Non-Prehensile Tool-Object Manipulation by Integrating LLM-Based Planning and Manoeuvrability-Driven Controls

Hoi-Yin Lee, *Student Member, IEEE*, Peng Zhou, *Member, IEEE*, Anqing Duan, Wanyu Ma, Chenguang Yang, *Fellow, IEEE*, and David Navarro-Alarcon, *Senior Member, IEEE*

***Abstract*—The ability to wield tools was once considered exclusive to human intelligence, but it's now known that many other animals, like crows, possess this capability. Yet, robotic systems still fall short of matching biological dexterity. In this paper, we investigate the use of Large Language Models (LLMs), tool affordances, and object manoeuvrability for non-prehensile tool-based manipulation tasks. Our new method employs LLMs based on scene information and natural language instructions for symbolic task planning. Using a new tool affordance model derived from visual feedback, we develop a manoeuvrability-driven controller to guide the robot's tool utilisation and manipulation actions. The proposed methodology is evaluated with experiments to prove its effectiveness under various manipulation scenarios.**

***Index Terms*—Tool manipulation; Task allocation; Affordance modelling; Task and motion planning; Robotics.**

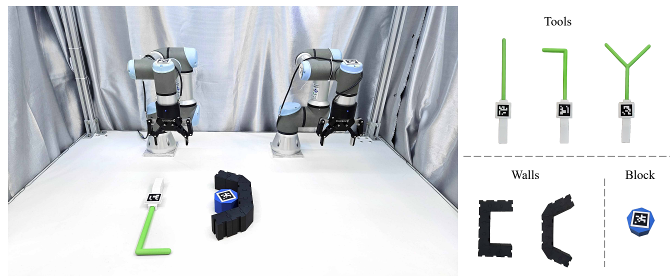

Fig. 1. Tool-Object manipulation in a dual-arm robotics system with environmental constraints using the non-prehensile approach.

I. Introduction

BEING able to use tools is a widely recognised indicator of intelligence across species [1], [2]. Humans, for instance, have demonstrated mastery of tool use for over two million years [3]. The ability to use tools is invaluable as it extends an organism's reach and enhances its capacity to interact with objects and the environment [1]. Being able to understand the geometric-mechanical relations between tools-objects-environments allows certain species (e.g., apes and crows [4]) to reach food in narrow constrained spaces. The same principles of physical augmentation and its associated non-prehensile manipulation capabilities also apply to robotic systems [5]. For example, by instrumenting them with different types of end-effectors, robots can (in principle) dexterously interact (e.g., push and flip) with objects of various shapes and masses akin to its biological counterpart [6]–[8]. However, developing this type of manipulation skill is still an open research problem. The goal of this paper is to develop a methodology to effectively transport objects through non-prehensile tool manipulation actions.

Effective tool utilisation by a robot involves primarily two aspects: (1) task planning and (2) tool movement [9]–[11]. Task planning is typically regarded as a cognitive high-level process in robotics, mainly used for environmental reasoning, task decomposition, allocation of action sequences, etc. [12]. However, recent trends have been pushing towards the use of LLMs to leverage the domain knowledge for semantically decomposing and planning the execution of manipulation tasks [13]–[16]. Some examples of this directions include [14], [15], which developed an environmental feedback-based system for context-aware improvement planning. Leveraging the generative capabilities of LLMs, motion sequences can be generated for robots as demonstrated in [16]–[18]. The combination of traditional motion planners with LLMs has been explored in [13], [19], [20].

In addition to task planning, various manipulation methodologies have been developed to model the relation between tools and objects [21]. The success of a given tool-object manipulation task largely depends on the appropriate selection of the tool. For example, robots can identify the tool type, potential uses, and contact approaches based on the tool's geometry, see e.g., [2], [9]. In [22], tool features are learned through observation of the task's effects and experimental validation of feature hypotheses. Affordance models are a common technique used for tool feature selection [23]–[25] and tool classification [25]–[27]. The relation between tool actions and its effects on objects is explored in [27], [28], where robots acquire affordance knowledge through predefined actions (e.g., pull, push, rotate). Recently, researchers have also explored the use of LLM in accelerating affordance learning in tool manipulation [2]. Some works have studied tool-based manipulation under constraints and from demonstrations [29], [30]. Non-prehensile object manipulation strategies have been used in [31], [32].

Although there are many studies on robotic tool use, the col-

H.-Y. Lee, A. Duan, W. Ma, and D. Navarro-Alarcon are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong. (contact e-mail: dna@ieee.org)

P. Zhou is with the Department of Computer Science, The University of Hong Kong.

C. Yang is with the Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK. (e-mail: cyang@ieee.org).

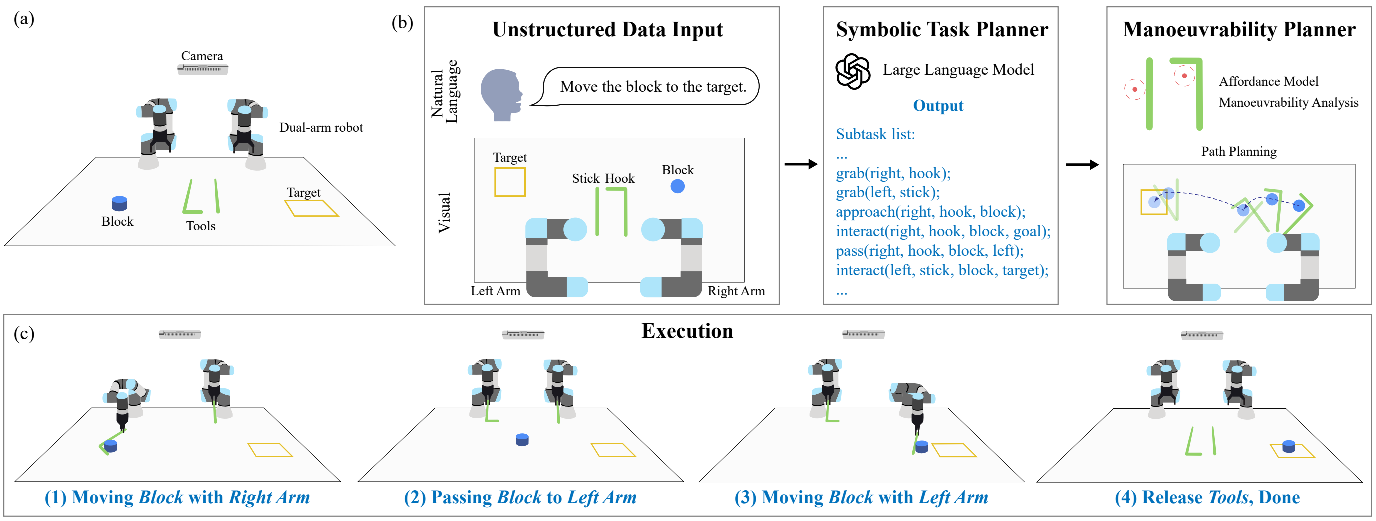

Fig. 2. (a) The task environment includes a camera for real-time top-view capturing, a dual-arm robot, tool(s), and a blue block to be manipulated to the target location. (b) The architecture of our system: Unstructured data input is converted to a subtask list in the symbolic task planner with an LLM, a manoeuvrability-driven planner to compute the tool's manoeuvrability and generate an affordance-oriented motion and path. (c) Execution process of the result given by the system: dual-arm robots take turns pushing the blue block from one side to another via collaboration.

laborative tool-based object manipulation by dual-arm systems based on non-prehensile actions remains an underexplored problem. To address this research gap, in this work we propose an novel LLM-based manoeuvrability-driven method with the following original contributions: (1) We develop an effective model to represent the geometric-mechanical relations and manoeuvrability of tools and objects; (2) We propose a non-prehensile strategy to manoeuvre objects under different constraints with tools; (3) We evaluate the performance of the proposed methodology with real-world experiments on a dual-arm robotic system.

The rest of the manuscript is organised as follows: Sec. II presents the methodology, Sec. III presents the results, Sec. IV gives final conclusions.

II. Methodology

A. Problem Formulation

Consider a dual-arm robotic system using a tool to manipulate a block at a far distance (see Fig. 1). Given the input is a free-form language task $\mathbf{L}$ (e.g., "move the block to Point B"), we apply a high-level symbolic planner (i.e., a LLM) to decompose the task into multiple subtasks $\mathbf{l}_i$, $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, \dots\}$ where $\mathbf{L}$ contains a list of pre-defined motion functions $\mathbf{l}_i$.

We define a *tool* as a manipulable object that is graspable by a robot, a *manipulandum* [9] as an object (e.g. a block) that is manipulated via a tool, and a *wall* as a static non-manipulable object. Tool use by robots is challenging as the tools can have various shapes, the environment can be dynamic, and the contact between the tool and the manipulandum may be hard to maintain in a long-horizon task. Depending on the geometric features of a tool and a wall, the available affordance for manoeuvring a manipulandum may be different. Affordance here refers to the available action-effects offered by the tool or the environment. In this work, we classify affordance into two types: active and passive. Active affordance is given from a manipulable object, i.e. a tool, and it is directly related to the manoeuvrability when driving a manipulandum. Passive affordance is given from a static non-manipulable object.

To derive our methodology, the following setup assumptions are made: (1) The manipulation motion is planar, and (2) the size of the manipulandum is not larger than any one of the segments of the tool. Throughout this paper, we denote "tool-based object manipulation" as TOM, and "tool-based object manipulation under environmental constraints" as TOME. Also, we use $\mathbf{p}^{\circ}$ to represent the 2D pose of an object $\circ$. The complete architecture of our method is depicted in Fig. 2.

B. LLM-Based High-Level Symbolic Task Planner

To obtain a valid task decomposition for a long-horizon task, the system needs to understand the requirements and generate an executable subtask list. We develop a symbolic task planner that takes natural language instructions with scene descriptions as input, and outputs a list of high-level subtasks. The list involves the tool selection/sharing between two arms, the sequence to manipulate the tool with the manipulandum, and the interaction between the two arms. The model is fine-tuned with around 20,000 lists of example data. These two data sets are generated randomly with different environment settings, such as different locations of the manipulandum, target destination, robots, and tool shapes.

The system interprets the provided high-level task $\mathbf{L}$, which can have a structure like "Please move the blue block to the right-hand side", "Can you push the block to the target?", etc. Visual information of the scene is grounded to the system from the observation data $\mathbf{o}$, where $\mathbf{o}$ is composed of a series of data points, such as the pose of the block (manipulandum), tools, robots, and walls. The system embeds the environmental information with the task instruction to produce a desired configuration requirement, denoted as $\{\mathbf{p}^{\text{obj}}, \mathbf{p}^{\text{target}}, \dots\} \leftarrow f(\mathbf{L}, \mathbf{o})$ where $f(\mathbf{L}, \mathbf{o})$ is the embedded result.

The LLM interprets the output of $f(\mathbf{L}, \mathbf{o})$ to generate a subtask list $\{\mathbf{l}_1, \mathbf{l}_2, \dots\} \leftarrow f_{\text{llm}}(f(\mathbf{L}, \mathbf{o}))$ where $\mathbf{l}_i$ is a

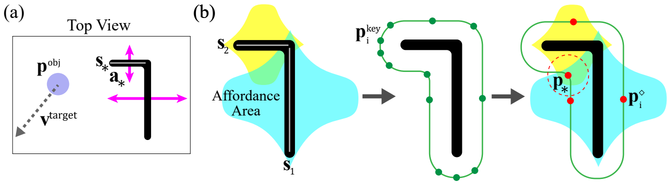

Fig. 3. (a) Affordance vectors are shown in pink arrows. Grey arrow is $\mathbf{v}^{\text{target}}$ and the desired affordance vector is denoted as $\mathbf{a}_*$. (b) shows the manoeuvrability analysis flowchart: affordance area is visualised with the Gaussian function in yellow and blue; expand and downsample the tool's shape to get key points $\mathbf{P}^{key}$ (green colour dots); combine the affordance area with the key points $\mathbf{P}^{key}$ to get the non-redundant points $\mathbf{P}^{\diamond}$ (red dots), and combine the affordance $\mathbf{a}_*$ found in (a) to obtain the position for the manipulandum to be at with the tool (labelled as $\mathbf{p}_*$ with a red dot) and the highest manoeuvrability region is shown with a dashed red circle.

subtask describing the manipulation phase of each robot [33] and is corresponding to a high-level robot motion function. The motion functions are designed to be simple and specify a short-term goal of the concerned object (these functions omit low-level motion commands). For simplicity, here we use *m* to represent manipulandum in the following function definitions. We use $\texttt{grab}(arm, tool)$ for grabbing a *tool* with the robot *arm*; $\texttt{approach}(arm, tool, m)$ for approaching the location of *m* with *tool* using *arm*; $\texttt{interact}(arm, tool, m, goal)$ for moving *m* to the *goal* location with the *tool*; $\texttt{stepping}(arm, tool, m)$ for moving *m* out from the bounded are with the *tool* of the *arm* through contact pulsing motions; $\texttt{pass}(arm1, tool, m, arm2)$ for passing *m* to another arm's workspace; $\texttt{release}(arm, tool)$ for releasing the *tool* back to its original place with the *arm*.

A sample motion task with a dual-arm robot can given as: $\{\texttt{pass}(right, hook, block, left); \texttt{approach}(left, stick, block)$ $\texttt{interact}(left, stick, block, target); \dots\} \leftarrow f_{\text{llm}}(f(\mathbf{L}, \mathbf{o}))$, where both arms take turns manipulating the block. The right arm passes the block to the left by pushing it to an area where both arms can reach it. The left arm approaches the block with a stick and manipulates the block to the target. To this end, the symbolic task planner converts the unstructured data to a series of motion functions, including robot motion, tool planning, manipulation sequence, and collaboration.

C. Visual Affordance Model

Tools can have various shapes and complex structures. In this paper, we focus on the following tool geometries: a stick, an L-shaped hook, and a Y-shaped hook. Affordances are related to the geometric features of a tool [34]. To analyse the possible affordances, we divide the tool into smaller segments (i.e. a line), and denote them as $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$ where $\mathbf{s}_i$ and $\mathbf{s}_{i+1}$ are segments next to each other. We compute the normal vectors of the segment at the middle point and scale them by half of the segment's length. This is done to weight the affordance effect this regions carries. There are two affordance vectors per segment $\mathbf{s}_i$, each pointing in opposite directions, as depicted in Fig. 3(a). Let us define $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{2n}\}$ as the structure that contains all the affordance vectors $\mathbf{a}_i$, for n as the number of segments.

To determine which affordance vector $\mathbf{a}_i$ will be used to interact with the manipulandum, we compare the similarity between $\mathbf{a}_i$ and the vector from the manipulandum's position to the target point $\mathbf{v}^{\text{target}}$ by:

$$\theta_i = \cos^{-1}\left(\frac{\mathbf{v}^{\text{target}} \cdot \mathbf{a}_i}{\|\mathbf{v}^{\text{target}}\|\|\mathbf{a}_i\|}\right) \tag{1}$$

where θ_i is the similarity score. The optimal affordance vector $\mathbf{a}_*$ and its according segment $\mathbf{s}_*$ are found by locating the vector that has the minimum similarity score $\arg\min_{\mathbf{a}}(\Theta)$ where $\Theta = \{\theta_1, \theta_2, \dots\}$.

D. Manoeuvrability Analysis

A tool can push the manipulandum from the side, from the tip, or other areas. However, the relative location of the manipulandum respective to the tool affects its manoeuvrability. In other words, the affordance provided by the tool is proportional to manoeuvrability. Consider using a rotating stick to push an object with its end tip. In this situation, the tool may lose contact with the manipulandum as it rolls outwards, hence, the manoeuvrability of this point is low. On the other hand, the midpoint of the stick has a high manoeuvrability, which proportionally decreases as the contact point is further away from the midpoint. We can model this behaviour with a Gaussian function, where its centre is the segment's centre and the peak height is half the segment's length, see Fig. 3(b). We refer to this region as an affordance area.

All the pixels in the affordance area of $\mathbf{s}_i$ are set to 1 in an image frame $\mathbf{I}_i$ and the rest to 0, which creates a binary image; This process is repeated for all segments. All binary images are then summed as $\hat{\mathbf{I}} = \sum_{i=1}^{n} \mathbf{I}_i$ where n is the number of segments. The affordance of tool segment is quantified with the (normalised) manoeuvrability matrix: $\mathbf{M} = \hat{\mathbf{I}}/\hat{I}_{max}$, for $\hat{I}_{max}$ as the maximum value in $\hat{\mathbf{I}}$. Tool regions with high values in the image $\mathbf{M}$ reflect a high manoeuvrability.

These computed manoeuvrability values are useful to determine the location where the tool interacts with the manipulandum. To determine the centre of the object, we then expand the contour of the tool by the object's radius r^{obj}. This contour is downsampled with the Ramer-Douglas-Peucker algorithm [35], then, parameterised with the spline fitting technique reported in [36]. To extract key features of the tool geometry, we use a sliding window strategy to examine a small number of neighbouring points. If there exists a point where its curvature is larger than a threshold in the local neighbourhood, we consider this point as one of the feature points.

To compute the minimal number of key points (denoted as $\mathbf{P}^{key} = \{\mathbf{p}_1^{key}, \mathbf{p}_2^{key}, \dots\}$) that capture the highest manoeuvrability among feature points, we use the density-based clustering algorithm from [37]. By integrating the affordance areas we obtained earlier, we can filter out some redundant key points. For example, if there exists a point $\mathbf{p}_i^{key}$ located outside the affordance area (visualised in Fig. 3(b)), we consider this point as redundant. All the non-redundant points are then grouped into $\mathbf{P}^{\diamond} = \{\mathbf{p}_1^{\diamond}, \mathbf{p}_2^{\diamond}, \dots\}$. To find the point in $\mathbf{P}^{\diamond}$ with the highest manoeuvrability (defined as $\mathbf{p}_*$), we use the

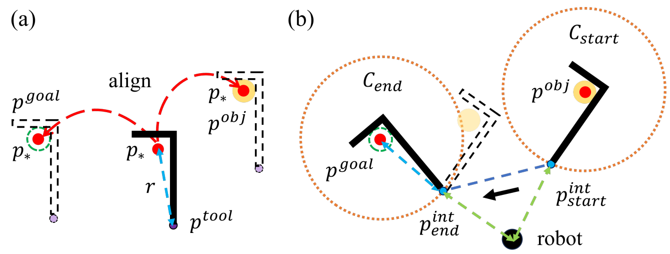

Fig. 4. (a) The tool is virtually aligned to the current object and the goal location, with $\mathbf{p}_* = \mathbf{p}^{obj}$ and $\mathbf{p}_* = \mathbf{p}^{goal}$. (b) The light blue dashed line is the radius of the orange circle $\mathbf{C}_{\text{start}}$ and $\mathbf{C}_{\text{end}}$, which equals the distance between $\mathbf{p}^{tool}$ and $\mathbf{p}_*$. The tool moves from $\mathbf{p}^{\text{int}}_{\text{start}}$ to $\mathbf{p}^{\text{int}}_{\text{end}}$ by following the dark blue dashed trajectory line.

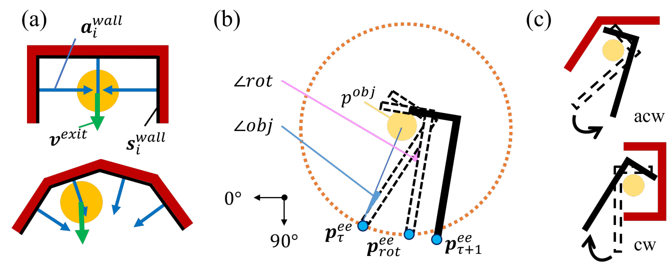

Fig. 5. (a) Walls are in red with the segment of the wall $\mathbf{s}_i^{\text{wall}}$ highlighted in black; blue arrows are the passive affordance vector and green arrows indicate the moving direction of $\mathbf{v}^{\text{exit}}$. (b) The tool pose moves from τ to $\tau+1$ by rotating with $\angle$rot and translating linearly to $\mathbf{p}^{\text{ee}}_{\tau+1}$. (c) Rotation direction of a tool: anti-clockwise and clockwise direction.

manoeuvrability matrix $\mathbf{M}$ and distance between $\mathbf{p}_i^\diamond$ and $\mathbf{a}_*$ as described in the metric below:

$$\mathbf{p}_* = \underset{\mathbf{p}_i^\diamond}{\arg\min}((1 - [\mathbf{M}]_{\mathbf{p}_i^\diamond}) + \|\mathbf{p}_i^\diamond - \mathbf{a}_*\|) \tag{2}$$

where $[\mathbf{M}]_{\mathbf{p}_i^\diamond}$ denotes to the image value of $\mathbf{M}$ at point $\mathbf{p}_i^\diamond$. The region with the highest manoeuvrability is defined as the circle (with object radius) centred at $\mathbf{p}_*$. (see Fig. 3(b))

E. Manoeuvrability-Oriented Controller

The subtask "`interact`" triggers the robot to use the selected tool to drive the manipulandum towards the desired location. In this section, we derive our method to perform this type of motion assuming that the tool approaches the object and is going to make contact with it in the subtask "`interact`".

1) Initial and Final Poses: The tool's pose corresponds to its grasping configuration, which coincides with the robot end-effector's pose when the robot grasps the tool (see Fig. 4). We use $\mathbf{p}^{\text{tool}}$ to denote the tool's grasping point (x, y coordinates) when it has not come in contact with the object. To construct a trajectory for tool-based object transport, we need to find out the tool's desired initial and final poses for the subtask "`interact`". We first define these poses (which include the orientation) of the chosen tool as $\mathbf{p}^{\text{int}}_{\text{start}}$ and $\mathbf{p}^{\text{int}}_{\text{end}}$ respectively.

To efficiently move the object, we propose a method that reduces the travel distance while ensuring continuous contact. In the first contact, we align the highest manoeuvrability point $\mathbf{p}_*$ of the tool to the object's centre $\mathbf{p}^{\text{obj}}$, where $\mathbf{p}_* = \mathbf{p}^{\text{obj}}$.

The motion trajectory of a tool, moving along the z-axis of the object's centre without displacing it can be described as a circular trajectory with the centre $\mathbf{p}^{\text{obj}}$ and radius r, where $r = \|\mathbf{p}_* - \mathbf{p}^{\text{tool}}\|$. We represent the trajectories for the initial and final configurations as $\mathbf{C}_{\text{start}}$ and $\mathbf{C}_{\text{end}}$ (see Fig. 4(a)).

The possible location for $\mathbf{p}^{\text{int,x,y}}_{\text{start}}$ will be lying on $\mathbf{C}_{\text{start}}$ and can be determined by finding a point on $\mathbf{C}_{\text{start}}$ which it is the closest point to the robot (the distance is indicated with a light green dashed line in Fig. 4(b)). Based on the tool's geometry, we can determine the orientation of the initial pose $\mathbf{p}^{\text{int}}_{\text{start}}$; The same approach applies to $\mathbf{p}^{\text{int}}_{\text{end}}$.

2) Motion Strategy: To stably move from $\mathbf{p}^{\text{int}}_{\text{start}}$ to $\mathbf{p}^{\text{int}}_{\text{end}}$, the following motion strategy is implemented to achieve the task: First, the robot aligns $\mathbf{p}_*$ with $\mathbf{p}^{\text{obj}}$ and matches $\mathbf{p}^{\text{tool}}$ with $\mathbf{p}^{\text{int}}_{\text{start}}$; then translates along the x and y axes until it reaches $\mathbf{p}^{\text{int,x,y}}_{\text{end}}$; lastly, the tool is rotated to align with the orientation of $\mathbf{p}^{\text{int}}_{\text{end}}$.

F. Application with Environmental Constraints

When moving an object across a table, we may encounter constraints from the environment, such as walls. These constraints restrict the potential movement directions of the object. Formally, a constrained area can be defined by a series of points where more than one axis of freedom of the manipulandum motion may be restricted. In this section, we focus on the motion triggered by the subtask '`stepping`'.

Consider the manipulandum is tightly confined within a concave-shaped wall, as shown in Fig. 5(a), with an unknown exit and assume that the tool can enter the constrained area. To move the manipulandum out from the bounded area with small movement space, we determine the direction from the manipulandum to the exit by considering the overall affordance of the wall boundary. We denote this direction vector as $\mathbf{v}^{\text{exit}}$, and its magnitude is defined as the minimum travel distance for the manipulandum. We consider the inner edge of the wall as a segment $\mathbf{s}_i^{\text{wall}}$ where $i = \{1, \ldots, n^{\text{wall}}\}$ and n^{wall} is the number of the wall segment. The affordance of a wall is passively provided and is defined as $\mathbf{a}_i^{\text{wall}}$ with the model shown in Sec. II-C. The passive affordance vector is the normal vector of $\mathbf{s}_i^{\text{wall}}$ located in the middle, and the magnitude is scaled to half of $\mathbf{s}_i^{\text{wall}}$ with the direction pointing towards the constrained area. The moving direction for the manipulandum to the exit can be obtained by: $\mathbf{v}^{\text{exit}} = \sum_{i=1}^{n^{\text{wall}}} \mathbf{a}_i^{\text{wall}} + \mathbf{p}^{\text{obj}}$ where $\mathbf{v}^{\text{exit}}$ integrates all passive wall affordance vectors, see 5(a).

Given that only part of the tool can enter the confined area, our primary focus is the tip of the tool. The segment connecting of the tool's tip is denoted as $\mathbf{s}^{\text{tip}}$, with its corresponding affordance vector denoted as $\mathbf{a}^{\text{tip}}$. The desired rotation angle of the end pose of $\mathbf{a}^{\text{tip}}$ is the angle of $\mathbf{v}^{\text{exit}}$.

The highest manoeuvrability region can be obtained by treating $\mathbf{v}^{\text{exit}}$ as the target vector $\mathbf{v}^{\text{target}}$, $\mathbf{a}^{\text{tip}}$ as the desired affordance $\mathbf{a}_*$, and assuming the tool is rotated such that $\mathbf{a}^{\text{tip}} = b\mathbf{v}^{\text{exit}}$ with $b > 0$ as a scaling factor. We first align $\mathbf{s}^{\text{tip}}$ to the first segment of the wall (i.e. $\mathbf{s}_1$), with $\mathbf{p}^{\text{obj}}$ inside the highest manoeuvrability region of the tool. The tool approaches the object and maintains contact with the manipulandum by minimising the distance $\|\mathbf{p}_* - \mathbf{p}^{\text{obj}}\|$.

To move in the limited area while interacting with the block, we employ a stepping approach to manipulate the block in the confined area. As the possible movement area is small and highly restricted, an incremental pulsing motion is adopted to make small adjustments with high accuracy motion control to the tool and the manipulandum. Inspired by the animal manipulation study in [4] (where a crow uses a tool to get the food from the box slot by rotating and dragging the tool outwards), we adopt a similar approach to retrieve the object from confined spaces. This strategy continuously alternates between "repositioning" the tool and incremental "rotation-dragging" the object towards the exit until it can be fully extracted. This strategy is illustrated in Fig. 5.

We define "repositioning" as moving the tool closer to the object and realigning $\mathbf{p}_*$ with $\mathbf{p}^{\text{obj}}$ by k amount. In "rotation-dragging", the tool maintains contact with the manipulandum when it rotates by a certain angle as $\angle$rot shown in Fig. 5(b) and moves outwards by extending $\overrightarrow{\mathbf{p}_\tau^{\text{ee}}\mathbf{p}_{\text{rot}}^{\text{ee}}}$ by a $w > 0$ amount.

We define τ as an action step variable and is incremented by 1 if an action (reposition/rotation-dragging) is fulfilled (i.e. $\tau = 0, 1, 2, \dots$). To control the change of action, a step function (denoted as $u(\tau)$) is implemented as a trigger with the step variable τ. This kind of non-prehensile crow-inspired behaviour can be unified and modelled as:

$$\mathbf{p}_{\tau+1}^{\text{ee}} = \begin{bmatrix} \mathbf{p}_\tau^{\text{ee},x} \\ \mathbf{p}_\tau^{\text{ee},y} \\ \phi_\tau \end{bmatrix} + u(\tau) \begin{bmatrix} k(\mathbf{p}_\tau^{obj,x} - \mathbf{p}_*^x) \\ k(\mathbf{p}_\tau^{obj,y} - \mathbf{p}_*^y) \\ 0 \end{bmatrix} + u(\tau+1) \begin{bmatrix} w(\mathbf{p}_\tau^{\text{obj},x} - r\cos(\phi_\tau) - \mathbf{p}_\tau^{\text{ee},x}) \\ w(\mathbf{p}_\tau^{\text{obj},y} + r\sin(\phi_\tau) - \mathbf{p}_\tau^{\text{ee},y}) \\ f(\phi_{\tau+1}) \end{bmatrix}$$

$$u(\tau) = \begin{cases} 0, & \text{if } \tau \text{ is odd} \\ 1, & \text{if } \tau \text{ is even} \end{cases} \tag{3}$$

where $\mathbf{p}_{\tau+1}^{\text{ee}}$ is the next target pose of the end-effector at the action step $\tau + 1$ for the affordance vector $\mathbf{a}^{\text{tip}}$ not parallel to $\mathbf{v}^{\text{exit}}$, such that $\mathbf{a}^{\text{tip}} \neq b\mathbf{v}^{\text{exit}}$. The angle of the tool at $\tau + 1$ (denoted as $\phi_{\tau+1}$) depends on the rotational direction (see Fig. 5), that $\phi_{\tau+1}$ is computed as

$$f(\phi_{\tau+1}) = \begin{cases} -\angle\text{obj} - \angle\text{rot} & \text{if direction is anti-clockwise} \\ -\phi_\tau + \pi - \angle\text{obj} - \angle\text{rot} & \text{otherwise} \end{cases} \tag{4}$$

where ϕ_τ is the tool's angle at the action step τ, $\angle$obj is the angle between the block, grasping point, and a tool's keypoint, $\angle$rot is the amount of angle to rotate.

III. Results

To validate our methodology in terms of accuracy and robustness, we have conducted around 200 experiments in a dual-arm robot system. In the experiment, two sets of UR-3 robotic arms are used and GPT 3.5 is implemented for task decomposition. Three types of tools are selected which are a stick, an L-shaped hook, and a Y-shaped hook (see Fig.1). Different tool combinations are evaluated with diverse movement directions and tasks. A RealSense D415 captures the images of the whole process. Data is passed to a Linux-based computer with the Robot Operating System (ROS) for image process and robot control. Aruco code is used for providing accurate pose tracing in real time.

These experiments include validating the task decomposition performance in a single and dual-arm robot setup, the robustness of the affordance and manoeuvrability model in various shapes of tools, and evaluating the overall performance.

A. Single-Arm Robot with a Single Tool

We first evaluate the task decomposition performance of LLM. For that, a tool and a blue block are placed on the table with the target given as shown in Fig. 6. The task is to manipulate the block within a close distance, which is sufficient for a single-arm robot. The embedded information which contains the task, the environment and the geometry of the tool is passed to the LLM. In the experiment shown in Fig. 6(a), the robot executes the subtasks generated by the high-level symbolic task planner which include: `grab`$(right, hook)$; `approach`$(right, hook, block)$; `interact`$(right, hook, block, target)$; `release`$(right, hook)$, where the right arm first moves and grabs the hook, then moves the block to the target, and lastly releases the tool back to its original place. The experiment showcases the application of the proposed affordance and manoeuvrability model in locating the highest manoeuvrability region for block transportation. During the manipulation stage, the block is kept within the highest manoeuvrability region (indicated with a red circle in Fig. 6) to receive affordance effectively from the tool. The minimisation of the error between the $\mathbf{p}^{\text{obj}}$ and the $\mathbf{p}^{\text{target}}$ for each experiment is shown in Fig. 7. These results corroborate that the proposed method can be used to actively drive a robot to manipulate an object via a tool.

B. Dual-Arm Robot with Long-Horizon Task

We then evaluate the long-horizon task performance where the block has to travel from far right to far left, far right/left to top right/left, and vice versa. The long-horizon task is evaluated with multiple tool combinations. The system observes and generates a collaborative motion plan. In the experiment shown in Fig. 8(a), the right and left arms pick up the stick and the hook respectively. The right arm uses the stick to push the block to the left side, allowing the left arm to continue the task. The robot leverages the advantage of the hook to drag the block closer to its working area and push the block to the desired location. In Fig. 8(b), the right and left arms grasped the Y-shaped tool and the stick respectively. The right arm uses the tool to pass the block to the left. The left arm uses the stick to push the block to the target location.

The long-horizon task performance is evaluated with the tool-sharing ability. Assuming there is only one tool available, it has to be shared among the dual-arm robot. Fig. 8(c) demonstrates the tool is passed to another arm once the block is pushed to the middle of the table. The block is moved accurately to the target with motion-decomposed: '`grab`; `approach`; `interact`; `pass`; `release`; `grab`; `approach`; `interact`; `release`' where the left arm releases the tool once it is done and the right picks up the tool to continue moving the block. Though the hook is in a two-link geometry, the pushing

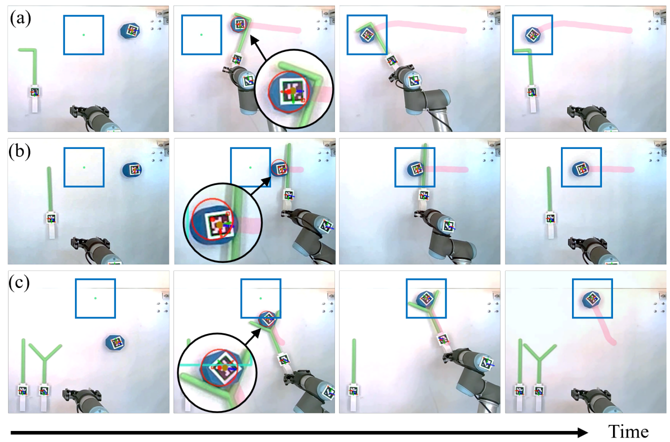

Fig. 6. Single-arm robot with a single tool: moving the block from (a) right to left with a hook, (b) right to left with a stick, (c) bottom to top with a Y-shaped tool. The trajectory of the block is reflected in the red line. The highest manoeuvrability point is indicated with a red circle.

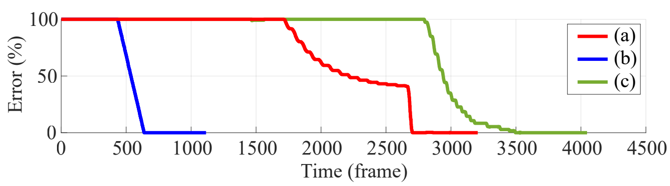

Fig. 7. Evolution of the minimisation process of the error between the current object position and the target for the tasks shown in Fig. 6.

is afforded by the right side of the tool (a single segment) with the highest manoeuvrability region.

The minimisation of the error between $\mathbf{p}^{\text{obj}}$ and $\mathbf{p}^{\text{target}}$ for each experiment is shown in Fig. 9. Similar to the single-arm robot with a single tool experiment, this long-horizon task experiment also demonstrates the robustness of the proposed methodology such that the tasks are successfully decomposed into multiple collaborative subtasks, and the highest manoeuvrability region of the tool is leveraged in block manipulation.

C. Tool-Object Manipulation in Constrained Environments

To further evaluate the performance of the model in application scenarios, different shapes of walls are constructed as shown in Fig. 8(d)–(e). Two walls are designed with 90-degree and 65-degree for the inner-angles. Maneuvering a hook within a confined space presents greater challenges compared to using a stick. Additionally, a Y-shaped hook proves unsuitable for dragging objects in tight quarters. Therefore, in this experimental study, we opt for a hook tool with a right arm to navigate effectively within the constrained environment. Similar to the previous results, Fig. 8(d)–(e) also implements the task planner successfully to decompose the task and applies the stepping controller for object manipulation. The tool first aligns its $\mathbf{s}^{\text{tip}}$ to the first segment of the wall and adopts the proposed non-prehensile stepping motion controller stated in (3). The block is dragged out from the confined area by alternating between the action of 'repositioning' and 'rotation-dragging'.

TABLE I
SUCCESS RATE COMPARISON FOR TASK DECOMPOSITION

Methods	SRST	Dual	Sharing	TOME	Overall(%)
Zero-shot learning	2/10	1/10	1/10	5/10	22.5%
Few-shot learning	3/10	1/10	2/10	7/10	32.5%
FT (1000 data)	7/10	6/10	7/10	9/10	72.5%
Ours	10/10	9/10	9/10	10/10	95.0%

During the pulsing manipulation, the block maintains contact with the highest manoeuvrability region. We visualise the contact changes between the centre of the highest manoeuvrability region $\mathbf{p}_*$ with the block in Fig. 10(a). The error between the $\mathbf{p}^{\text{obj}}$ and the wall exit for each experiment are minimised with time, as shown in Fig. 9.

D. Comparison

We analysed the affordance utilisation and provision for the selected tools. This evaluation involves assessing the frequency of contact between the block and the sides of the tool segments. In the majority of instances, the block interacts with the affordance primarily in the red region, as indicated in Fig. 10(b) and aligns closely with our proposed model.

We have compared our system with other state-of-the-art methods. In terms of the task decomposition with LLM, we compare ours with zero-shot, few-shot learning [38], [39], and a smaller dataset. The results of root mean square error (RMSE) and mean absolute error (MAE) are shown in Table I where FT states for fine-tunning, SRST states for a single-arm robot with a single tool, Dual refers to dual arms collaboration with two tools, and Sharing refers to tool-sharing collaboration. We observe that prompting (zero-shot and few-shot learning) is relatively unreliable, especially in long-horizon tasks. This may caused by insufficient manipulation examples given in the prompt. A smaller dataset with GPT 3.5 generates an acceptable result, yet, it occasionally provides unnecessary/infeasible steps in long-horizon tasks. In general, all methods demonstrate relatively positive outcomes in TOME, potentially attributed to the task's simplicity: extracting the block from the constrained environment rather than aiming for a specific destination. In summary, employing a larger dataset with GPT 3.5 yields enhanced task decomposition performance, leading to more precise results.

We assess the tool analysis method by identifying the highest manoeuvrability point across 30 tool images, with results outlined in Table II. While the total variation regularisation approach [40] is suitable for the stick case, the results are not satisfactory. The keypoints-inspired approach [41] yields comparable results to ours; however, its accuracy diminishes with increasingly complex shapes. Overall, our approach achieves a better performance in terms of manoeuvrability computation.

IV. CONCLUSION

In this paper, we present a new manoeuvrability-driven approach for tool-object manipulation. The LLM is integrated for task decomposition, generating collaborative motion sequences for a dual-arm robot system. A compact geometrical-based affordance model for describing the potential functionality and

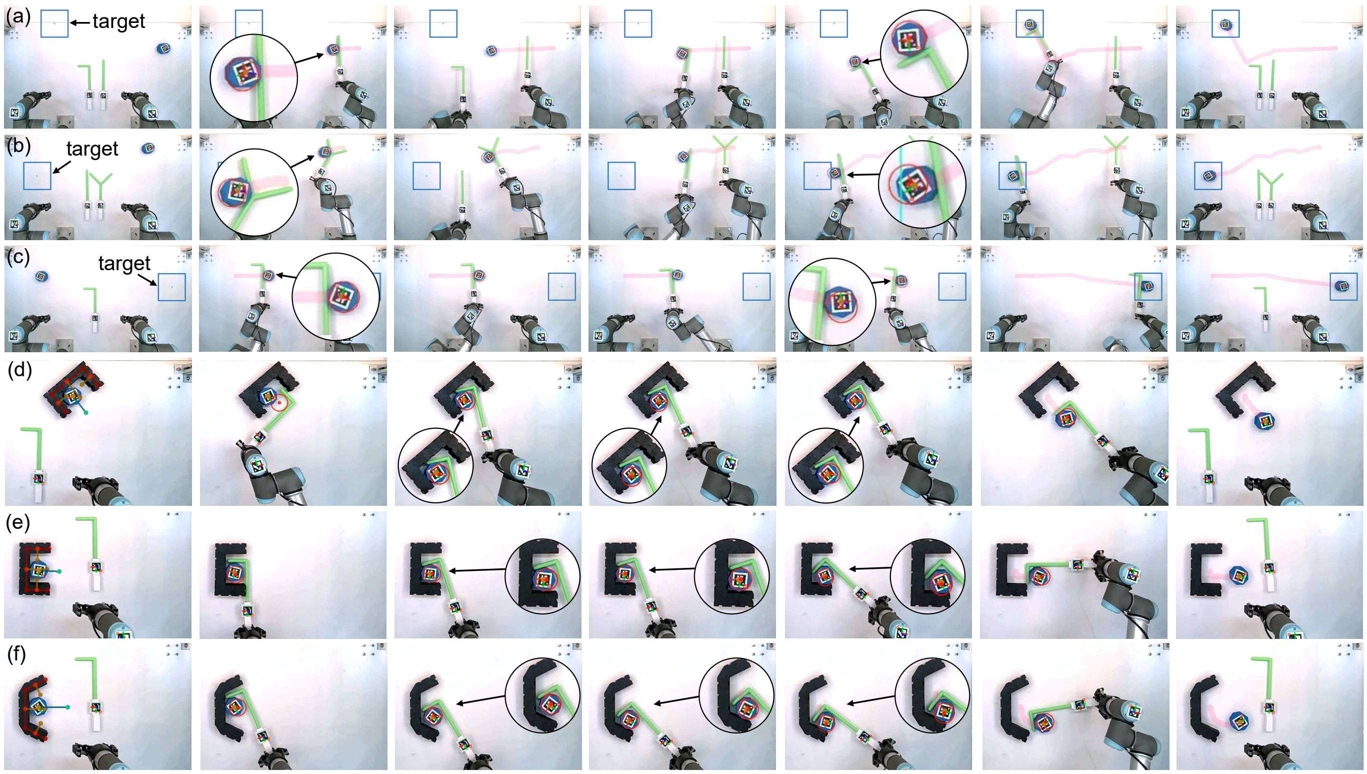

Fig. 8. Long-horizon task: moving the block from (a) far right to far left with a hook and a stick, (b) far top right to far left with a stick and a Y-shaped tool, (c) far left to far right with a hook; and (d)–(f) exit from a confined area with a stepping controller. The block trajectory is reflected in pink and the target is labelled with a blue square.

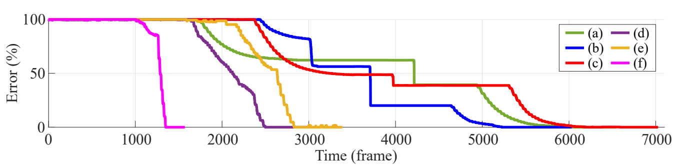

Fig. 9. Evolution of the minimisation process of the error between the current object position and the target for the tasks shown in Fig. 8.

TABLE II
ACCURACY COMPARISON OF MANOEUVRABILITY ANALYSIS

Methods	RMSE	MAE	Overall
Total variation regularisation [40]	102.4	115.9	109.2
Keypoints-based [41]	31.7	31.5	31.6
Ours	28.6	29.2	28.9

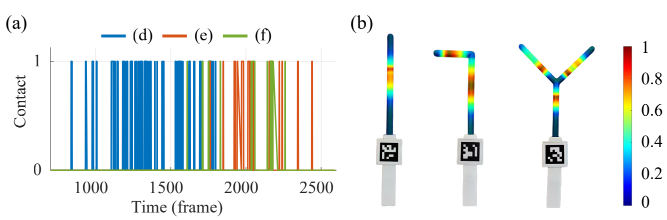

Fig. 10. (a) Stepping movement evolution of the change in contact between the block and the highest manoeuvrability point for the tasks shown in Fig. 8(d)–(f). 1 refers to in-contact and 0 refers to no contact. (b) Contact frequency of a segment side: regions depicted in deeper red indicate higher contact frequency with the block and a higher occurrence of affordance provision.

computing the highest manoeuvrability region of a tool is developed. A non-prehensile motion controller and a stepping manipulation model are derived for TOM and incremental movements in a constrained area. Experimental results are reported and analysed for the proposed methodology validation. We illustrate the performance of the proposed methods in the accompanying video https://vimeo.com/917120431.

Our method introduces a new affordance and manoeuvrability paradigm for tool-based object manipulation. To obtain a better performance, we split the model into task decomposition and mathematical motion models. However, the logical fault in the LLM's response may be unseen and thus lead to inappropriate motion. In our experiments, there are a few times that the LLM presents infeasible plans. Moreover, the current affordance model presents promising results with simple geometrical shapes. Dynamics shapes like deformable objects may be complicated to perform accurate modelling. In terms of manoeuvrability, it may be complicated to compute an accurate result for scenes with unstable illumination, low contrast in images, large height differences in objects (tools and the block), etc. We simplified these cases using ArUco code for real-time object tracking in the experiments.

For future work, we would like to extend our method to deal with deformable objects and/or environments, e.g., in the case of manipulating objects with ropes or fabrics.

References

[1] A. Stoytchev, *Robot tool behavior: A developmental approach to autonomous tool use*. Georgia Institute of Technology, 2007.

[2] A. Z. Ren, *et al.*, "Leveraging language for accelerated learning of tool manipulation," in *Conf. on Robot Learning*, 2023, pp. 1531–1541.

[3] E. S. Reed, "Applying the theory of action systems to the study of motor skills," in *Advances in psychology*. Elsevier, 1988, vol. 50, pp. 45–86.

[4] D. E. McCoy, *et al.*, "New caledonian crows behave optimistically after using tools," *Current Biology*, vol. 29, no. 16, pp. 2737–2742, 2019.

[5] L. Jamone, "Modelling human tool use in robots," *Nature Machine Intelligence*, vol. 4, no. 11, pp. 907–908, 2022.

[6] H. Huang, *et al.*, "Toward generalizable robotic dual-arm flipping manipulation," *IEEE TIE*, 2023.

[7] S. Y. Shin *et al.*, "Human-like motion generation and control for humanoid's dual arm object manipulation," *IEEE TIE*, vol. 62, no. 4, pp. 2265–2276, 2014.

[8] X. Wu *et al.*, "Cooperative manipulation of wearable dual-arm exoskeletons using force communication between partners," *IEEE TIE*, vol. 67, no. 8, pp. 6629–6638, 2019.

[9] M. Qin, *et al.*, "Robot tool use: A survey," *Frontiers in Robotics and AI*, vol. 9, p. 1009488, 2023.

[10] O. Kroemer, *et al.*, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1395–1476, 2021.

[11] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Syst.*, vol. 1, pp. 1–28, 2018.

[12] H.-Y. Lee, *et al.*, "A distributed dynamic framework to allocate collaborative tasks based on capability matching in heterogeneous multi-robot systems," *IEEE Trans. on Cognitive and Developmental Syst.*, 2023.

[13] N. Wake, *et al.*, "Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration," *arXiv preprint arXiv:2311.12015*, 2023.

[14] M. Ahn, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[15] W. Huang, *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[16] I. Singh, *et al.*, "Progprompt: Generating situated robot task plans using large language models," in *IEEE ICRA*, 2023, pp. 11 523–11 530.

[17] J. Wu, *et al.*, "Tidybot: Personalized robot assistance with large language models," *arXiv preprint arXiv:2305.05658*, 2023.

[18] S. Huang, *et al.*, "Instruct2act: Mapping multi-modality instructions to robotic actions with large language model," *arXiv preprint arXiv:2305.11176*, 2023.

[19] K. Lin, *et al.*, "Text2motion: From natural language instructions to feasible plans," *arXiv preprint arXiv:2303.12153*, 2023.

[20] M. Xu, *et al.*, "Creative robot tool use with large language models," *arXiv preprint arXiv:2310.13065*, 2023.

[21] T. Tsuji, *et al.*, "Dynamic object manipulation considering contact condition of robot with tool," *IEEE TIE*, vol. 63, no. 3, pp. 1972–1980, 2015.

[22] H. Wicaksono *et al.*, "Relational tool use learning by a robot in a real and simulated world," in *Proceedings of ACRA*, 2016.

[23] J. Brawer, *et al.*, "A causal approach to tool affordance learning," in *IEEE ICIRS*, 2020, pp. 8394–8399.

[24] N. Saito, *et al.*, "Tool-use model considering tool selection by a robot using deep learning," in *IEEE ICHR*, 2018, pp. 270–276.

[25] P. Zech, *et al.*, "Computational models of affordance in robotics: a taxonomy and systematic classification," *Adaptive Behavior*, vol. 25, no. 5, pp. 235–271, 2017.

[26] K. P. Tee, *et al.*, "Towards emergence of tool use in robots: Automatic tool recognition and use without prior tool learning," in *IEEE ICRA*, 2018, pp. 6439–6446.

[27] J. Sinapov *et al.*, "Detecting the functional similarities between tools using a hierarchical representation of outcomes," in *IEEE ICDL*, 2008, pp. 91–96.

[28] S. Forestier *et al.*, "Modular active curiosity-driven discovery of tool use," in *IEEE ICIRS*, 2016, pp. 3965–3972.

[29] S. Ding, *et al.*, "Task-oriented adaptive position/force control for robotic systems under hybrid constraints," *IEEE TIE*, 2024.

[30] H. Ochoa *et al.*, "Impedance control architecture for robotic-assisted mold polishing based on human demonstration," *IEEE TIE*, vol. 69, no. 4, pp. 3822–3830, 2021.

[31] M. Selvaggio, *et al.*, "Non-prehensile object transportation via model predictive non-sliding manipulation control," *IEEE Trans. on Control Systems Technology*, 2023.

[32] M. B. Imtiaz, *et al.*, "Prehensile and non-prehensile robotic pick-and-place of objects in clutter using deep reinforcement learning," *Sensors*, vol. 23, no. 3, p. 1513, 2023.

[33] W. Huang, *et al.*, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[34] T. Mar, *et al.*, "Multi-model approach based on 3d functional features for tool affordance learning in robotics," in *IEEE ICHR*, 2015, pp. 482–489.

[35] D. H. Douglas *et al.*, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the int. journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[36] A. Fowler *et al.*, "Cubic spline: A curve fitting routine," Union Carbide Corp., Oak Ridge, Tenn. Y-12 Plant, Tech. Rep., 1966.

[37] H.-P. Kriegel, *et al.*, "Density-based clustering," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[38] T. Brown, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[39] P. Liu, *et al.*, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[40] M. Pragliola, *et al.*, "On and beyond total variation regularization in imaging: the role of space variance," *SIAM Review*, vol. 65, no. 3, pp. 601–685, 2023.

[41] L. Manuelli, *et al.*, "kpam: Keypoint affordances for category-level robotic manipulation," in *The Int. Symposium of Robotics Research*. Springer, 2019, pp. 132–157.

Hoi-Yin Lee received her B.Eng. degree in Mechanical Engineering from The Hong Kong Polytechnic University of Hong Kong (PolyU), Hong Kong, in 2021. She was a visiting scholar at Bristol Robotics Laboratory, United Kingdom, in 2024. She is currently pursuing her Ph.D. degree in Mechanical Engineering at PolyU. Her research interests include tool manipulation, multi-robot systems, perceptual robots, image processing, and automation.

Peng Zhou received his Ph.D. degree in robotics from PolyU, Hong Kong, in 2022. In 2021, he was a visiting Ph.D. student at KTH Royal Institute of Technology, Stockholm, Sweden. He is currently a Research Officer at the Centre for Transformative Garment Production and a Postdoctoral Research Fellow at The University of Hong Kong. His research interests include deformable object manipulation, robot reasoning and learning, and task and motion planning.

Anqing Duan received his bachelor's degree in mechanical engineering from Harbin Institute of Technology, Harbin, China, in 2015, his master's degree in mechatronics from KTH, Sweden, in 2017, and his Ph.D. degree in robotics from the Italian Institute of Technology and the University of Genoa, Italy, in 2021. Since 2021, he has been a Research Associate with The Hong Kong Polytechnic University. His research interest includes robot learning and control.

Wanyu Ma received the B.Eng. and M.Eng. degrees in control science and engineering from the Harbin Institute of Technology, China, and the Ph.D. degree in mechanical engineering from The Hong Kong Polytechnic University, Hong Kong. She is currently a Postdoctoral Fellow at the Department of Surgery, The Chinese University of Hong Kong, Hong Kong. Her research interests include robot manipulation, machine intelligence and human-robot interaction.

Chenguang Yang (Fellow, IEEE) received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xian, China, in 2005, and the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010. He performed postdoctoral studies in human robotics at the Imperial College London, London, U.K from 2009 to 2010. He is Chair in Robotics with Department of Computer Science, University of Liverpool, UK. He was awarded UK EPSRC UKRI Innovation Fellowship and individual EU Marie Curie International Incoming Fellowship. As the lead author, he won the IEEE Transactions on Robotics Best Paper Award (2012) and IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award (2022). He is the Corresponding Co-Chair of IEEE Technical Committee on Collaborative Automation for Flexible Manufacturing. His research interest lies in human robot interaction and intelligent system design.

David Navarro-Alarcon (Senior Member, IEEE) received his Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong in 2014. He is currently an Associate Professor at the Department of Mechanical Engineering at The Hong Kong Polytechnic University (PolyU). His current research interests include perceptual robotics and control theory. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS.