*CAAI Transactions on Intelligence Technology*

WILEY

ORIGINAL RESEARCH OPEN ACCESS

# SFK: Shape- and Function-Grounded Keypoint Representation for Sequential Manipulation

Yaxin Liu | Yan Liu | Xinyu Song | Ruiqing Han | Haiyong Liu | Bo Huang | Yufeng Yao | Ming Zhong

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, Heilongjiang, China

**Correspondence:** Yufeng Yao (yufeng.yao@hit.edu.cn) | Ming Zhong (zhongming@hit.edu.cn)

**Received:** 10 November 2025 | **Revised:** 11 March 2026 | **Accepted:** 2 April 2026

**Keywords:** hierarchical framework | keypoint representation | sequential manipulation

## ABSTRACT

Sequential manipulation is the process by which robots perform multiple interdependent steps to accomplish composite tasks, demanding tight integration of perception, planning and execution. Existing methods incorporate explicit features such as category, semantics, 6D pose or affordance to enhance consistency, yet single-feature representations limit generalisation and adaptability across diverse task stages. To overcome these limitations, we propose a shape- and function-grounded keypoint (SFK) representation that provides interpretable, task-relevant object abstractions. The SFK representation preserves topological consistency, enabling the inference of functional semantics, 6D pose, spatial constraints and task-oriented grasp configurations. Building upon this, a keypoint-driven hierarchical framework is developed to unify perception, planning and execution, thereby enabling task-level action reasoning conditioned on keypoint representations and coherent multistep action execution. For evaluation, a keypoint dataset containing over 1600 RGB images of 29 household objects is constructed, and extensive experiments demonstrate task-relevant keypoint generation with reduced computational time, as well as higher task execution success rates and broader task coverage.

## 1 | Introduction

Robots, as the embodiment of artificial intelligence (AI) interacting with the physical world, have attracted widespread attention in tasks such as bin-picking, meal preparation, feeding, dressing and assembly. Given that these tasks consist of multiple interdependent steps, achieving sequential manipulation that tightly integrates perception, planning and execution is a central challenge in embodied AI research [1].

In sequential manipulation, perception actively extracts environmental attributes and derives task-relevant high-level representations from observations and instructions [2], providing essential priors for downstream processes. Planning then translates abstract task instructions into executable kinematic plans [3], thereby grounding the instructions in physical execution. However, task-irrelevant grasps [4], fixed trajectories [5] or improper skill combinations [6] may hinder execution, and the system should dynamically adapt motion to unstructured environments during sequential manipulation.

Traditional preprogramming is ineffective for composite tasks requiring adaptation to environmental variability. Imitation learning enables robots to extract task features from human demonstrations but suffers from high data collection costs [7]. Reinforcement learning discovers optimal strategies via reward feedback [8], yet its policies often exhibit limited generalisation beyond homogeneous tasks. Data-driven methods leverage large-scale synthetic data to model robot–environment interactions [9], but limited model capacity and growing dataset complexity hinder their scalability in sequential manipulation. Consequently, recent work combines data-driven learning,

Yaxin Liu and Yan Liu contributed equally to this work.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

which excels at feature extraction and subtask generalisation, with explicit representations to introduce logical constraints and interpretability for sequential manipulation [10].

In explicit representations, category information provides coarse task priors that assist scene understanding but lacks the geometric precision required for sequential manipulation [11]. In response to this limitation, 6D pose estimation is employed to recover category-level geometric information. However, voxel representations derived from 6D poses contain redundant spatial information [12]. To further improve geometric understanding, category-level detection is extended to segmentation to capture more precise object shapes but remains insufficient for fine-grained control [13]. To capture task-relevant functional information, component-level segmentation with semantic labels has been introduced, which enhances task-oriented grasping and imposes task-specific constraints [14]. Keypoint-based methods are further proposed to enhance object abstraction by encoding functional keypoints and relational constraints [15], thereby reducing redundant functional information resulting from component-level segmentation. Nevertheless, existing keypoint schemes are typically task-specific and lack consideration of geometric and functional properties of diverse objects, which limits their generalisation across composite tasks. These limitations motivate the development of a fine-grained, interpretable representation that can be transformed across modalities without additional annotation, thereby enabling a unified framework that supports instruction reasoning, task-oriented grasping, task-related spatial constraints, trajectory planning, obstacle avoidance and skill execution.

This paper proposes a shape- and function-grounded keypoint (SFK) representation for building interpretable, task-relevant object abstractions. Based on this representation, a keypoint-driven hierarchical framework is developed to unify perception, planning and execution, enabling task reasoning and operation for composite manipulation. As illustrated in Figure 1, task-relevant keypoints are identified in real time by the visual module, whereas task instructions are provided as text sequences composed of semantic labels. During planning, keypoint information and commonsense priors generate prompts that guide a large language model (LLM) to infer an intention sequence linking keypoint semantics to corresponding actions. The instruction grounding stage progressively maps this abstract intention to executable layers, including (a) task-oriented grasp generation, (b) 6D pose estimation via keypoint

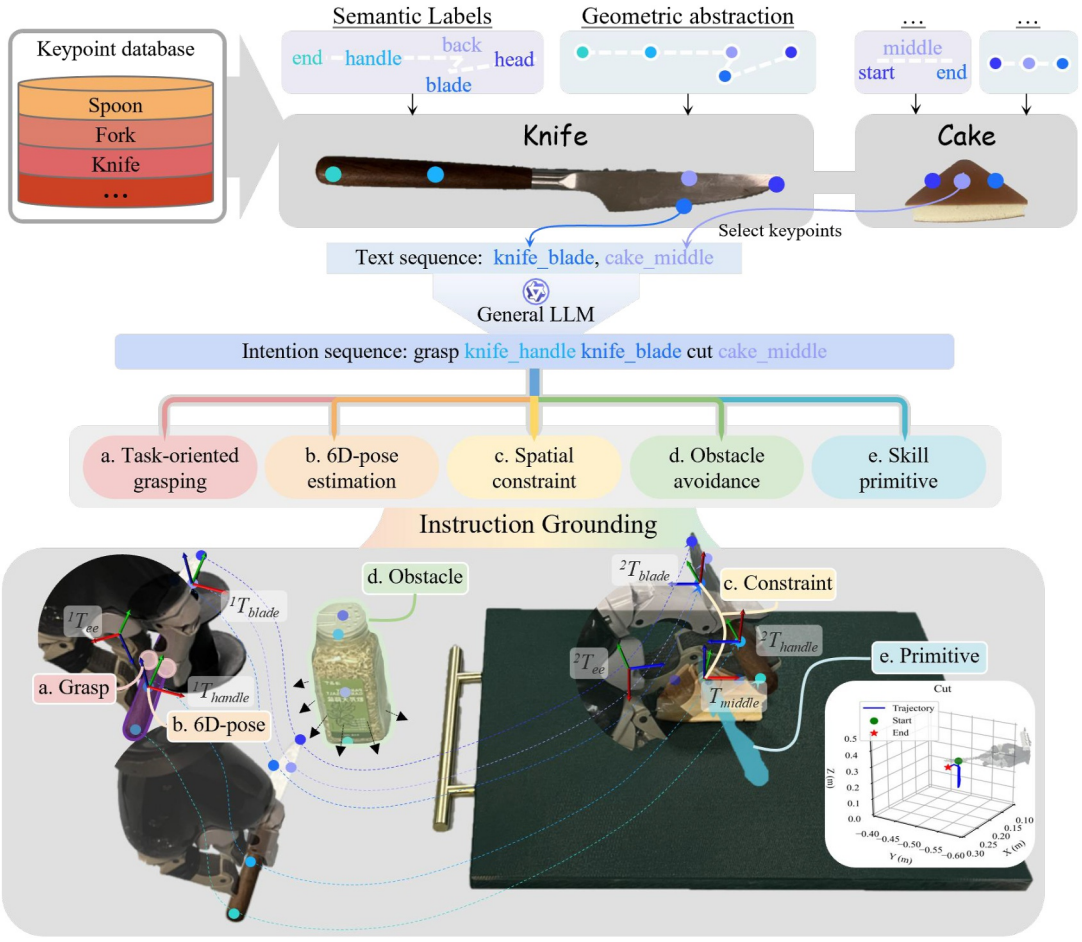



**FIGURE 1** | Illustration of the keypoint-driven framework for task planning and execution. The framework constructs semantic labels and geometric abstractions of objects to achieve a unified representation of shape and function and supports optional human-in-the-loop intent grounding, whereas keypoint interpretation, task reasoning and execution are handled within the system. Taking the example of 'cutting a cake with a knife', the keypoints 'knife_blade' and 'cake_middle' are used to generate an action intention sequence via a large language model, which is then grounded to physical execution.

topology, (c) target pose determination based on task logic and spatial constraints, (d) obstacle cloud generation using keypoint prompts and SAM and (e) trajectory planning and skill execution for manipulation. For evaluation, a keypoint dataset containing over 1600 RGB images of 29 household objects is constructed, and extensive experiments are conducted, including keypoint generation comparison, robustness analysis, task-level reasoning evaluation and real-world manipulation tasks.

Contribution:

1. This paper presents a keypoint-driven hierarchical manipulation system that unifies perception, planning and execution through a shared intermediate representation, enabling consistent information flow across task reasoning, grasping, motion planning and skill execution.
2. This paper introduces an open-source keypoint dataset of 29 common household objects annotated with shape- and function-related keypoints, supporting reproducible training and evaluation with mainstream open-source perception algorithms.
3. Extensive experiments validate the efficiency and robustness of keypoint generation under out-of-distribution and occlusion conditions, as well as the effectiveness of the proposed framework in real-world manipulation tasks.

## 2 | Related Work

Existing studies on keypoint extraction strategies can be categorised into two types. Learning-based approaches observe motion-relevant regions from demonstration data to extract task-specific keypoints of objects, whereas rule-based approaches define keypoints a priori according to the objects' functional logic and spatial constraints.

### 2.1 | Learning-Based Keypoints Extraction

Qin et al. propose a self-supervised framework for learning keypoint representations through robot–environment interactions, eliminating manual annotations and enabling grasping, placing and hitting tasks [16]. Lin et al. design an end-to-end regression network leveraging keypoint offsets for 6D pose estimation, facilitating small-object grasping and placement [17]. Ma et al. propose a graph-based method representing deformable objects as interacting keypoints extracted via unsupervised learning, achieving rope and fabric manipulation through keypoint-level model predictive control [18]. Simeonov et al. introduce a neural-descriptor system based on query points that generalises grasping and placing to unseen objects from a few demonstrations [19]. Zhang et al. present a contrastive network aligning simulated and real garments in an embedding space to infer physical parameters and enable active grasping from multiview RGB-D inputs [20]. Bechtle et al. fuse visual and proprioceptive cues via an autoencoder to detect keypoints and regress virtual joints for self-supervised grasping and placement [21]. Gao et al. extract sparse, execution-independent keypoints from human videos for imitation learning, achieving grasping, tipping and insertion via geometric constraints and motion primitives [22].

Existing supervised and self-supervised keypoint methods mainly capture geometric or motion cues without modelling functional semantics or interaction relationships, limiting interpretability and transferability. Their reliance on large-scale interaction or simulated data also impedes generalisation to real robots. As a result, the above keypoint extraction is task-specific and lacks semantic and functional consistency across objects and tasks.

Recent studies leverage vision-language models (VLMs) for keypoint extraction in zero- or few-shot learning. Palo et al. distil 3D keypoints from robotic operations without labels and encode their trajectories as text prompts for few-shot learning, though the resulting keypoints lack semantics and limit interpretability and generalisation [23]. Huang et al. propose a VLM-based framework that generates semantic keypoints and filters task-oriented grasps via spatial reasoning but remains constrained by VLMs' 2D understanding of complex 3D surfaces [24]. To address this, Huang et al. propose relational spatio-temporal reasoning, producing semantic 3D keypoints and predicting interobject spatial constraints to drive task sequences [10]. Pan et al. develop an open-vocabulary robotic system that combines VLM-based task decomposition and semantic planning with low-level 6D pose tracking, bridging high-level reasoning and geometric precision [25]. Zhang et al. further propose task-driven keypoint selection that identifies minimal keypoints in simulation and transfers them to real execution, but its hyperparameter sensitivity and lack of explicit semantics hinder interpretability and multitask generalisation [26].

Although VLMs provide some high-level semantic labels, their keypoints lack operational physical and geometric constraints, restricting controllability in task planning. Moreover, the gap between 2D semantics and 3D geometric reasoning continues to constrain generalisation and interpretability in composite tasks.

### 2.2 | Rule-Based Keypoints Definition

Xu et al. construct the UMD + GT affordance dataset with five unified keypoints for common household objects, encoding affordance information to enable joint visual-action modelling [27]. Manuelli et al. propose a category-level object representation using semantic 3D keypoints constrained geometrically, forming a unified pipeline from segmentation to execution and enhancing interpretability and generalisation [28]. Huang et al. generate task-oriented keypoints for deformable object manipulation by integrating task goals with object geometry, supporting adaptive grasping in flexible scenarios [29]. Tang et al. build a task-oriented grasping dataset aligning semantic keypoints with language instructions via multiview RGB-D sampling, enabling prediction of feasible grasp regions [3]. Nasiriany et al. embed keypoint generation into the vision-language reasoning process through iterative visual prompts, producing action-consistent keypoints without supervision [30]. Fang et al. extend prior definitions [16, 28] and propose a keypoint-based affordance generation

mechanism that distinguishes functional and target keypoints to represent action dependencies in multistage tasks [31]. Tang et al. further propose one-shot imitation learning based on 3D functional keypoints, achieving tool transfer via human demonstrations but limited to three coarse keypoints—grasping, centre and function [32].

Rule-based keypoint generation methods rely on manually defined geometric constraints or task priors to achieve structured modelling of specific scenarios. Although these methods offer advantages in interpretability and operational control, their expressive capacity is inherently limited by the predefined rules, making it difficult to adapt to sequential manipulation tasks in open environments. Furthermore, when handling complex physical interactions or diverse objects, rule-based approaches often exhibit limited generalisation and lack the ability to transfer across tasks or semantic contexts.

To date, keypoint representations have advanced significantly in task-driven, semantic-rich and adaptive modelling. By leveraging dataset expansion, self-supervised learning and vision-language model integration, these methods strengthen the coupling between perception and action, combining keypoints with object geometry, functional attributes and operational semantics to map visual observations to grasping and manipulation actions. Nevertheless, many approaches define only a limited number of keypoints, limiting their ability to capture fine-grained functional features in complex objects or multistage operations. In addition, the generalisation and interpretability of these representations across tools, object categories and tasks remain insufficient. Accordingly, developing a unified keypoint representation that integrates functional and geometric information is essential for advancing research on sequential robotic manipulation.

## 3 | Method

Figure 1 illustrates that the robot interacts with different types of objects in the environment. To clarify this process, we refer to the categorisation of Fang et al. [31] and classify objects into two types: object in hand $o_{in\text{-}hand}$ (e.g., a fork) and object unattached $o_{unattached}$ (e.g., a cake). $o_{in\text{-}hand}$ denotes the object that the robot needs to grasp for task execution, whereas $o_{unattached}$ denotes the object that interacts with $o_{in\text{-}hand}$ during the manipulation sequence. The execution process is organised into three stages: grasping, motion and skill execution. During the grasping stage, the robot approaches and grasps an object $o_{in\text{-}hand}$. In the motion stage, it moves the $o_{in\text{-}hand}$ into the operational space of $o_{unattached}$, satisfying the spatial constraints required for the subsequent skill. In the skill execution stage, the robot manipulates the object $o_{in\text{-}hand}$ to perform a task-specific action, thereby enabling functional interaction with the object $o_{unattached}$. In certain tasks, the robot interacts with only one type of object. For instance, when twisting off a bottle lid or opening a drawer, it acts solely on $o_{in\text{-}hand}$, whereas in tasks such as squeezing a sponge, it interacts only with $o_{unattached}$. In these cases, the corresponding stages can be omitted depending on task requirements.

In this section, we first introduce a unified keypoint representation that encodes both geometric and functional object features. Next, we describe keypoint-guided task planning that leverages the LLM to perform instruction reasoning and parsing. We then present an instruction-driven, task-oriented grasping policy. We further present spatial constraint modelling for multiobject interactions to guide the robot in moving $o_{in\text{-}hand}$ into the operational space of $o_{unattached}$. Building on this, we describe a keypoint-informed obstacle avoidance planner to ensure safe and coherent operation. Finally, we introduce a library of skill primitives driven by keypoint semantics that maps task instructions to embodied executions.

### 3.1 | Shape- and Function-Grounded Keypoint Representation

We extend the keypoint definitions of [27] and [31] to develop a unified keypoint representation that integrates object geometry and functional attributes. In terms of geometry, keypoint annotation within the same object category follows two principles: structural salience and topological consistency. Structural salience requires keypoints to reflect geometrically constrained locations that characterise the object's overall scale and configuration, such as endpoints, edge points or centroids, thereby enabling a comprehensive description of object shape and dimensions. Topological consistency is maintained by defining connectivity relationships among keypoints to preserve the integrity of the object's global geometric structure. For instance, the keypoints $p$ of a knife $o_{knife}$ include the head $p_{head}$, blade $p_{blade}$, back $p_{back}$, handle $p_{handle}$ and end $p_{end}$, whose spatial connections form a geometric skeleton that captures the object's morphology. Semantic labels of keypoints are assigned based on their local structural characteristics. Because a keypoint may correspond to multiple functional attributes, functional semantics are not explicitly associated with individual keypoints.

Functional attributes are implicitly inherent in parts that enable interaction. Semantic labels are integrated with commonsense knowledge during task planning to support function reasoning. For example, although the keypoint $p_{blade}$ is named after a structural feature, its functional role (cutting, $a_{cut}$) can be inferred from task semantics. Figure 2 illustrates the keypoints and their corresponding functional roles for several household objects, categorised into two types: active (e.g., $a_{stick}$, $a_{cut}$ and $a_{scoop}$) and passive operations (e.g., $a_{be_cut}$, $a_{be_scooped}$ and $a_{be_poured}$). These serve as functional cues for downstream intention reasoning and motion planning. In addition, we treat grasping as an essential subtask in sequential manipulation rather than a functional attribute associated with individual keypoints. To improve semantic consistency and simplify functional categorisation, we unify functional descriptions that exhibit similar motion characteristics. For example, the wiping motion of a cleaning tool on a surface and the spreading motion of a knife on food both involve reciprocating linear movement along a specific direction and are thus classified as the 'wipe'.

Based on the above definitions and annotation principles, we construct a keypoint dataset that includes 29 common household

objects across 13 categories. The dataset is available in Ref. [33] and can be directly trained with existing open-source deep learning models to enable efficient keypoint detection. In this work, we implement an SFK detection model based on the YOLO framework [34], leveraging its end-to-end learning capability for real-time keypoint localisation in RGB images. With this detector, the robot obtains keypoint information during visual perception, which serves as a knowledge foundation for sequential manipulation.

### 3.2 | Task Reasoning With the LLM

To perform task intention reasoning based on the SFK representation, we employ a pretrained LLM, denoted as $\mathcal{M}(\cdot)$. The model, trained on internet-scale corpora, possesses strong general language understanding and reasoning capabilities. As shown in Figure 3, we design a prompt template $\rho$ with semantic slots that can be dynamically filled with keypoint semantic labels. This template reformulates intention reasoning as a question–answering problem solvable by the LLM. The optional use of touchscreen, laser or voice input serves as an intent grounding interface, whereas task-level reasoning and action sequencing are performed based on the selected keypoints. The framework takes the semantic label set $P$ of keypoints and category label set $O$ as input and, combined with the commonsense-enriched prompt $\rho$ of household tasks, infers the task intentions $I(\cdot)$ implied by the selected keypoints:

$$I(A, O, P_j) = \mathcal{M}(\rho(O, P_i)) \tag{1}$$

where $I(A, O, P_j)$ denotes the structured text encompassing the semantic elements required for task execution.

The output of intention reasoning undergoes normalisation to remove natural language elements that are irrelevant to the task, retaining the target objects and action instructions. The cleaned output forms a top-down sequential representation that guides

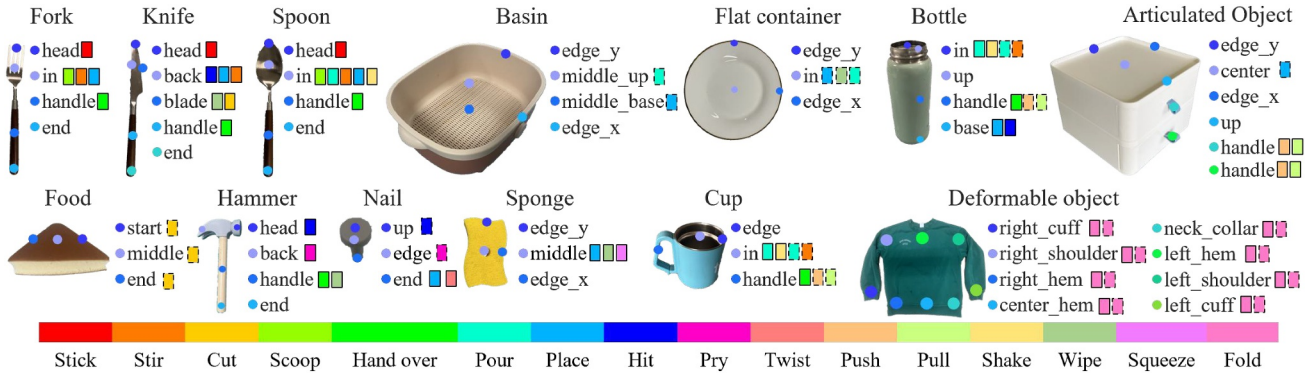



**FIGURE 2** | Keypoint representations of some household objects. Rectangles in different colours represent distinct functional attributes implied by the keypoints. Solid bounding boxes indicate active operational affordances associated with a keypoint. Dashed bounding boxes indicate passive operational affordances.

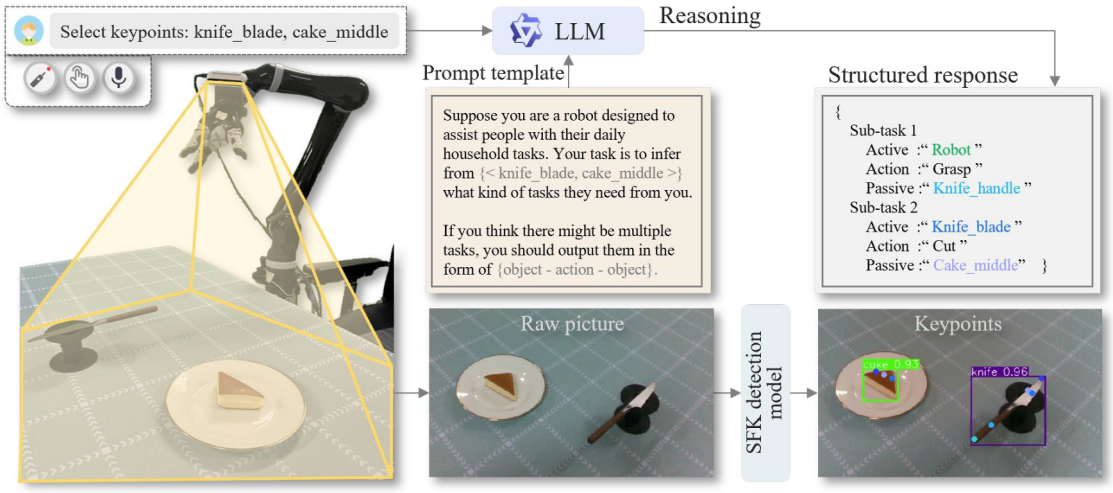



**FIGURE 3** | Task planning illustration based on keypoint semantics and the LLM. The robot continuously perceives the task scene through an RGB-D camera and performs real-time recognition of object keypoints using the SFK detection model. By selecting the semantic labels of keypoints, 'knife_blade' and 'cake_middle', these labels are embedded into a prompt template and input into the LLM to generate structured output. In the structured output, the intention sequence 'grasp knife_handle knife_blade cut cake_middle' is automatically decomposed into two subtasks, 'Robot grasp knife_handle' and 'knife_blade cut cake_middle'. Each subtask is further broken down into an active object, an action and a passive object, guiding the logical sequence of subsequent task execution.

subsequent instruction grounding for the robot. The intention sequence is decomposed into unit subtasks, each expressed as an active object, an action and a passive object triple to describe the operation relationship and to provide the structured details needed for motion synthesis. Grasp regions are matched using commonsense priors for household tasks to produce concrete grounding. For example, 'robot may grasp the knife_handle and then use the knife_blade to cut the cake_middle' is decomposed into two subtasks that specify the grasp keypoint $p_{handle}$ of the object $o_{(in\text{-}hand,\ knife)}$, the functional keypoint $p_{blade}$ of the object $o_{(in\text{-}hand,\ knife)}$, the corresponding action $a_{cut}$ and the functional keypoint $p_{middle}$ of the object $o_{(unattached,\ cake)}$.

### 3.3 | Keypoint-Prompted Task-Oriented Grasping

The task-oriented grasping module based on task instructions is built upon our previous study [35]. As illustrated in Figure 4, the module first acquires the semantic information of grasp-related keypoints from a structured instruction and localises their pixel coordinates in a 2D image. These pixel coordinates are then combined with the corresponding RGB image and input to the prompted Segment Anything Model (SAM). Prompted SAM, a zero-shot segmentation model with strong generalisation capabilities, produces precise object masks under the guidance of prompts, which can be provided in two forms, namely, points and bounding boxes. In this work, the pixel coordinate of the grasp-related keypoint is embedded into the prompt to guide the model towards the functional regions of the target object, enabling task-relevant local segmentation.

After obtaining the segmentation mask, the module extracts the mask boundaries to capture the geometry of functional regions and computes antipodal grasp points based on force-closure analysis. The corresponding point cloud of the object is then obtained from the depth image, and surface normals at the grasp points are calculated to derive the 6D grasp pose $T_{grasp}$. By integrating keypoint semantic cues with force-closure analysis, the system enforces task constraints at the perception level, enabling task-oriented grasp generation.

### 3.4 | 6D Pose Estimation Based on Keypoint Topological Constraints

To ensure that the robot can correctly execute subsequent actions in a task scenario, the pose of each object in SE(3) space is estimated in advance. The object's set of keypoints is first mapped from the 2D image to 3D space. Depth information for each keypoint is obtained from a depth image captured synchronously with the RGB image and combined with pixel coordinates to transform the keypoints into the camera coordinate system, yielding their 3D positions. The 6D pose estimations for several household objects are shown in Figure 5. To provide an intuitive visualisation, the pose coordinate frames are overlaid on the point cloud scenes captured by the camera, clearly

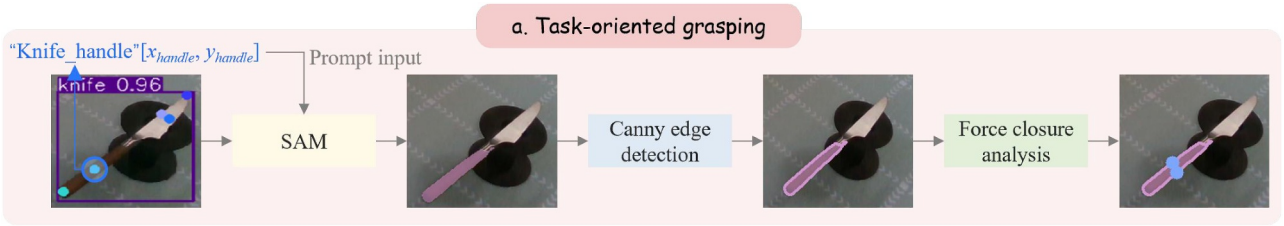



**FIGURE 4** | The pipeline of keypoint-prompted task-oriented grasping. Once the grasp-related keypoint is identified as "knife_handle", its pixel coordinate and the corresponding RGB image are provided to SAM. SAM uses the pixel coordinate as a prompt to segment the "knife_handle" and generate a mask. Antipodal grasp points on the handle are then computed via force-closure analysis, enabling task-oriented grasping.

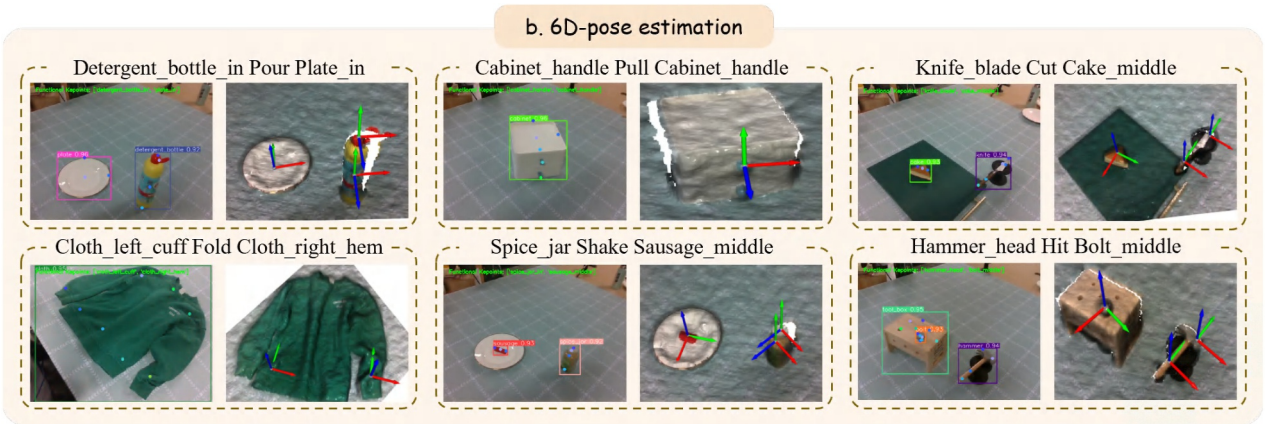



**FIGURE 5** | 6D pose estimation of objects for different tasks. When the task intention sequence is obtained, the 6D poses of task-relevant objects are estimated and located at the corresponding keypoints.

showing the task-relevant spatial positions and orientations of each object.

The 6D pose of an object is estimated following a unified scheme. The translation $t$ is defined as the 3D coordinates of the task-relevant functional keypoints. The rotation $R$ is derived from the geometric topology of the object's keypoint set. In SO(3) space, the object's $z$-axis is aligned with the upward direction perpendicular to the ground, denoted as $R_z = (r_{zx}, r_{zy}, r_{zz})$. The $y$-axis is determined by a pair of geometrically meaningful keypoints. The endpoint is defined by keypoints representing edges or extremities, such as $p_{start}$, $p_{edge_y}$ $p_{edge}$ and $p_{head}$, whereas the origin is defined by keypoints representing the centre or connection, such as $p_{middle}$, $p_{in}$ and $p_{handle}$. The initial $y$-axis direction is obtained from the 3D coordinate difference between these two keypoints. For example, for a cake, the initial $y$-axis direction $R_y^*$ is determined by the spatial difference between the keypoints $p_{start}$ and $p_{middle}$.

$$R_y^* = \begin{bmatrix} r_{yx}^* \\ r_{yy}^* \\ r_{yz}^* \end{bmatrix} = \begin{bmatrix} x_{start} \\ y_{start} \\ z_{start} \end{bmatrix} - \begin{bmatrix} x_{middle} \\ y_{middle} \\ z_{middle} \end{bmatrix} \tag{2}$$

where $r_{yx}^*$, $r_{yy}^*$ and $r_{yz}^*$ represent the projections of the $R_y^*$ onto the $x$-, $y$- and $z$-axes of the camera coordinate system, respectively. $(x_{start}, y_{start}, z_{start})$ represents the 3D coordinates of the cake's keypoint $p_{start}$ in the camera coordinate system. $p_{middle}$ is defined similarly.

The initially computed $R_y^*$ cannot be directly used as the $y$-axis of the 6D pose in SO(3) because it may not be orthogonal to the $z$-axis $R_z$ in real-world scenarios. The $x$-axis $R_x$ is instead derived from the cross product of $R_y^*$ and $R_z$, ensuring a valid orthogonal basis.

$$R_x = R_y^* \times R_z \tag{3}$$

where $\times$ denotes the cross product between two vectors. The $x$-axis $R_x$ is defined to be perpendicular to the plane formed by $R_y^*$ and $R_z$.

Then, the $x$-axis $R_x$ and $z$-axis $R_z$ in SO(3) are determined. Based on the right-hand rule, the final $y$-axis $R_y$ can be computed from the cross product of $R_z$ and $R_x$, defining the object's rotation $R$. $R_y$ is expressed as follows:

$$R_y = R_z \times R_x \tag{4}$$

On the basis of Equations (2–4), the 6D pose $T$ in the SE(3) space is expressed as follows:

$$T = \begin{bmatrix} R_x & R_y & R_z & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

### 3.5 | Task-Related Spatial Relationship Constraints

Spatial constraints are defined as the relative spatial relationships between the object in hand and the object unattached during task execution, which constrain the possible spatial locations the object in hand can occupy to facilitate successful action. Unlike continuous spatial constraints defined along a path trajectory, the constraints in this work focus on task-specific local spatial relationships. These constraints are constructed from commonsense priors derived from household tasks. For instance, in a pouring task, the cup must first be positioned directly above the centre of the bowl. Similarly, in a hammering task, the hammer should initially be placed above the centre of the nail. Such commonsense priors define the target 6D poses of the objects during task execution.

Because of the invariance of spatial constraints under the SE(3) group, their definition can directly rely on the known 6D poses of objects in the task scene, ensuring pose consistency across different coordinate transformations. To determine the optimal spatial constraints before skill execution, we fix the positions of two objects in the task scene and compute their relative 6D pose relationship, which is then used to derive the spatial constraints.

Taking the spatial constraint $T_{\text{cut}}$ of the action 'Cut' in Figure 1 as an example, the end-effector pose ${}^2T_{\text{ee}}$ of the robot is guided by this spatial constraint and can be expressed as follows:

$${}^2T_{ee} = T_{middle} T_{cut}{}^{handle}T_{blade}{}^{grasp}T_{handle} \tag{6}$$

where $T \in \text{SE}(3)$ and is expressed in the camera coordinate system. $T_{middle}$ denotes the 6D pose at the keypoint 'cake_middle', whereas ${}^{handle}T_{blade}$ denotes the 6D pose transformation from $T_{blade}$ to $T_{handle}$. This can be computed using the initial pose estimation of ${}^1T_{blade}$ and ${}^1T_{handle}$:

$${}^{handle}T_{blade} = \left({}^1T_{blade}\right)^{-1}{}^1T_{handle} \tag{7}$$

where $T^{-1}$ denotes the inverse of a transformation matrix in SE(3).

${}^{grasp}T_{handle}$ represents the coordinate transform from the $T_{grasp}$ to $T_{handle}$ and can be computed as follows:

$${}^{grasp}T_{handle} = \left({}^1T_{handle}\right)^{-1}{}^1T_{grasp} \tag{8}$$

where $T_{grasp}$ equals to ${}^1T_{ee}$.

For the implicit functions depicted in Figure 2, we derive sixteen spatial relational constraints, $T$. Each function's constraint is computed from the functionally relevant interobject keypoints and their 6D poses using the formulas above. Figure 6 illustrates a portion of the spatial constraints.

### 3.6 | Keypoint-Based Obstacle Avoidance Strategy

After the target pose of the object in hand is determined through the spatial constraints, we generate motion-stage trajectories using the artificial potential field (APF) method [36]. Traditional centroid-based obstacle avoidance methods struggle to account for the object's shape, whereas approaches based on CAD models or dense point clouds incur high computational costs. To balance shape integrity and computational efficiency,

we propose a keypoint-based obstacle avoidance strategy that enforces geometric abstraction derived from object keypoints, as shown in Figure 7.

Specifically, images containing object keypoints along with instruction information are first input into the SAM model, which identifies the target object under the guidance of the instruction semantics. Task-unrelated objects are treated as obstacles, and SAM automatically generates corresponding image masks. Combined with depth information, these masks are used to construct point cloud representations of the obstacles. The system then iterates over the keypoints of the hand-held object, treating each as a point mass, and computes an obstacle-avoiding trajectory using an artificial potential field. After trajectory generation, the system further checks whether the remaining keypoints collide with obstacles. This check is performed by computing the minimum distance between trajectory points and the obstacle point cloud using a K-nearest neighbours (KNN) algorithm. When none of the keypoints overlap with obstacles, the trajectory is considered a valid motion path that satisfies the obstacle avoidance constraints.

### 3.7 | Skill Primitive Construction

Once the object in hand reaches its target pose, the robot executes the corresponding action skill. By analysing the characteristics of each task skill, we extract the minimal units of motion to construct skill primitives. Although these primitives are simplified in space and time, they retain the core kinematic features of the actions. Building on our previously developed skill library [35], we further extend it to create a library of skill primitives to support the execution of diverse tasks. The generated primitive trajectories can be integrated with task-oriented grasping and motion planning modules to form a complete sequential manipulation pipeline. Sixteen skill primitives are constructed, and examples of some primitive trajectories are shown in Figure 8.

## 4 | Experiment

In this section, the proposed SFK-based framework is evaluated across perception, task-level reasoning and execution. Different keypoint generation methods are first compared, followed by a robustness evaluation of the proposed SFK representation under out-of-distribution and occlusion conditions. Task-level action prediction is then analysed through comparisons with representative reasoning approaches. Finally, grasp generation comparisons and real-world task-level manipulation evaluations are conducted to assess the effectiveness of the proposed framework across diverse manipulation scenarios.

Our experimental platform is built upon a Kinova Jaco 6-DoF robotic arm. For perception, we employ an Intel RealSense D435 RGB-D camera mounted on the robot arm to provide a first-person view of the manipulation workspace. The system runs on a workstation equipped with an Intel i7-12700F CPU and an NVIDIA RTX 3060 GPU, with GPU acceleration enabled via CUDA 11.7. For high-level task planning, we use the online API of the Qwen Plus 0728 large language model. On the execution side, the robot is equipped with the Kinova Gen3 three-finger gripper for manipulation.

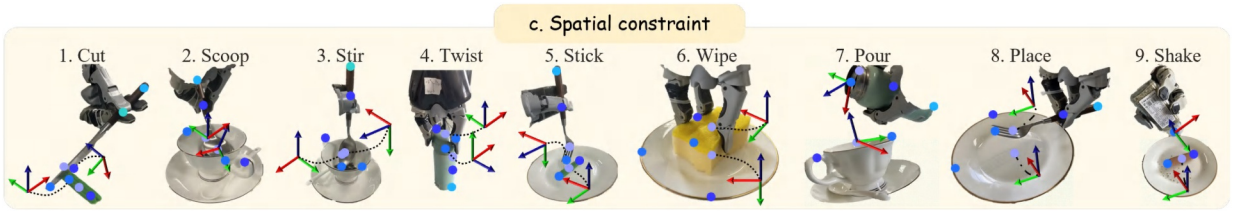



**FIGURE 6** | Illustration of task-related spatial relationship constraints.

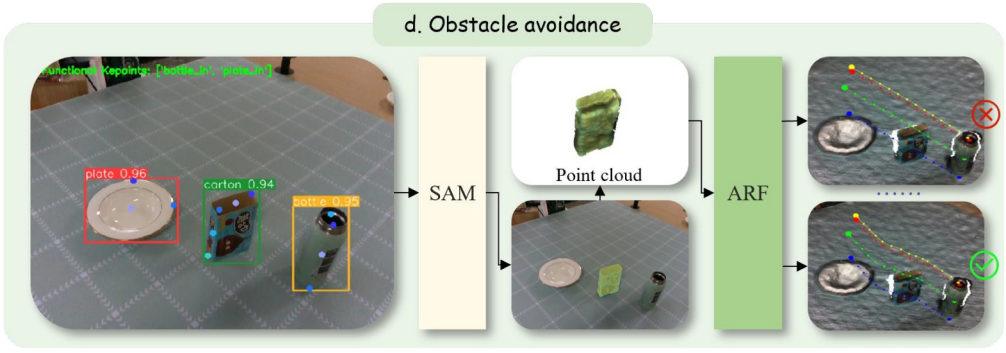



**FIGURE 7** | Pipeline of the keypoint-based obstacle avoidance strategy. In the task 'grasp bottle_handle bottle_in pour plate_in', a carton along the bottle's motion path is identified as a task-irrelevant object (obstacle). Its bounding box (green) and RGB image are input to SAM to generate an obstacle mask, which is fused with depth data to obtain the obstacle point cloud. The point cloud and the bottle's keypoint coordinates are then fed into APF for trajectory planning, producing a feasible path that avoids collision.

### 4.1 | Keypoint Generation and Robustness Evaluation

First, a qualitative comparison is conducted between SFK and three representative keypoint generation methods, including DINOv2 (self-supervised pretraining) [37], CLIP (vision-language contrastive pretraining) [38] and ViT (supervised pretraining) [39]. For DINOv2, the candidate keypoints per mask, spatial coordinate weight and minimum keypoint distance are set to 8, 1 and 22, respectively. For CLIP, keypoints are directly generated based on visual feature representations. For ViT, the number of keypoints per mask, the PCA-reduced feature dimensionality and the minimum spatial distance are set as 6, 16 and 18, respectively. The detailed parameter settings are presented in Supporting Information S1: Figures S1–S3. For all baseline methods, an object prior is required to constrain the feature extraction region using a segmentation algorithm, for which SAM is adopted following a ReKep-style pipeline. The extracted feature maps are then clustered, and the resulting cluster centres are projected into the RGB image space as keypoint representations. Figure 9 presents qualitative results across four different scenes.

#### *4.1.1 | Comparisons of Different Methods for Keypoint Generation*

As shown in Figure 9, SFK generates keypoints without an explicit object prior, whereas the baseline methods rely on SAM-based segmentation. Although the object prior facilitates target localisation, it introduces additional computational overhead. The average inference time per image is 9.116 s for

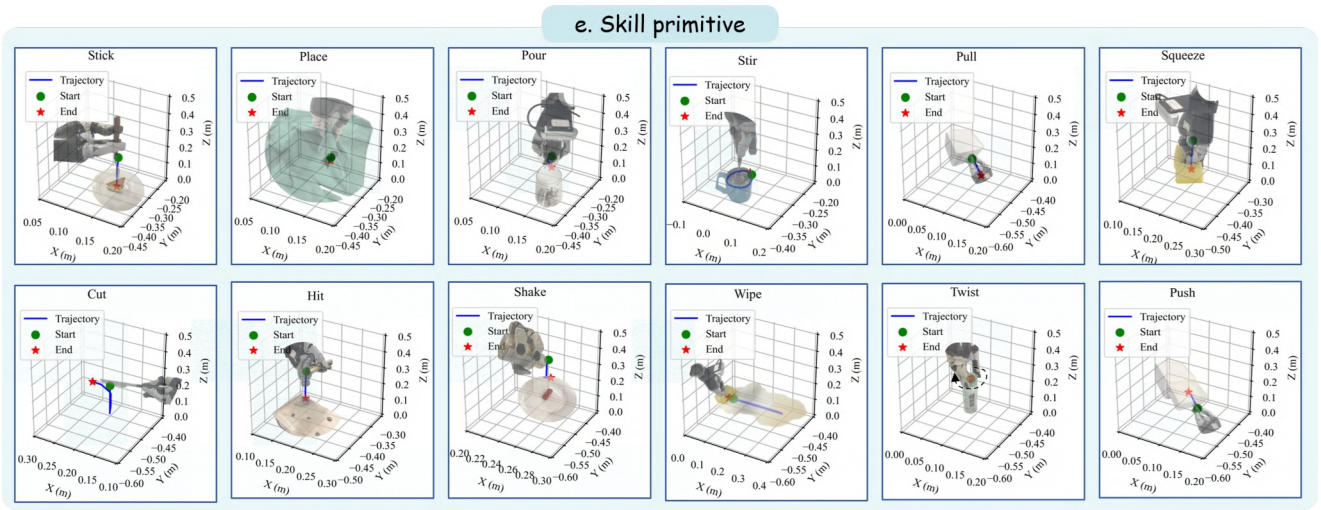



**FIGURE 8** | Skill primitive library.

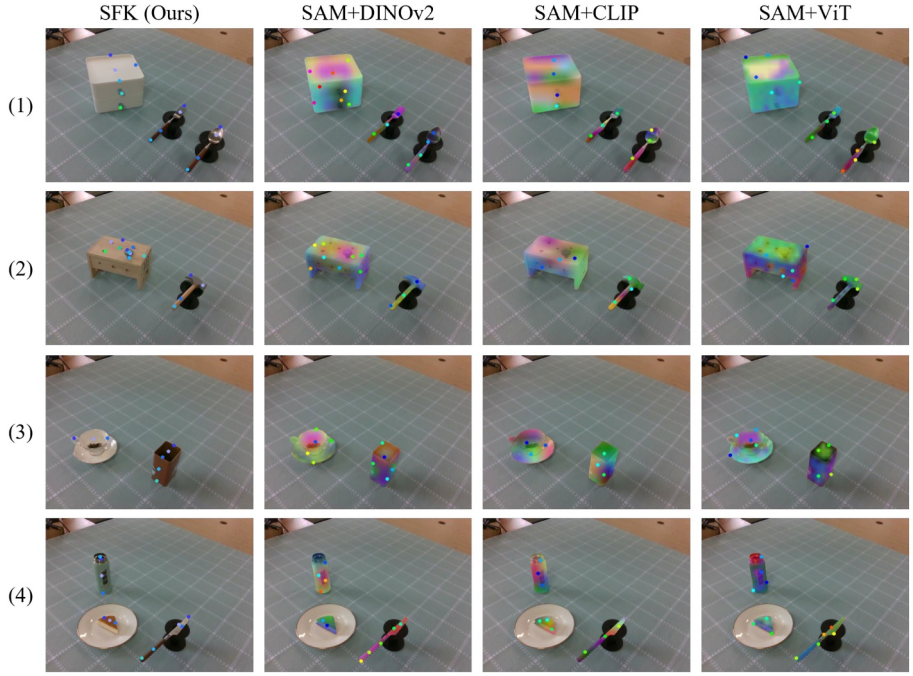



**FIGURE 9** | Comparisons of different methods for keypoint generation.

SAM + DINOv2, 5.716 s for SAM + CLIP, 7.211 s for SAM + ViT and 0.007 s for SFK, resulting in the lowest inference latency among the compared methods.

Although all methods can generate keypoints that provide visual guidance for manipulation, SFK tends to capture functionally relevant object regions. Across different scenes, SFK produces semantically meaningful keypoints aligned with manipulation affordances, such as tool interaction regions and graspable parts, supporting multiobject interactions. In contrast, DINOv2 and ViT mainly respond to fine-grained geometric structures and may focus less on task-relevant regions such as cabinet handles, nail tips and toolbox openings, which may limit their applicability to manipulation. CLIP can localise functional regions in certain cases, but its performance is less stable across multiobject scenarios. For elongated tools (e.g., hammers, knives, forks and spoons), baseline methods tend to reflect overall structure or coarse functional partitions, whereas SFK more consistently captures task-critical functional regions (e.g., knife blades or hammer heads). These differences directly affect the feasibility and stability of keypoint-driven robotic execution. Overall, compared with representative baselines, SFK generates manipulation-oriented keypoints with higher functional relevance and lower time.

#### 4.1.2 | *Generalisation Experiments Under Out-of-Distribution Conditions*

To further validate the generalisation capability and keypoint consistency of the proposed SFK method under out-of-distribution (OOD) conditions, we conduct a systematic evaluation of keypoint generation performance along three dimensions, including background variation, instance-level differences and lighting changes. The corresponding figures are provided in Supporting Information S1: Figure S4. We first assess the effect of background variation using object categories within the annotated distribution by constructing four visually distinct background textures, including a pink tablecloth, a wooden surface, a white smooth surface and a green fabric surface. Next, we evaluate instance-level generalisation by introducing OOD object instances that differ from the annotated instances in terms of texture, scale and geometric structure to examine the robustness and consistency of the generated keypoints. Finally, we conduct experiments under four representative lighting conditions, including natural lighting, artificial lighting, low natural lighting and extremely low artificial lighting, to analyse the impact of lighting intensity variations on keypoint generation performance.

We select 10 object categories for evaluation, including forks, spoons, knives, cans, plates, bottles, hammers, spice jars, cabinets and cups. For instance-level generalisation, two OOD instances are introduced for each category. For all experiments, keypoint generation is performed three times, and we report the mean and standard deviation (std. dev.) of both class recognition confidence and keypoint detection confidence, with the quantitative results summarised in Table 1.

As shown in Table 1, the mean of category recognition confidence remains largely stable under OOD background and lighting conditions, decreasing only slightly from 0.923 to 0.917 and 0.912, respectively, with consistently low standard deviations. In contrast, instance-level OOD variations cause a more pronounced degradation, with the mean confidence dropping to 0.652 and the standard deviation increasing to 0.263. This degradation is closely related to morphological differences across object instances; objects with strong function-driven shape similarity (e.g., forks, spoons, knives and cans) are less affected, whereas objects with more diverse appearances exhibit lower confidence and higher variance. Keypoint detection confidence shows smaller variation across all OOD settings, with the mean remaining above 0.8 and a low standard deviation. This stability arises because keypoint generation is performed after category recognition and relies on ROI-level features, making it less sensitive to background, lighting and instance variations. Overall, the results in Table 1 indicate that the proposed SFK maintains stable keypoint generation under background and lighting shifts while exhibiting a reasonable degree of robustness under instance-level variation.

#### 4.1.3 | *Keypoint Generation Robustness Experiments Under Occlusion Conditions*

To evaluate the robustness of the proposed SFK method under occlusion, we conduct an assessment under two representative occlusion settings, namely, random occlusion and manual occlusion. In this study, occlusion is not explicitly annotated in the dataset. Random occlusion is implemented by randomly inserting occluding blocks within the object region, with occlusion ratios set to 10%, 20%, 30% and 40%, to emulate partial occlusions in realistic environments. Manual occlusion is applied by masking one to two keypoints to examine the behaviour of SFK when functionally relevant regions are obstructed. The representative qualitative examples are provided in Supporting Information S1: Figure S5.

Quantitative results for 10 object categories are reported in Table 2. Under random occlusion, as the occlusion ratio

**TABLE 1** | Class and keypoint confidence under in-distribution and out-of-distribution conditions.

| | | Out-of-distribution | | |
|---|---|---|---|---|
| **Metric** | **In-distribution** | **Background** | **Instances** | **Lighting** |
| Class confidence (mean) | 0.923 | 0.917 | 0.652 | 0.912 |
| Class confidence (std. dev.) | 0.012 | 0.009 | 0.263 | 0.010 |
| Keypoint confidence (mean) | 0.996 | 0.992 | 0.856 | 0.992 |
| Keypoint confidence (std. dev.) | 0.002 | 0.003 | 0.028 | 0.002 |

increases from 10% to 40%, the mean category recognition confidence decreases from 0.901 to 0.449, with standard deviation increasing from 0.011 to 0.278 and recognition precision dropping from 0.959 to 0.432. In contrast, the mean keypoint detection confidence decreases more gradually, from 0.994 to 0.816. Under manual occlusion, category recognition confidence drops to 0.632, whereas keypoint detection confidence remains at 0.917, indicating that keypoints can still be generated even when partial discriminative characteristics are occluded.

These results indicate that occlusion affects category recognition and keypoint generation; however, reasonable keypoint predictions can still be obtained under partial and substantial occlusion of critical functional regions.

## 4.2 | Task-Level Action Prediction Evaluation

In task planning, we output manipulation intent as a probability distribution over candidate actions for different objects, conditioned on the perceived scene. This formulation reflects the system's understanding of object affordances and task semantics. Based on this formulation, we further conduct experiments to analyse the performance of different reasoning methods in task-level action prediction. Specifically, we select two representative reasoning paradigms for comparison: an LLM-based approach using GPT-4 and a probabilistic graphical model based on conditional random fields (CRFs). For consistency, Qwen and GPT-4 predict action intents using the same prompt template, whereas the CRF model is constructed based on the examples provided in the prompts. The prompt templates and corresponding examples are shown in Supporting Information S1: Figures S6 and S7.

To analyse the behaviours of different reasoning methods under different manipulation settings, we compare their action prediction results under in-distribution and out-of-distribution object combinations. In this setting, in-distribution refers to manipulation scenarios explicitly covered by the prompt templates and examples, whereas out-of-distribution scenarios do not appear in the prompts or examples. Figure 10 illustrates the action probability distributions of different methods under two settings. Under in-distribution conditions, large language model-based reasoning methods tend to assign higher probabilities to actions consistent with object functional properties and task semantics while effectively suppressing less task-

**TABLE 2** | Quantitative evaluation of class recognition and keypoint generation under different occlusion conditions.

| | Random block occlusion | | | | Manual occlusion |
|---|---|---|---|---|---|
| **Metric** | **10%** | **20%** | **30%** | **40%** | **Hand** |
| Class confidence (mean) | 0.901 | 0.837 | 0.814 | 0.449 | 0.632 |
| Class confidence (std. dev.) | 0.011 | 0.121 | 0.191 | 0.278 | 0.074 |
| Class precision | 0.959 | 0.920 | 0.857 | 0.432 | 0.903 |
| Keypoint confidence (mean) | 0.994 | 0.921 | 0.917 | 0.816 | 0.917 |
| Keypoint confidence (std. dev.) | 0.009 | 0.014 | 0.035 | 0.093 | 0.181 |

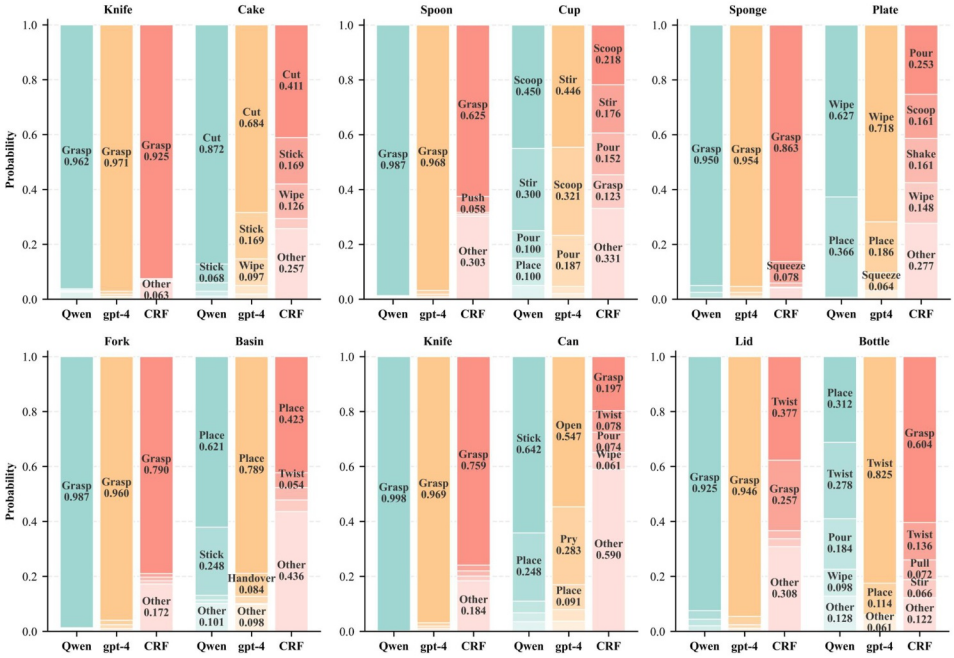



**FIGURE 10** | Comparison of action probability distributions across object-centric manipulation scenarios. Top row: action probability distributions for in-distribution tasks. Bottom row: action probability distributions for out-of-distribution tasks.

relevant candidates. In contrast, the CRF-based method exhibits more dispersed probability distributions across object categories, assigning relatively higher probability to suboptimal or less task-relevant actions.

Under out-of-distribution conditions, the action probability distributions predicted by Qwen and GPT-4 remain relatively consistent and semantically coherent across different objects, whereas the CRF-based method is more prone to inference confusion. These results suggest that large language model-based reasoning provides stronger semantic modelling at the task level, enabling more reliable high-level guidance for subsequent motion planning. Considering reasoning performance, resource overhead and system-level deployability, we adopt Qwen as the task intent planning model in our system. In addition, we evaluated the consistency of the large language model by repeatedly providing identical input labels and analysing the generated outputs. For six groups of input labels, each repeated 10 times, the results indicate that the executability rate of the generated outputs reaches approximately 91.7%, whereas the average output consistency is about 58.33%. The corresponding raw data are provided in the supplementary materials.

## 4.3 | Task-Oriented Grasping Experiment

To evaluate the effectiveness of the proposed keypoint representation for task-oriented grasp generation, we conduct experiments across multiple object categories and manipulation tasks. The principle of task-oriented keypoint extraction is illustrated in Figure 4, where grasp orientations are determined in the SO(3) space of the estimated 6D object pose (Section 3.4). Representative results are shown in Figure 11, with grasp generation without keypoint guidance used as the baseline. The results demonstrate that keypoint-guided grasping enables the robot to select grasps aligned with object functional regions for different tasks. For example, in cutting tasks, grasps are generated on the handle while avoiding the blade, whereas in handover tasks, commonsense priors lead to grasping the blade for safe transfer.

To further compare grasping performance, we select AnyGrasp and GraspNet as baselines for qualitative comparison. To ensure consistent keypoint guidance, both methods generate candidate grasp poses on the target object, which are then filtered using a task-relevant grasp prompt to select the most spatially consistent grasp.

As shown in Figure 12, the grasp poses produced by different methods exhibit noticeable differences. AnyGrasp and GraspNet primarily generate grasp candidates based on geometric feasibility, leading to a large number of grasp poses distributed over the object surface. Although this provides broad geometric coverage, the lack of prior modelling of object functional structure and manipulation semantics often results in spatially dispersed and redundant grasp poses.

After introducing grasp prompt constraints, the grasp poses produced by the baseline methods can be guided towards task-relevant regions to a limited extent. In contrast, the grasping approach adopted in this work directly generates grasp poses aligned with object functional regions, such as handles, edges or operable surfaces, resulting in clearer grasp localisation. This function-oriented keypoint representation preserves functional consistency across different object instances, thereby providing structured, task-relevant guidance for grasp planning.

Taken together, the qualitative comparisons indicate that our grasp method tends to generate grasp poses that are more consistently aligned with object functional regions, potentially improving the suitability of the resulting grasps for downstream manipulation.

## 4.4 | Obstacle Avoidance Experiment

To evaluate the impact of geometric representation on obstacle-aware trajectory planning, we consider scenarios with one, two and five obstacles along the task path. The experiments involve four geometric representations, including centroid-based,




**FIGURE 11** | The result of task-oriented antipodal grasp points generation. Row (a) shows keypoint detection for different objects. Row (b) shows antipodal grasp points generated without keypoint guidance. Rows (c) and (d) show antipodal grasp points generated under keypoint guidance for different tasks.

bounding box-based, point cloud-based and SFK-based representations, under an artificial potential field (APF) planner and a rapidly exploring random tree (RRT) planner. To improve continuity, trajectories are postprocessed using a Savitzky–Golay filter.

Figure 13 illustrates trajectories generated under different geometric representations and obstacle configurations, with the corresponding planning time summarised in Table 3. Centroid-based planning does not account for the object extent, which would lead to collisions under tight obstacle configurations. Point cloud-based representations are observed to remain collision-free in the evaluated scenarios, with planning time increasing as obstacle complexity grows. Bounding box-based representations approximate object geometry using coarse spatial envelopes and incur higher planning time under both APF and RRT planners. The SFK-based representation captures object geometry using distributed keypoints and incurs lower planning time than bounding box-based and point cloud-based representations, supporting its use in our framework for balancing geometric awareness and planning efficiency.

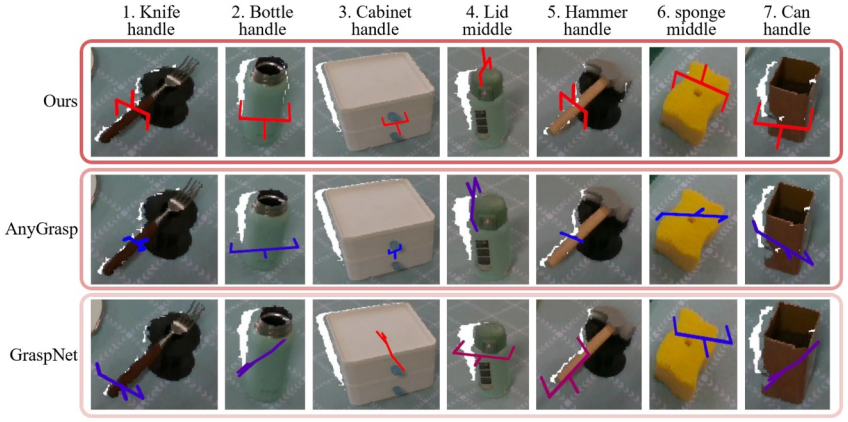



**FIGURE 12** | Representative qualitative examples under different grasp generation methods.

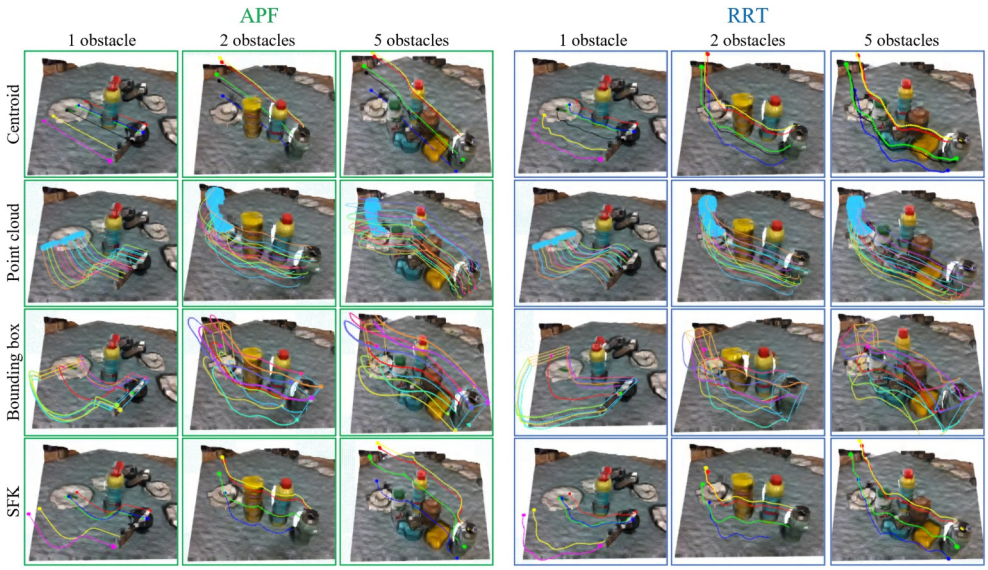



**FIGURE 13** | Comparison of obstacle avoidance trajectories using different geometric representations and planning methods.

**TABLE 3** | Planning time (s) of different geometric representations under APF and RRT planners.

| | APF | | | RRT | | |
|---|---|---|---|---|---|---|
| **Representation** | **1 obstacle** | **2 obstacles** | **5 obstacles** | **1 obstacle** | **2 obstacles** | **5 obstacles** |
| Centroid | 0.019 | 0.022 | 0.036 | 0.559 | 0.717 | 0.750 |
| Point cloud | 0.244 | 0.591 | 1.223 | 0.826 | 0.908 | 3.107 |
| Bounding box | 0.433 | 1.133 | 3.457 | 2.942 | 7.197 | 19.525 |
| SFK | 0.019 | 0.056 | 0.079 | 0.581 | 0.761 | 0.904 |

## 4.5 | Task-Level Manipulation Evaluation in Real-World Scenarios

To evaluate the feasibility of the proposed framework, we conduct real-world sequential manipulation experiments. A total of 22 tasks across six household object categories are evaluated, achieving an overall mean success rate of 78.01%, suggesting that the proposed framework can be applied in real-world environments. Detailed data are provided in Supporting Information S1: Figure S11. To further examine its effectiveness, we conduct comparative evaluations with representative baseline methods and perform module-level ablation experiments to analyse the contributions of individual system components.

### *4.5.1 | Comparative Evaluation of Task Execution*

We select 10 representative manipulation tasks. As shown in Table 4, these tasks involve common manipulation scenarios, including interactions between rigid objects (e.g., pouring and insertion) and operations involving articulated components or continuous state changes (e.g., opening drawers, twisting bottle caps and wiping plates). The tasks range from basic pick-and-place to manipulation with directional constraints and articulated objects, allowing us to assess the system's task execution performance in realistic environments.

For each task, we conduct 10 independent trials, with the object layout reconfigured after each trial. We select ReKep as the baseline method for comparison [10]. ReKep employs SAM and DINOv2 for keypoint extraction and AnyGrasp for grasp pose generation and decomposes complex tasks into multiple stages using a hierarchical optimisation framework. This framework encodes task constraints and generates trajectories through geometric relationships among keypoints, with the $z$-axis predefined to point in the antigravity direction. Representative task trajectories produced by the hierarchical optimisation are shown in Supporting Information S1: Figure S13.

**TABLE 4** | Quantitative results across 10 real-world manipulation tasks.

| Task | Keypoints | ReKep | SFK |
|---|---|---|---|
| Pour tea | "bottle_in", "cup_in" | 3/10 | 8/10 |
| Fit the lid onto the bottle | "lid_middle", "bottle_in" | 2/10 | 7/10 |
| Place the kitchenware into the basin | "knife_back", "basin_middle_in" | 7/10 | 9/10 |
| Insert the knife into the can | "knife_head", "can_in" | 3/10 | 6/10 |
| Lift the cake from the plate | "cake_middle", "plate_in" | 9/10 | 9/10 |
| Hammer the nail | "hammer_head", "nail_middle" | 2/10 | 7/10 |
| **Total** | | **43.3%** | **76.7%** |
| Open the drawer | "cabinet_handle" | — | 7/10 |
| Close the drawer | "cabinet_handle" | — | 8/10 |
| Twist the lid of the bottle | "lid_middle", "bottle_in" | — | 4/10 |
| Use the sponge to wipe the plate | "sponge_middle", "plate_in" | — | 9/10 |
| **Total** | | — | **67.5%** |

*Note:* Bold values indicate the average task success rate computed from the above data.

Table 4 reports the results of the two methods across different tasks. For tasks executable by both methods, SFK achieves a higher success rate (76.7%) than ReKep (43.3%). For tasks involving articulated components or continuous state changes, ReKep is unable to execute the tasks due to its underlying principle. These results indicate that the proposed method achieves higher task success rates and broader task coverage.

### *4.5.2 | Module-Level Ablation Experiment*

To further analyse the factors underlying the performance differences, we conduct module-level ablation experiments by individually replacing the keypoint detection, trajectory planning and grasping modules with their corresponding baselines and comparing task success rates and execution efficiency across different module configurations, as summarised in Table 5.

From Table 5, replacing the spatial constraint-based planner with hierarchical optimisation leads to performance degradation in tasks requiring strict end-effector orientation control. For example, in cake cutting and hammering tasks, which require precise directional control, the hierarchical optimisation-based planner fails to adequately constrain the orientations of the knife blade and hammer head, resulting in task failure. These results indicate that hierarchical optimisation methods relying primarily on geometric keypoint constraints are limited when applied to orientation-sensitive manipulation tasks.

When the keypoint generation module is replaced with SAM + DINOv2 or SAM + CLIP, the system maintains relatively high success rates on rigid-object manipulation tasks. However, for tasks involving directional constraints or articulated structures, such as hammering and drawer opening, performance degrades noticeably and execution time increases. These results indicate that keypoint representations lacking explicit functional semantic constraints are limited in supporting such manipulation tasks.

When the grasping module is replaced with AnyGrasp, the generated grasp poses exhibit limited consistency in both position and orientation, even after filtering based on key functional areas. This reduces execution stability during subsequent manipulation and is reflected in degraded overall task performance.

The ablation results suggest that combining SFK with spatially constrained planning and task-oriented grasping supports both common rigid-object manipulation and tasks requiring explicit orientation awareness. This combination contributes to improved success rates and execution efficiency in real-world scenarios.

**TABLE 5** | Module-level ablation results on manipulation task performance.

| ID | Keypoint detection | Trajectory planning | Grasping method | Place the tools | Open the drawer | Hammer the nail | Cut the cake |
|---|---|---|---|---|---|---|---|
| 1 | SFK (ours) | Spatial constraint (ours) | Task-oriented grasping (ours) | 9/10 | 7/10 | 7/10 | 8/10 |
| | 0.007 s | 0.003 s | 3.281 s | 22.425 s | 17.687 s | 22.831 s | 26.324 s |
| 2 | Ours | Optimisation-based | Ours | 9/10 | 6/10 | — | — |
| | 0.008 s | 0.084 s | 3.297 s | 23.386 s | 18.296 s | — | — |
| 3 | SAM + DINOv2 | Optimisation-based | Ours | 9/10 | — | — | — |
| | 9.116 s | 0.087 s | 3.393 s | 35.976 s | — | — | — |
| 4 | SAM + CLIP | Optimisation-based | Ours | 8/10 | 3/10 | 1/10 | — |
| | 5.716 s | 0.079 s | 3.411 s | 28.471 s | 23.131 s | 29.752 s | — |
| 5 | Ours | Ours | AnyGrasp | 8/10 | 4/10 | 4/10 | 3/10 |
| | 0.007 s | 0.003 s | 4.324 s | 23.597 s | 18.405 s | 23.971 s | 31.093 s |

## 5 | Conclusions

This paper presents a shape- and function-grounded keypoint (SFK) representation and a unified keypoint-driven framework for sequential manipulation. SFK enables real-time keypoint inference (0.007 s per image), faster than SAM + DINOv2, SAM + CLIP and SAM + ViT. Under out-of-distribution settings, the mean of keypoint confidence remains above 0.85, and under 40% occlusion, the mean remains 0.816, demonstrating robustness. In real-world experiments, the proposed system achieves a task success rate of 76.7%, outperforming ReKep (43.3%), while additionally supporting articulated and continuous-state tasks with an average success rate of 67.5%. In obstacle-aware planning, SFK-based geometric abstraction reduces planning time compared with bounding box and point cloud representations while maintaining collision-free trajectories. These results indicate that SFK provides an efficient and reliable intermediate representation for integrating perception, reasoning and execution in sequential robotic manipulation.

## 6 | Limitations and Future Work

Although the robustness of keypoint generation has been validated under moderate occlusion and out-of-distribution conditions, missing or unstable keypoints would reduce the accuracy of SE(3) pose estimation and consequently affect downstream planning and execution, as shown in Supporting Information S1: Figure S9. In addition, the sparse keypoint abstraction used in this work focuses on capturing task-relevant functional regions rather than representing complete object geometry, which may introduce approximation limitations when dealing with elongated or irregularly shaped objects. Furthermore, the current framework mainly targets household manipulation tasks composed of two to three consecutive actions, and its applicability to more complex or long-horizon task sequences has not yet been systematically evaluated.

In future work, we plan to extend the proposed framework along several directions to further enhance its robustness, autonomy and scalability. First, the robustness of 6D pose estimation under severe occlusion will be improved through uncertainty-aware and temporally consistent keypoint modelling, including confidence-weighted pose estimation, temporal smoothing and multiview keypoint fusion. Second, more comprehensive geometric modelling will be explored by integrating keypoint topology with object shape reconstruction or full-scene perception, enabling more reliable collision handling across a broader range of household objects. Third, the currently manually abstracted spatial constraints will be extended towards a more general constraint learning framework, where manipulation constraints can be incrementally acquired from common-sense sources such as instructional videos or manuals while preserving interpretability. In addition, the framework will be further extended towards learning-based manipulation systems in which grasping and manipulation policies are learnt directly from data. In this direction, recent studies such as RTAGrasp [40] provide valuable references, and we plan to conduct more systematic comparisons as the framework evolves towards a more data-driven manipulation architecture. Finally, the framework will be expanded to support long-horizon and large-scale household manipulation tasks by introducing hierarchical task monitoring, intermediate state verification and recovery mechanisms, enabling systematic evaluation in more complex and dynamic real-world environments.


### Funding

This research was supported by the National Key Research and Development Program of China (Grant No. 2024YFB4709400) and Supported by the Taishan Industrial Experts Program.


### Conflicts of Interest

The authors declare no conflicts of interest.

### Data Availability Statement

The data that support the findings of this study are openly available in Roboflow Universe at https://universe.roboflow.com/harbin-institute-of-technology-9q8vx/feed_keypoints-2-v1frv.

## References


1. R. Pfeifer and F. Iida, Embodied Artificial Intelligence: Trends and Challenges, (2004), 1–26.

2. E. Triantafyllidis, F. Acero, Z. Liu, and Z. Li, "Hybrid Hierarchical Learning for Solving Complex Sequential Tasks Using the Robotic Manipulation Network ROMAN," *Nature Machine Intelligence* 5, no. 9 (2023): 991–1005, https://doi.org/10.1038/s42256-023-00709-2.

3. C. Tang, D. Huang, L. Meng, W. Liu, and H. Zhang, "Task-Oriented Grasp Prediction With Visual-Language Inputs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2023), 4881–4888.

4. H. Liang, X. Ma, S. Li, et al., "PointNetGPD: Detecting Grasp Configurations From Point Sets," in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019)

5. Z. Lu, N. Wang, and C. Yang, "A Constrained DMPs Framework for Robot Skills Learning and Generalization From Human Demonstrations," *IEEE/ASME Transactions on Mechatronics* 26, no. 6 (2021): 3265–3275, https://doi.org/10.1109/tmech.2021.3057022.

6. M. Guo and M. Burger, "Geometric Task Networks: Learning Efficient and Explainable Skill Coordination for Object Manipulation," *IEEE Transactions on Robotics* 38, no. 3 (2022): 1723–1734, https://doi.org/10.1109/tro.2021.3111481.

7. P. Englert and M. Toussaint, "Learning Manipulation Skills From a Single Demonstration," *International Journal of Robotics Research* 37, no. 1 (2018): 137–154, https://doi.org/10.1177/0278364917743795.

8. P. F. Lucas Manuelli and R. Tedrake, "Keypoints Into the Future: Self-Supervised Correspondence in Model-Based Reinforcement Learning," in *Conference on Robot Learning* (PMLR, 2020), 693–710.

9. P. M. Scheikl, E. Tagliabue, B. Gyenes, et al., "Sim-To-Real Transfer for Visual Reinforcement Learning of Deformable Object Manipulation for Robot-Assisted Surgery," *IEEE Robotics and Automation Letters* 8, no. 2 (2023): 560–567, https://doi.org/10.1109/lra.2022.3227873.

10. W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "ReKep: Spatio-Temporal Reasoning of Relational Keypoint Constraints for Robotic Manipulation," preprint arXiv:2409.01652 (2024).

11. C. Tang, J. Yu, W. Chen, B. Xia, and H. Zhang, "Relationship Oriented Semantic Scene Understanding for Daily Manipulation Tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2022), 9926–9933.

12. M. Zaccaria, F. Manhardt, Y. Di, F. Tombari, J. Aleotti, and M. Giorgini, "Self-Supervised Category-Level 6D Object Pose Estimation With Optical Flow Consistency," *IEEE Robotics and Automation Letters* 8, no. 5 (2023): 2510–2517, https://doi.org/10.1109/lra.2023.3254463.

13. E. P. Örnek, A. K. Krishnan, S. Gayaka, et al., "SupeRGB-D: Zero-Shot Instance Segmentation in Cluttered Indoor Environments," *IEEE Robotics and Automation Letters* 8, no. 6 (2023): 3709–3716, https://doi.org/10.1109/lra.2023.3271527.

14. A. Delitzas, A. Takmaz, F. Tombari, R. Sumner, M. Pollefeys, and F. Engelmann, "SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), 14531–14542.

15. H. G. Cao, W. Zeng, and I. C. Wu, "Learning Sim-to-Real Dense Object Descriptors for Robotic Manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023), 9501–9507.

16. Z. Y. Qin, K. Fang, Y. K. Zhu, F. F. Li, and S. Savarese, "KETO: Learning Keypoint Representations for Tool Manipulation," in *IEEE International Conference on Robotics and Automation ICRA. IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), 7278–7285.

17. S. Lin, Z. Wang, Y. Ling, Y. Tao, and C. Yang, "E2EK: End-to-End Regression Network Based on Keypoint for 6D Pose Estimation," *IEEE Robotics and Automation Letters* 7, no. 3 (2022): 6526–6533, https://doi.org/10.1109/lra.2022.3174261.

18. X. Ma, D. Hsu, and W. S. Lee, "Learning Latent Graph Dynamics for Visual Manipulation of Deformable Objects," in *2022 International Conference on Robotics and Automation (ICRA)* (2020), 8266–8273.

19. A. Simeonov, Y. Du, A. Tagliasacchi, et al., "Neural Descriptor Fields: Se(3)-Equivariant Object Representations for Manipulation," in *2022 International Conference on Robotics and Automation (ICRA)* (2022), 6394–6400.

20. F. Zhang and Y. Demiris, "Learning Garment Manipulation Policies Toward Robot-Assisted Dressing," *Science Robotics* 7, no. 65 (2022): eabm6010, https://doi.org/10.1126/scirobotics.abm6010.

21. S. Bechtle, N. Das, and F. Meier, "Multimodal Learning of Keypoint Predictive Models for Visual Object Manipulation," *IEEE Transactions on Robotics* 39, no. 2 (2023): 1212–1224, https://doi.org/10.1109/tro.2022.3204509.

22. J. Gao, X. Jin, F. Krebs, N. Jaquier, T. Asfour, "Bi-Kvil: Keypoints-Based Visual Imitation Learning of Bimanual Manipulation Tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, (2024), 16850–16857.

23. N. D. Palo and E. Johns, Keypoint Action Tokens Enable In-Context Imitation Learning in Robotics, (2024).

24. H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao, "Copa: General Robotic Manipulation Through Spatial Constraints of Parts With Foundation Models," preprint arXiv:2403.08248 (2024).

25. M. Pan, J. Zhang, T. Wu, Y. Zhao, W. Gao, and H. Dong, "Omnimanip: Towards General Robotic Manipulation via Object-Centric Interaction Primitives as Spatial Constraints," in *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), 17359–17369.

26. Y. Zhang, S. Mittal, Z. Zhang, L. Ke, S. Srinivasa, and A. Gupta, ATK: Automatic Task-Driven Keypoint Selection for Robust Policy Learning, (2025).

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* 6, no. 2 (2021): 2870–2877, https://doi.org/10.1109/lra.2021.3062560.

28. L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "KPAM: Keypoint Affordances for Category-Level Robotic Manipulation," in *Robotics Research* (Springer International Publishing, 2019), 132–157.

29. J. Huang and K. W. Samuel Au, "Task-Oriented Grasping Position Selection in Deformable Object Manipulation," *IEEE Robotics and Automation Letters* 8, no. 2 (2023): 776–783, https://doi.org/10.1109/lra.2022.3231495.

30. S. Nasiriany, F. Xia, W. Yu, et al., "PIVOT: Iterative Visual Prompting Elicits Actionable Knowledge for VLMs," preprint arXiv: 2402.07872 (2024).

31. K. F. Levine, F. Liu, P. Abbeel, and S. Moka, "Open-World Robotic Manipulation Through Mark-Based Visual Prompting," in *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA*, (2024).

32. T. Chao, X. Anxing, D. Yuhong, and T. Hu, "FUNCTO: Function-Centric One-Shot Imitation Learning for Tool Manipulation," arXiv (2025).

33. L. Yan and S. Xinyu, "Shape- and Function-Centric Keypoint Representation," Roboflow Universe (2025), https://universe.roboflow.com/harbin-institute-of-technology-9q8vx/feed_keypoints-2-v1frv.

34. J. Glenn and Q. Jing. 2024. Ultralytics YOLO11, https://github.com/ultralytics/ultralytics.

35. Y. Liu, Y. Liu, Y. Yao, and M. Zhong, "A Laser Pointer-Based Interaction Method for Providing Robot Assistance to People With Upper Limb Impairments," *Expert Systems with Applications* 292 (2025): 128562, https://doi.org/10.1016/j.eswa.2025.128562.

36. C. Warren, "Global Path Planning Using Artificial Potential Fields," in *Proceedings, 1989 International Conference on Robotics and Automation*, (1989)316–321.

37. M. Oquab, T. Darcet, T. Moutakanni, et al., "Dinov2: Learning Robust Visual Features Without Supervision," preprint arXiv: 2304.07193 (2023).

38. A. Radford, J. W. Kim, C. Hallacy, et al., "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of Machine Learning Research*, (2021).

39. D. Alexey, B. Lucas, K. Alexander, and W. Dirk, "An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale," preprint arXiv:2010 (2020): 11929.

40. W. Dong, D. Huang, J. Liu, C. Tang, and H. Zhang, "Rtagrasp: Learning Task-Oriented Grasping From Human Videos via Retrieval, Transfer, and Alignment," in *2025 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2025), 1–7.

## Supporting Information

Additional supporting information can be found online in the Supporting Information section.
**Supporting Information S1:** cit270136-sup-0001-suppl-data.pdf. **Supporting Information S2:** cit270136-sup-0002-suppl-data.zip.