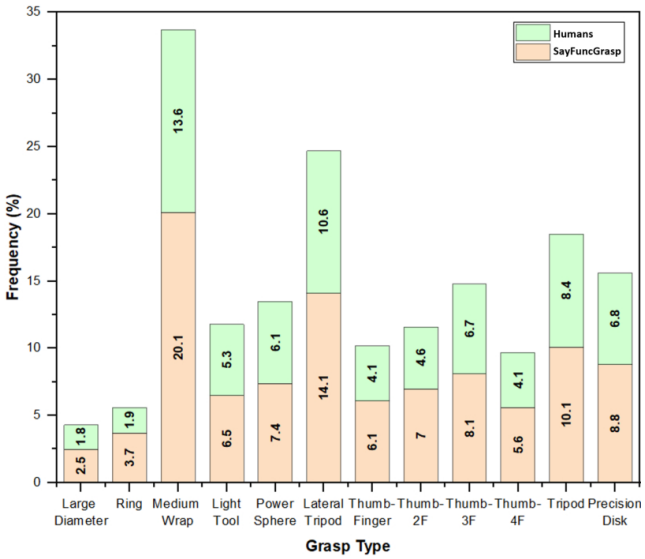

Fig. 13. Comparison of the frequency of 12 functional grasp types between SayFuncGrasp and human experts in executing grasps. The benchmark distribution for human grasps is sourced from [23].

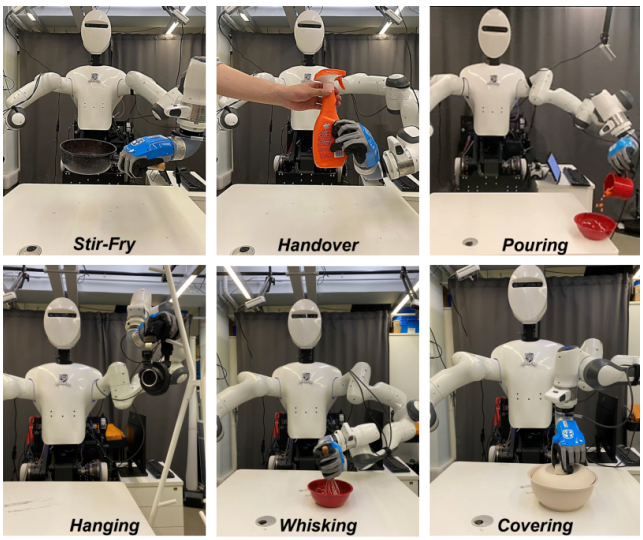

Fig. 14. Real-robot experiments on six post-grasp manipulation tasks.

cality of SayFuncGrasp in post-grasp manipulation, we conducted experiments on six representative tasks: stir-fry, handover, pouring, hanging, whisking, and covering. As shown in figure 14, these tasks cover diverse manipulation scenarios that require precise grasping and subsequent task-specific actions. Each of these tasks was supported by pre-defined skills, implemented as motion primitives combined with SayFuncGrasp's functional grasping capabilities. Experiment results shown in Table VI demonstrate that SayFuncGrasp performs well across all post-grasp manipulation tasks, achieving an average manipulation success rate of 70.41%.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented SayFuncGrasp, a novel language-guided DFG framework that can synthesize versatile functional grasps of high-DoF dexterous hands based on language instructions. SayFuncGrasp first leverages an LLM to infer grasp functionality from the given instruction. Then, it adopts a task-oriented grasp pose generation module and a synergy-based functional grasping policy to efficiently generate feasible DFG actions with hand synergy representations. Unlike existing DFG methods, SayFuncGrasp does not rely on any pre-defined grasp functionality set. Instead, it employs the open-end manipulation knowledge from an LLM to infer grasp functionality for unseen object classes and manipulation tasks, thus attaining generalization towards novel functional concepts. Experimental results demonstrate the effectiveness of SayFuncGrasp in the real-world, achieving a 64.66% grasp success rate and a 70.41% manipulation success rate for various humanoid tasks with open-set instructions. Overall, SayFuncGrasp possesses the potential to further advance the DFG field by introducing an interactive, generalized functional grasping system. This serves as a fundamental basis for non-expert users to efficiently and intuitively craft diverse dexterous manipulation behaviors of humanoid robots through verbal commands.

One of the limitations of SayFuncGrasp is that it does not explicitly account for the kinematic reachability of the functional grasps generated. This consideration is crucial for humanoid robots equipped with dual dexterous hands, as the robot may be unable to execute a desired grasp if it falls outside the arm's workspace or exceeds the joint limits. Future work will explore integrating reachability-aware grasp evaluation modules, such as the Grasp Reachability Evaluator [34], to ensure that the dexterous grasps generated are not only functionally appropriate but also kinematically feasible for the robot to execute. Another limitation is the computational overhead introduced by the use of LLMs for grasp functionality inference. We will investigate techniques like model compression, quantization, and distillation to optimize the LLM inference time, aiming to reduce it to the millisecond level and enable real-time humanoid manipulation.

REFERENCES

[1] C. L. Teo, Y. Yang, H. Daumé, C. Fermüller, and Y. Aloimonos, "Towards a watson that sees: Language-guided action recognition for robots," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 374–381.

[2] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[3] K. Namasivayam, H. Singh, V. Bindal, A. Tuli, V. Agrawal, R. Jain, P. Singla, and R. Paul, "Learning neuro-symbolic programs for language guided robot manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7973–7980.

[4] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Conference on Robot Learning*. PMLR, 2023, pp. 3766–3777.

[5] K. Zheng, X. Chen, O. C. Jenkins, and X. Wang, "Vlmbench: A compositional benchmark for vision-and-language manipulation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 665–678, 2022.

[6] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, "Contactdb: Analyzing and predicting grasp contact via thermal imaging," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8709–8719.

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

[8] S. Brahmbhatt, C. Tang, C. D. Twigg, C. C. Kemp, and J. Hays, "Contactpose: A dataset of grasps with object contact and hand pose," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, 2020, pp. 361–378.

[9] W. Wei, P. Wang, and S. Wang, "Generalized anthropomorphic functional grasping with minimal demonstrations," *arXiv preprint arXiv:2303.17808*, 2023.

[10] R. Wu, T. Zhu, W. Peng, J. Hang, and Y. Sun, "Functional grasp transfer across a category of objects from only one labeled instance," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2748–2755, 2023.

[11] Y. Zhang, J. Hang, T. Zhu, X. Lin, R. Wu, W. Peng, D. Tian, and Y. Sun, "Functionalgrasp: Learning functional grasp for robots via semantic hand-object representation," *IEEE Robotics and Automation Letters*, 2023.

[12] T. Zhu, R. Wu, J. Hang, X. Lin, and Y. Sun, "Toward human-like grasp: Functional grasp by dexterous robotic hand via object-hand semantic representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[13] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *Microsoft Auton. Syst. Robot. Res*, vol. 2, p. 20, 2023.

[14] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping," *IEEE Robotics and Automation Letters*, 2023.

[15] X. Zhao, M. Li, C. Weber, M. B. Hafez, and S. Wermter, "Chat with the environment: Interactive multimodal perception using large language models," *arXiv preprint arXiv:2303.08268*, 2023.

[16] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li, "Instruct2act: Mapping multi-modality instructions to robotic actions with large language model," *arXiv preprint arXiv:2305.11176*, 2023.

[17] M. Santello, M. Bianchi, M. Gabiccini, E. Ricciardi, G. Salvietti, D. Prattichizzo, M. Ernst, A. Moscatelli, H. Jörntell, A. M. Kappers *et al.*, "Hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands," *Physics of life reviews*, vol. 17, pp. 1–23, 2016.

[18] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3270–3275.

[19] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.

[20] F. Ficuciello, G. Palli, C. Melchiorri, and B. Siciliano, "Experimental evaluation of postural synergies during reach to grasp with the ub hand iv," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1775–1780.

[21] J. Starke, C. Eichmann, S. Ottenhaus, and T. Asfour, "Synergy-based, data-driven generation of object-specific grasps for anthropomorphic hands," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 327–333.

[22] Z. He and M. Ciocarlie, "Discovering synergies for robot manipulation with multi-task reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2714–2721.

[23] 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*, vol. 46, no. 1, pp. 66–77, 2015.

[24] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee, 2011, pp. 127–136.

[25] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[26] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[27] R. Xu, F.-J. Chu, C. Tang, W. Liu, and P. A. Vela, "An affordance keypoint detection network for robot manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2870–2877, 2021.

[28] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[29] P. Sun, S. Chen, C. Zhu, F. Xiao, P. Luo, S. Xie, and Z. Yan, "Going denser with open-vocabulary part segmentation," *arXiv preprint arXiv:2305.11173*, 2023.

[30] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[31] A. Kasper, Z. Xue, and R. Dillmann, "The kit object models database: An object model database for object recognition, localization and manipulation in service robotics," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 927–934, 2012.

[32] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.

[33] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 509–516.

[34] Z. Li, S. Li, K. Han, X. Li, Y. Xiong, and Z. Xie, "Planning multi-fingered grasps with reachability awareness in unrestricted workspace," *Journal of Intelligent & Robotic Systems*, vol. 107, no. 3, p. 39, 2023.

[35] C. Della Santina, C. Piazza, G. Grioli, M. G. Catalano, and A. Bicchi, "Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1141–1156, 2018.

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

[37] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

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

[39] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[40] A. T. Miller and P. K. Allen, "Examples of 3d grasp quality computations," in *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, vol. 2. IEEE, 1999, pp. 1240–1246.

[41] Y. Jin, D. Li, A. Yong, J. Shi, P. Hao, F. Sun, J. Zhang, and B. Fang, "Robotgpt: Robot manipulation learning from chatgpt," *IEEE Robotics and Automation Letters*, 2024.

[42] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[43] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635.

[44] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *Conference on Robot Learning*. PMLR, 2021, pp. 1602–1611.

[45] J. Liu, C. Li, D. Delehelle, Z. Li, and F. Chen, "Rofunc: The full process python package for robot learning from demonstration and robot manipulation," Jun. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8084510

[46] R. Mirjalili, M. Krawez, S. Silenzi, Y. Blei, and W. Burgard, "Langrasp: Using large language models for semantic object grasping," *arXiv preprint arXiv:2310.05239*, 2023.

[47] X. Chen, J. Yang, Z. He, H. Yang, Q. Zhao, and Y. Shi, "Qwengrasp: A usage of large vision language model for target-oriented grasping," *arXiv preprint arXiv:2309.16426*, 2023.

[48] Y. Qian, X. Zhu, O. Biza, S. Jiang, L. Zhao, H. Huang, Y. Qi, and R. Platt, "Thinkgrasp: A vision-language system for strategic part grasping in clutter," *arXiv preprint arXiv:2407.11298*, 2024.

[49] 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," *arXiv preprint arXiv:2307.05973*, 2023.

Zhuo Li received the M.S. degree in mechanical and automation engineering from Huazhong University of Science and Technology, Wuhan, China, in 2023. He is currently pursuing the Ph.D. degree at The Chinese University of Hong Kong. His current research interests are in humanoid robot grasping, dexterous manipulation, deep learning-based visual perception and 3D recognition, and large language models.

Junjia Liu received the M.S. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2021. He is currently pursuing the Ph.D. degree at The Chinese University of Hong Kong. His research interests are in humanoid robot manipulation, learning from demonstration, large models and reinforcement learning based decision making.

Zhihao Li was a research assistant at The Chinese University of Hong Kong (Shenzhen) Shenzhen, China, in 2021. He is currently pursuing a Ph.D. degree at The Chinese University of Hong Kong. His research interests include robot manipulation, robot learning, 3D perception, and multi-modal large models.

Zhipeng Dong received the B.S degree in mechanical design manufacture and automation from the Harbin Institute of Technology, Harbin, Heilongjiang, China, in 2013, and the M.S. degree in mechanical design and theory from Northeastern University, Shenyang, Liaoning, China, in 2015, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interest includes robotic vision, manipulation, and machine learning techniques.

Tao Teng (Member, IEEE) received his Ph.D. degree from the Istituto Italiano di Tecnologia and Università Cattolica del Sacro Cuore, Italy, in 2023. He earned his M.S. and B.S. degrees in automation from the South China University of Technology, Guangzhou, China, in 2019 and 2016, respectively. He is currently a postdoctoral fellow at the Hong Kong Centre for Logistics Robotics and The Chinese University of Hong Kong. His research focuses on human–robot interaction, robot learning, and optimal control.

Yongsheng Ou (Senior Member, IEEE) received the B.Sc. degree in mechanical and electrical engineering from the Beijing University of Aeronautics and Astronautics in 1995, the M.Sc. degree in electrical engineering from the Institute of Automation, Chinese Academy of Sciences, in 1998, and the Ph.D. degree in automation and computer-aided engineering from The Chinese University of Hong Kong in 2004. He is currently a Professor at the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. He is the author of more than 200 papers in major journals and international conferences and the coauthor of the monograph on Control of Single Wheel Robots (Springer, 2005). His research interests include developing service robotic systems, intelligent control, and robot navigation.

Darwin Caldwell (Fellow, IEEE) received the B.S. and Ph.D. degrees in robotics from the University of Hull, Hull, Yorkshire, U.K., in 1986 and 1990, respectively. He received the M.S. degree in management from the University of Salford, Salford, U.K. He is the Research Director and the Director of the Department of Advanced Robotics at the Italian Institute of Technology, Genoa, Italy, and an Honorary Professor at the Universities of Sheffield, Bangor, U.K., Kings College London, London, U.K., and Tianjin University, Tianjin, China.

Fei Chen (Senior Member, IEEE) received his B.S. degree in Computer Science from Xi'an Jiaotong University, China, in 2006, his M.S. degree in Computer Science from Harbin Institute of Technology, China, in 2008, and his Dr.Eng. degree from Fukuda Laboratory, Nagoya University, Japan, in 2012. From 2013 to 2020, Dr. Chen worked as a researcher at the Department of Advanced Robotics, Italian Institute of Technology, Genoa, Italy, where he headed the Active Perception and Robot Interactive Learning Laboratory. Since 2020, he has been working as an Assistant Professor, leading the Collaborative and Versatile Robots Laboratory, with T-Stone Robotics Institute (CURI) and Department of Mechanical and Automation Engineering at The Chinese University of Hong Kong, and contributing as Co-PI to the Hong Kong Center for Logistics Robotics. His research focuses on robot manipulation, human-robot collaboration, and embodied AI for humanoid robots. Dr. Chen holds several leadership positions within the IEEE community. He is co-chair of IEEE Robotics and Automation Society Technical Committee on Neuro-Robotics Systems. He also serves as Associate Editor for IEEE Transactions on Cognitive and Developmental Systems, IEEE Transactions on Emerging Topics in Computational Intelligence, and Frontiers in Neurorobotics. Additionally, he acts as chairs with several international conferences and workshops such as ICRA2024, RO-MAN2024, ROBIO2024.