2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Sparse Meets Dense: Correspondence Guided Robotic Manipulation with Rigid-Deformable Interactions

Ziyu Zhu [2,1] Yue Chen [1] Xirui Liang [2,1] Hojin Bae [1] Yuran Wang [1] Zhen Yuan [1]
Ruihai Wu [†1] Hao Dong [†1]
[1]CFCS, School of Computer Science, PKU [2]School of EECS, PKU

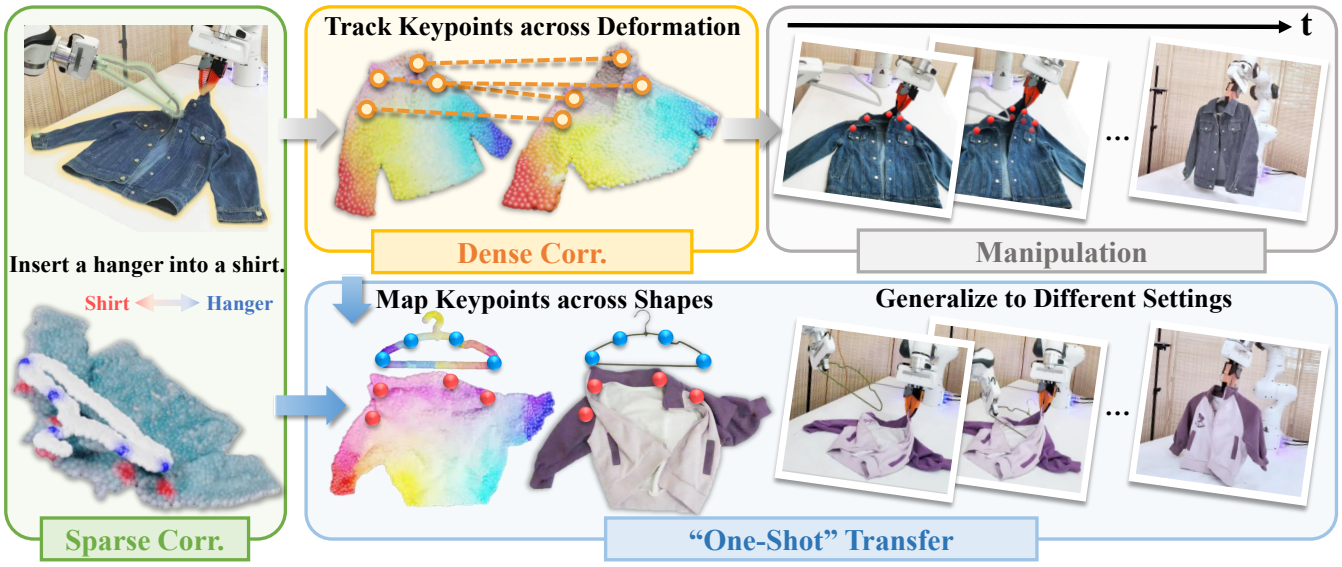

Fig. 1: **Correspondences for Rigid-Deformable Interactions.** We propose a hybrid correspondence-based representation for rigid-deformable interaction tasks. **(Left)** Sparse keypoint correspondences capture the interaction information between objects. **(Top-middle)** For long-horizon manipulation with a sequence of actions, dense correspondences can track keypoint accurately on soft bodies across deformations, and enable one-shot transfer by mapping keypoints to new shapes.

***Abstract*— Manipulation involving rigid-deformable interactions, such as hanging clothes or dressing humans, is common in daily life, making it essential for household robots. Compared to single-object manipulation or interactions between rigid bodies, these tasks are particularly challenging due to the rich multi-point contacts and the complex dynamics of the deformable bodies during interaction. Therefore, object-centric representations such as 6D poses or structural points without task-specific information become insufficient for these interactions. In this work, we propose a hybrid correspondence-based representation tailored for rigid-deformable interactions. First, to capture intricate interaction information, we introduce structure-, task-, and interaction-aware sparse keypoints. The keypoints are generated based on the global structures of both rigid and deformable objects, and filtered by their local interaction contacts. However, tracking these sparse keypoints through the interaction remains difficult due to the high-dimensional dynamics of deformable objects. Therefore, we further construct dense correspondences on the deformable objects for accurate keypoint tracking throughout the manipulation. This hybrid design combines the advantages of both representations: sparse keypoints encode rich, task-specific information for fine-grained manipulation, while dense correspondences ensure efficient tracking and generalization to novel deformations, shapes, and scenarios. Together, they enable one-shot transfer to new tasks with minimal demonstrations. Extensive experiments demonstrate the effectiveness and broad applicability of our method. Project Page: https://sparse-meets-dense.github.io/.**

I. INTRODUCTION

Manipulation tasks involving rigid-deformable interactions, such as hanging clothes or assisting with dressing, are common in daily-life scenarios, thus crucial for household robots. These tasks are significantly more challenging than single-object manipulation [40], [41] or rigid-rigid interactions [16] due to rich multi-point contacts and the complex, unpredictable dynamics of deformable objects [2], [32], [46]. Effective modeling of such interactions requires representations that capture both global object structure and local interaction contacts in a physically meaningful manner.

Existing approaches primarily rely on object-centric representations such as 6D poses [26], [37] or structural keypoints [7], [21]. While effective for rigid objects, these representations are limited in two ways: they cannot express the contact-rich interactions in rigid-deformable tasks, nor the high-DoF nature of deformable objects [1], [20], [35]. For deformable objects, recent work has shifted toward dense

representations—such as affordances [6], [40], [42] or dense correspondences [4], [39]. However, these methods face several limitations: they often require extensive annotated data, become less interpretable as object diversity increases, and introduce redundancy when applied to rigid bodies.

Therefore, we need a unified, generalizable, and data-efficient representation for such tasks. As rigid-deformable interactions are primarily governed by object structural alignments, we propose a hybrid correspondence-based representation to represent such alignments. We first introduce sparse keypoint correspondences (Fig. 1, left) to capture objects' global structures and local interaction information. To track these keypoints on high-dimensional deformable objects more accurately, we extend our representation with a dense correspondences module (Fig. 1, top middle). By combining the efficiency of sparse correspondence with the robustness of dense correspondences, our framework can generalize to new shapes from only a few demonstrations.

Concretely, given a single demonstration sequence and the point clouds of objects, we begin by extracting candidate structural keypoints on both objects. We then estimate the relative pose between the objects based on their final configuration, and use this local contact information to filter out task-relevant keypoint correspondences.

The framework then leverages the sparse keypoint correspondences to formulate constraints that can guide the robotic motion planning [7]. Due to the inherent alignment of keypoint pairs, they are particularly well-suited for defining such constraints to optimize manipulation trajectories.

However, constraint-based optimization requires accurate keypoint tracking over time, which is particularly challenging due to severe occlusions and the complex deformation of the soft bodies during interaction [47]. To address this, we introduce the dense correspondences that provide greater robustness in dynamic and continuous state spaces. Specifically, we adopt UniGarmentManip [39] and extend its correspondences defined on the canonical surface of the deformable object to the 3D space. This allows us to anchor the sparse keypoints into the dense space and track them accurately during interaction.

By taking the best of sparse and dense correspondences, we extract rich task-relevant information from demonstrations to guide the manipulation. Furthermore, the framework can easily generalize to novel shapes by mapping the extracted keypoints from the demonstration to novel objects undergoing different deformations with dense correspondences. We build 4 representative deformable-rigid interaction scenarios, and experimental results showcase the superiority of our proposed framework.

In conclusion, our primary contributions are as follows:

- We study novel tasks with contact-rich interactions between rigid and deformable bodies, and formulate them as optimization problems with keypoint constraints;
- We introduce sparse correspondences for structural keypoints to extract more fine-grained task-related physical information;
- We introduce dense correspondences to track the deformation of soft bodies during interaction, enabling real-time and accurate updates of constraints;
- Extensive experiments in both simulation and real world demonstrate the effectiveness and generalizability of our framework.

II. RELATED WORK

Structural Representations for Manipulation. Structural representations fundamentally influence manipulation capabilities. 6D poses efficiently capture long-range object dependencies and provide occlusion robustness [8], [9], [36], [37], but are limited to rigid objects since deformable objects lack well-defined poses. Keypoints offer flexibility and generalization [21], [30], [33], [38],yet are typically sparse and insufficient for capturing complex deformations of soft bodies [45], [48]. Moreover, keypoint proposal often requires manual task-specific annotations, limiting scalability [3], [44]. Although recent works [7], [17], [24] attempt to use VLMs to automate this process, they struggle to convey critical contact information for rigid-deformable interactions. To address these, we combine sparse keypoints to extract contact information while utilizing dense correspondences for deformation modeling.

Constrained Optimization in Manipulation. Constraints are often used to impose desired behaviors on robots [15]. Motion planning algorithms use geometric constraints to compute feasible trajectories that avoid obstacles and achieve goals [29], [31], [34]. For sequential manipulation tasks, task and motion planning (TAMP) [5], [10], [11] is a formulated as constraint satisfaction problems [13], [14], [18] with continuous geometric problems as subroutines. Recent works like ReKep [7] further utilize VLMs to automate this process. Building on this idea, our work also employs VLMs to analyze demonstrations and decompose the task into meaningful stages. However, instead of directly relying on VLMs to generate constraints for each stage, we draw inspiration from interaction constraints [22], [23], [27], [28] and leverage the physical interaction between rigid and deformable objects to derive constraints.

III. PROBLEM FORMULATION

Given a rigid-deformable task $\mathcal{T}$, we use VLM to decompose it into sequential stages $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\}$, each representing an independent sub-task. For instance, inserting a hanger into a shirt requires: $\mathcal{S}_1$=grab hanger, $\mathcal{S}_2$=lift collar, $\mathcal{S}_3$=insert right side, $\mathcal{S}_4$=insert left side, $\mathcal{S}_5$=lift up.

Our study mainly focuses on the manipulation stage containing rigid-deformable interactions (e.g., $\mathcal{S}_3$, $\mathcal{S}_4$). Given **one demonstration sequence**, the goal is to generate a successful action sequence $\mathcal{A} = \{P_1^{ee}, P_2^{ee}, \ldots, P_n^{ee}\}$, where each $P_i^{ee} \in SE(3)$ represents an end-effector pose. The action sequence is considered successful when interaction regions between rigid and deformable objects align within a desired spatial threshold.

Since target interaction regions are difficult to represent directly due to occlusion and dynamics, we introduce sparse keypoint correspondences ($\mathcal{C}_{sparse}$) to capture

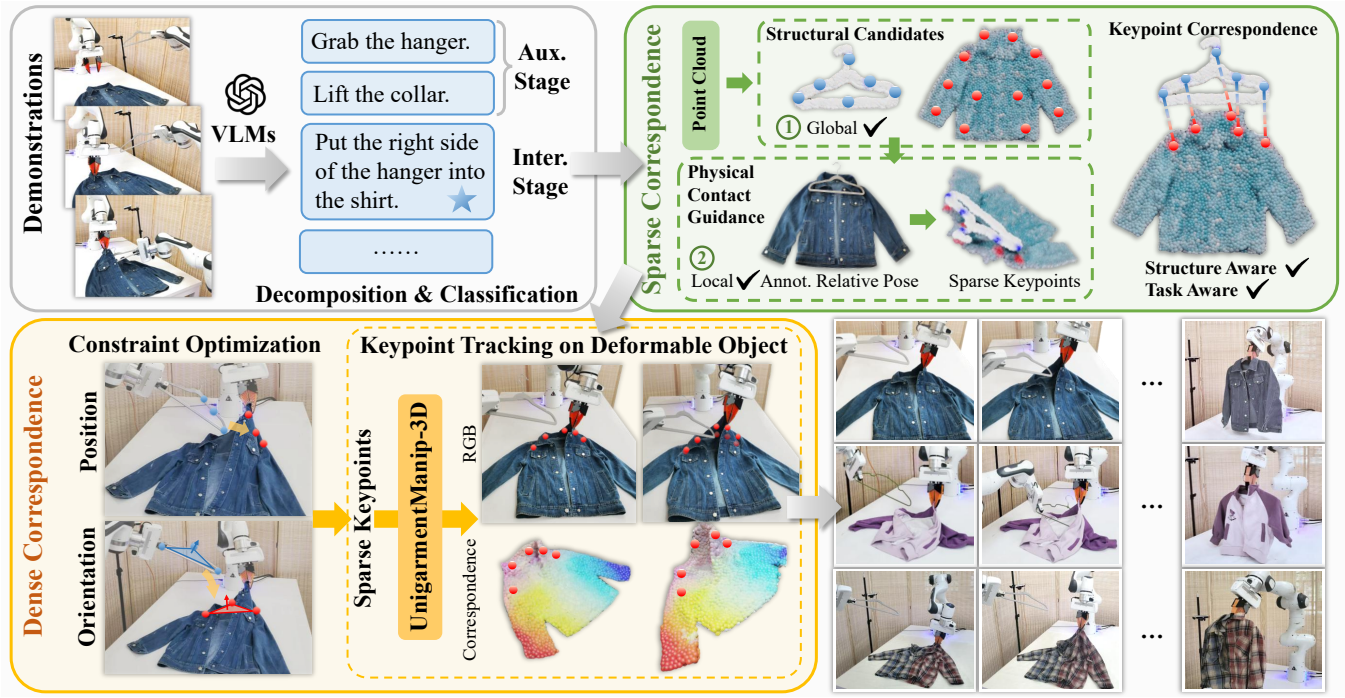

Fig. 2: **Framework Overview.** Given a task demonstration, VLMs decompose it into stages. For the stages with rigid-deformable interactions, we first extract sparse keypoints as spatial constraints, then apply constraint optimization to guide the manipulation sequence, with pretrained dense correspondence to track keypoints on deformable objects.

IV. METHOD

Our framework consists of four components: stage decomposition via VLMs (Section IV-A), sparse correspondences for structural and interaction modeling (Section IV-B), constraint optimization with dense correspondences (Section IV-C), and generalization to novel objects (Section IV-D).

A. Stage Decomposition

In our formulation, complex robotic tasks are decomposed into stages, which allows for the precise definition of task requirements and facilitates the execution of complex manipulation tasks. As shown in Fig.2, given a demonstration sequence and the task description, we first utilize VLM to decompose the task into multiple stages $\mathcal{S}$. These stages are then further categorized into two functional types: *auxiliary stages*, which involve preparatory actions, and *interaction stages*, which involve contact-rich interactions between rigid and deformable objects.

For auxiliary operation stages, the robot typically manipulates only a single object, and the operation point is often contained within the keypoints that constitute the sparse correspondences. Therefore, we construct the input prompt for VLMs using both the stage description and the candidate sparse keypoints. The VLMs then output both the specific manipulation point required for the auxiliary action and the constraint conditioned on that point.

B. Sparse Correspondences Modeling Interactions

In rigid-deformable manipulation tasks, obtaining accurate ground-truth interaction information is challenging due to surface deformation, complicated contact regions and dynamics. Motivated by the observation that most rigid-deformable interactions are mainly determined by the structural alignment of the target deformable and rigid objects, we propose to extract keypoints that can efficiently and effectively indicate such alignment, with the awareness of the object structures, contacts and the task. To this end, we first extract structural keypoints of objects (Section IV-B.1), then we further build keypoint correspondences to model interactions (Section IV-B.2).

1) Structural Keypoints of Objects: For both rigid and deformable objects, the structural keypoints can effectively represent them, and efficiently generalize to novel shapes.

For rigid objects, which often exhibit simple and stable geometric structures, we leverage its geometric and topological structure to efficiently select keypoints. In contrast, deformable objects have more complex and dynamic structures, making it harder to extract representative keypoints by only using geometric methods. To address this, we extract structural keypoints based on their underlying skeletons, which better capture the object's topology and deformation behavior.

For a class of rigid-deformable interaction tasks, given the point clouds of the rigid and deformable bodies (denoted as PC_r and PC_d), we first sample them to extract candidate keypoints with global structural features. Specifically, for PC_r, we adopt Farthest Point Sampling (FPS) to efficiently select N_r candidate keypoints, forming the set $\mathcal{K}_r = \{k_r^1, k_r^2, \ldots, k_r^{N_r}\}$ ($N_r \in [20, 100]$). For PC_d, we utilize Unigarmentmanip [39] to extract N_d skeleton points of the

deformable body, forming the set $\mathcal{K}_d = \{k_d^1, k_d^2, \ldots, k_d^{N_d}\}$ ($N_d \in [40, 200]$). These candidate keypoints are then further refined based on task-specific information, as detailed in (Section IV-B.2).

2) Structural Keypoints of Interactions: While the above keypoints indicate global structural information of rigid and deformable objects, they are insufficient to capture the fine-grained local physical interaction details. Therefore, in this section, we will further consider keypoints that encapsulate local interaction information.

Given a demonstration, the final state of object interaction often contains sufficient contact information. However, due to the severe occlusions that typically occur between rigid and deformable bodies at this stage, it is challenging to extract such information directly from visual data. Therefore, we try to derive it using more robust positional information instead.

Specifically, we manually annotate the relative pose between the rigid and soft bodies in the final state and transform PC_r and PC_d into the same 3D space according to the relative pose. Next, we traverse the keypoint sets K_r and K_d, and establish sparse correspondences by identifying mutually nearest keypoints between the rigid and deformable objects. Specifically, for each $k_r \in K_r$, we find its nearest neighbor $k_d \in K_d$, and require that k_r is also the nearest neighbor of k_d in K_r. In addition, we impose a distance threshold τ to retain only keypoint pairs that are close enough in interaction. The resulting sparse correspondences set $\mathcal{C}$ is defined as:

$$\begin{aligned}\mathcal{C} = \Big\{(k_r^i, k_d^j) \mid j = \arg\min_{j' \in \{1,\ldots,N_d\}} \|k_r^i - k_d^{j'}\|, \\ i = \arg\min_{i' \in \{1,\ldots,N_r\}} \|k_r^{i'} - k_d^j\|, \\ \|k_r^i - k_d^j\| < \tau\Big\}\end{aligned} \tag{1}$$

In order to execute the real-time constraint optimization more efficiently, we perform an additional point sampling operation on $\mathcal{C}$ to minimize the number of point pairs while preserving the geometric and interaction information. We apply the Furthest Point Sampling (FPS) algorithm to K_r (or K_d) and obtain indices $I = \{i_1, i_2, \ldots, i_{N_s}\}$ (or $I' = \{i'_1, i'_2, \ldots, i'_{N_s}\}$), where $N_s (3 \le N_s \le 6)$ is the number of sampled points. The final sparse correspondences $\mathcal{C}_{sparse}$ are then given by:

$$\mathcal{C}_{sparse} = \left\{\left(k_r^{i_k}, k_d^{i_k}\right) \mid i_k \in \{i_1, i_2, \ldots, i_{N_s}\}\right\} \tag{2}$$

C. Constraint Optimization for Manipulation Sequences

After obtaining the sparse correspondences, a natural question arises: how can these keypoint pairs be utilized to guide the manipulation task? In most existing works, the keypoint information is typically used as constraints to optimize the robotic motion planning. This process involves two fundamental questions: (1) how should these constraints be formulated (Section IV-C.1), and (2) how can these constraints be effectively optimized (Section IV-C.2)?

1) Optimization Constraints: To enable precise modeling of interactions between the rigid object and the deformable object, it is essential to focus on their alignment indicated by sparse keypoints. To quantify this alignment and thus guide the optimization of manipulation sequence, we decompose it into two components: a *positional constraint*, which aligns the rigid objects with the deformable objects in Euclidean space, and a *directional constraint*, which ensures correct relative orientation between the rigid object and the corresponding local region of the deformable object.

Positional Constraint: This constraint is based on the positional differences between keypoint pairs and enforces consistency in the relative positions of keypoint pairs across different frames or object states. Given $(k_r^i, k_d^i) \in \mathcal{C}_s$, we define a positional constraint as the Euclidean distance between them:

$$\mathbf{c}_{pos} = \frac{1}{N_s} \sum_{(k_r^i, k_d^i) \in \mathcal{C}_{sparse}} \left\|k_r^i - k_d^i\right\|_2 \tag{3}$$

Directional Constraint: This constraint is derived by constructing planes from keypoints, computing their corresponding normal vectors, and measuring the angular difference between these normals. Since a plane is uniquely determined by three points, we use the FPS algorithm to select three representative keypoint pairs when more than three are available. These selected pairs define two planes: $P_r = \{k_r^{i_1}, k_r^{i_2}, k_r^{i_3}\}$ on rigid object and $P_d = \{k_d^{i_1}, k_d^{i_2}, k_d^{i_3}\}$ on deformable object. The normal vectors of the two planes are computed as:

$$\begin{aligned}\mathbf{n}_r &= \frac{(k_r^{i_2} - k_r^{i_1}) \times (k_r^{i_3} - k_r^{i_1})}{\left\|(k_r^{i_2} - k_r^{i_1}) \times (k_r^{i_3} - k_r^{i_1})\right\|}, \\ \mathbf{n}_d &= \frac{(k_d^{i_2} - k_d^{i_1}) \times (k_d^{i_3} - k_d^{i_1})}{\left\|(k_d^{i_2} - k_d^{i_1}) \times (k_d^{i_3} - k_d^{i_1})\right\|}\end{aligned} \tag{4}$$

We then use the angular deviation to define the directional constraint as follows:

$$\mathbf{c}_{ori} = \Delta\theta = \arccos(\mathbf{n}_r \cdot \mathbf{n}_d) \tag{5}$$

To optimize the action sequence $\mathcal{A}$, it is necessary to establish a connection between constraint function and end-effector pose P^{ee}. For rigid object grasping, the target keypoint k_r can be directly computed via a rigid transformation of P^{ee}: $k_r = \phi_r(P^{ee})$ For deformable object grasping, we identify the keypoint k_d nearest to the end-effector and estimate its position based on the current end-effector pose: $k_d = \phi_d(P^{ee})$.

2) Constraint Optimization Guided Manipulation with Dense Correspondences: Once the interaction primitives and the corresponding spatial constraint are defined for each stage, the task execution can be formulated as a closed-loop optimization problem. To obtain the action sequence $\mathcal{A} = \{P_t^{ee}\}_{t=1}^n$, we optimize the end-effector pose at each time step t by minimizing the loss function. The optimization problem can be expressed as:

$$P_t^{ee} = arg\min_{P_t^{ee}} \left\{\sum_{j=1}^{N} \mathcal{L}_j(P_t^{ee})\right\}, \mathcal{L} = \{\mathbf{c}_{pos}, \mathbf{c}_{ori}\} \tag{6}$$

To enable closed-loop planning, we track keypoint pairs on both rigid and deformable objects. For rigid bodies, we follow prior studies [12] and leverage 3D tracker to track keypoints. For deformable objects, tracking sparse keypoints is challenging due to their high-dimensional state space and unpredictable deformations. Most existing research on deformable bodies seeks to use dense representations to capture their deformations [4], [39]. Following [39], we define dense correspondence as a point-wise matching between two garments (O_1, O_2), which evaluates the correspondence (normalized to $[-1, 1]$) in topology or function between each point pair (p_1, p_2), with p_1 from O_1 and p_2 from O_2. However, while these works build dense correspondence with deformable objects on tables or beds (for pick-and-place actions), our task requires point-level correspondences of objects in arbitrary 3D poses.

As directly obtaining reliable 3D point-level correspondences is challenging due to the instability of soft objects, we propose a data-efficient strategy that distills pretrained correspondences knowledge from UniGarmentManip [39], which is trained on garments in canonical near-planar states (e.g., laid on a flat surface). We leverage this planar prior to initialize dense feature learning in 3D space. Given a 3D point cloud $\mathcal{P}_{3D} \in \mathbb{R}^{N\times 3}$, we first obtain its planar projection:

$$\mathcal{P}_{2D} = \Pi_{xy}(\mathcal{P}_{3D}) \tag{7}$$

where Π_{xy} denotes projection along the Z-axis.

This projection serves only to align with the pretrained planar feature extractor and does not constitute the final representation.

We then use the pretrained UniGarmentManip model to extract 2D dense features:

$$\mathcal{F}_{2D} = \phi_{2D}(\mathcal{P}_{2D}), \quad \mathcal{F}_{2D} \in \mathbb{R}^{N\times d} \tag{8}$$

where ϕ_{2D} is the feature extractor.

Then, we train a PointNet++ nework ψ_{3D} that directly operates on the full 3D coordinates:

$$\hat{\mathcal{F}}_{3D} = \psi_{3D}(\mathcal{P}_{3D}). \tag{9}$$

We initialize the network by distilling the pretrained features:

$$\mathcal{L}_{\text{distill}} = \left\| \hat{\mathcal{F}}_{3D} - \mathcal{F}_{2D} \right\|_2^2. \tag{10}$$

To further improve the accuracy and robustness of dense correspondences, we annotate a relatively small set of 3D deformable object pairs with ground-truth correspondences $\mathcal{C}_{GT}$ and fine-tune the network with a contrastive loss $\mathcal{L}_{\text{con}}$ defined over matched and unmatched point pairs. This stage allows the network to incorporate full 3D geometric cues.

After obtaining dense correspondences $\mathcal{C}_{dense}$, we anchor sparse keypoints to its corresponding dense representations, and use $\mathcal{C}_{dense}$ to map keypoints to their corresponding positions after deformations.

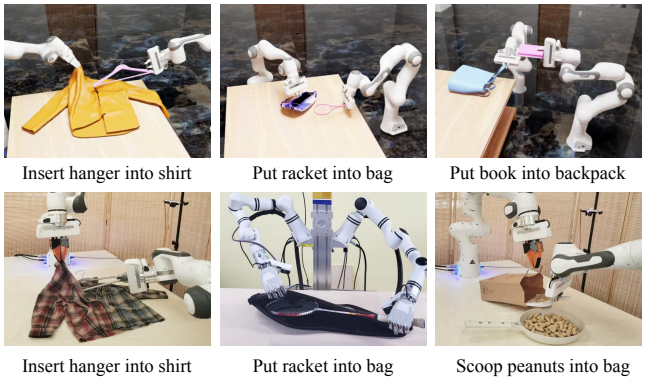

Fig. 3: **Task Illustrations.** Representative tasks in simulation and the real world.

D. Generalization to Novel Objects

To further generalize this pipeline to novel objects, it is crucial to accurately transfer the $\mathcal{C}_{sparse}$ from the demonstration instances to new object pairs. Therefore, we use $\mathcal{C}_{dense}$ to retrieve the dense feature embedding and perform a nearest-neighbor search in the dense feature space of the novel object to identify the most similar point. Specifically, let $\mathcal{C}_{sparse} = \{(k_r^i, k_d^i)\}_{i=1}^N$ denotes sparse keypoints in demonstration, we use $\mathcal{C}_{dense}^r$ for rigid body and $\mathcal{C}_{dense}^d$ for deformable body to map $\mathcal{C}_{sparse}$ to get new $\mathcal{C}'_{sparse}$:

$$\mathcal{C}'_{sparse} = \{(\mathcal{C}_{dense}^r(k_r^i), \mathcal{C}_{dense}^d(k_d^i)\}_{i=1}^N \tag{11}$$

Therefore, we can use $\mathcal{C}'_{sparse}$ to formulate new constraints and optimize $\mathcal{A}'$ in new scenarios.

V. EXPERIMENT

A. Environment, Assets, Data and Evaluation

We build the multi-material simulation environment using GarmentLab [19] implemented on Isaac Sim 4.5.0 [25]. As shown in Fig.3, we construct 4 different representative and realistic scenes: (1) Insert the hanger into the shirt; (2) Put racket into bag; (3) Put books into backpack; (4) Scoop peanuts into bag. For each task, we prepare a number of distinct rigid and deformable object assets with varying shapes to ensure diversity and generalization, and their configurations are generated automatically through code.

B. Baselines

We evaluate our method in both simulation and real-world scenarios. We compare our approach with two baselines: (1) **Annot. Rekep** [7], which employs human-annotated relational keypoint constraints and hierarchical optimization for real-time action generation; (2) **UniGarment** [39], which guides constraint formulation and optimization by establishing dense correspondences between rigid and deformable objects. Among them, Rekep relies entirely on sparse representations, while UniGarment adopts a fully dense representation. We compare our approach against both to highlight the advantages of taking the best of sparse and dense correspondences.

C. Results and Analysis

Tasks	Baseline Methods		Ours
	Wholly Sparse *(ReKep [7])*	**Wholly Dense** *(UniGarment [39])*	
SIMULATION RESULTS			
Insert hanger into shirt	60.9%	62.3%	**71.4%**
Put racket into bag	65.2%	65.4%	**73.6%**
Put books into backpack	58.7%	46.8%	**78.2%**
Scoop peanuts into bag	49.1%	45.1%	**70.9%**
Average	58.5%	54.9%	**73.5%**
REAL-WORLD RESULTS			
Insert hanger into shirt	5/10	5/10	**7/10**
Put racket into bag	5/10	6/10	**6/10**
Put books into backpack	5/10	5/10	**7/10**
Scoop peanuts into bag	4/10	4/10	**6/10**
Average	47.5%	50.0%	**65.0%**

TABLE I: **Quantitative Comparison with Baseline Methods.** Our approach demonstrates consistent improvements across all tasks in both simulation and real-world settings.

Tab. I shows quantitative comparisons with baselines. While Annot. Rekep utilizes manually annotated keypoints for its constraint optimization, it still underperform compared to our method. The main reason lies in its reliance on DINOv2 feature matching when tracking keypoints. As illustrated in the second row of Fig. 4, DINOv2 features lack consistency under soft-body deformation, leading to tracking errors and ultimately constraint optimization failures. UniGarment uses dense correspondences between rigid and deformable objects, providing useful interaction cues for constraint formulation. However, the high density increases computational cost, and inconsistent matches in occluded or highly deformable regions can reduce overall accuracy.

D. Ablation Study

	Insert hanger into shirt	Put racket into bag
Ours w/o $\mathcal{C}_{sparse}$	20.1%	28.6%
Ours w/o $\mathcal{C}_{dense}$	56.4%	63.2%
Ours w/o c_{pos}	0.0%	0.0%
Ours w/o c_{ori}	28.3%	32.7%
Ours w/ $3D-Tracker$	56.4%	63.2%
Ours	**71.4%**	**73.6%**

TABLE II: **Ablation Studies.** Section V-C provides detailed analysis.

To demonstrate the necessity of the proposed correspondences as well as designed constraints, we compare with the following ablated versions: (1) Ours w/o $\mathcal{C}_{sparse}$ that uses DINOv2 to generate keypoints instead of $\mathcal{C}_{sparse}$; (2) Ours w/o $\mathcal{C}_{dense}$ that tracks keypoints by matching pixel-wise DINOv2 features across frames instead of using $\mathcal{C}_{dense}$; (3) Ours w/o c_{pos} that removes the positional constraint; (4) Ours w/o c_{ori} that removes the orientational constraint; (5) Ours w/ $3D-Tracker$ that uses SpatialTracker [43] to track keypoints. Tab.II shows that sparse correspondences, composed of keypoints aware of structures and interactions, plays a crucial role in manipulation. Dense correspondences are also important. Compared to feature tracking based on the DINOv2 or recent 3D trackers, dense correspondence performs better in scenes with complex dynamics, where accurate tracking of deformable-object keypoints is required. The positional constraint is crucial for nearly all tasks. In contrast, the orientation constraint can be omitted in a few interactions where precise alignment is not strictly required.

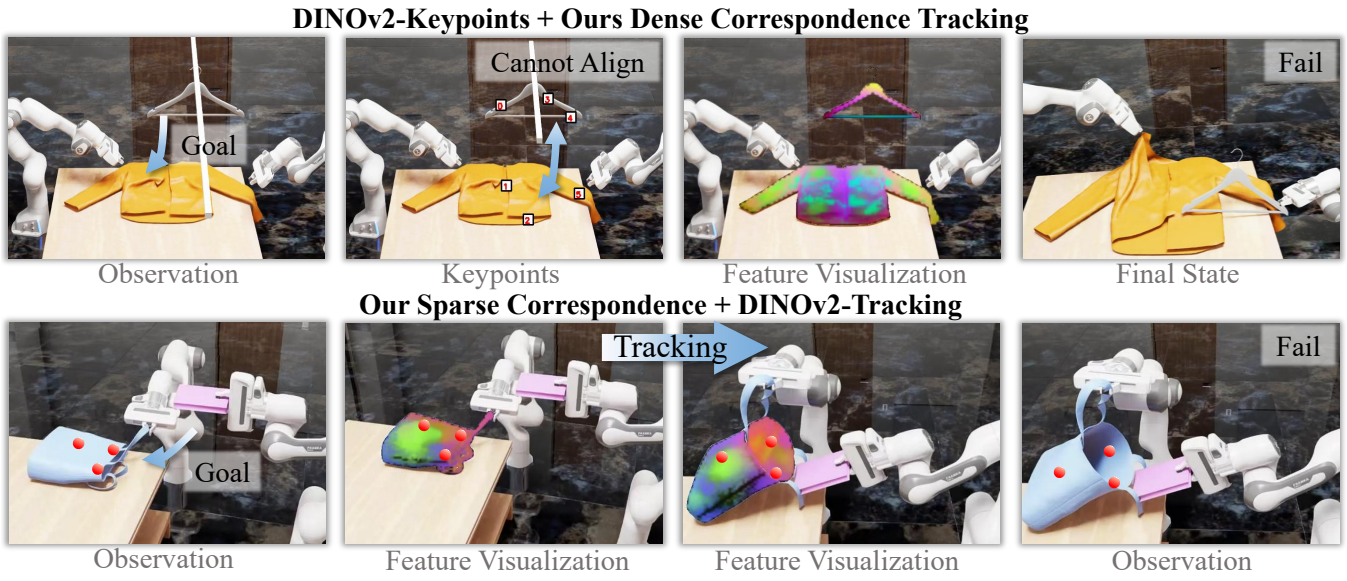

Fig. 4: **Ablation Studies.** The top and bottom rows respectively show results of replacing $\mathcal{C}_{sparse}$ and $\mathcal{C}_{dense}$ with DINOv2.

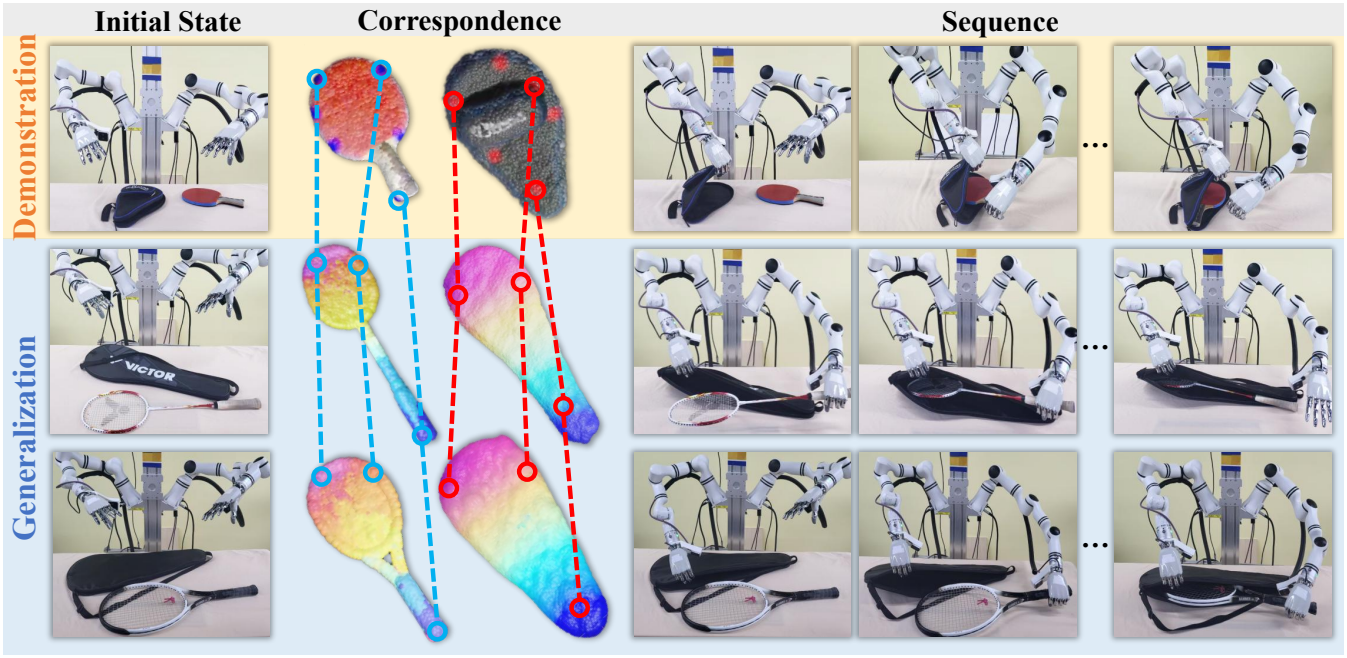

Fig. 5: **Dense Correspondences Enable Generalization.** Given the demonstration of table tennis racket (row 1), we can generalize to other rackets by mapping keypoints via $\mathcal{C}_{dense}$ (row 2,3).

Task	ReKep	UniGarment	Ours
Insert hanger into shirt	15.6%	57.1%	**65.2%**
Put racket into bag	20.7%	59.3%	**66.5%**
Put books into backpack	20.9%	39.4%	**69.1%**

TABLE III: **Generalization Results.** Section V-C provides detailed analysis.

E. Generalization

Tab. III and Fig. 5 demonstrate the generalization capability. For each task, we use 1 high-quality demonstration to extract sparse keypoints, which are mapped to novel objects with dense correspondences. Our generalization covers variations in object shapes, deformations and poses.

F. Real-World Evaluation

For real-world evaluation, we use a Franka Panda robot and two RealMan RM75-6F arms, each equipped with a Psibot G0-R dexterous hand. Depth images are captured using RealSense cameras. The last colum of Tab. I reports success rates. Supplementary video shows more demonstrations.

VI. CONCLUSIONS

We present a hybrid correspondence-based representation that effectively models complex interactions between rigid and deformable objects. We first introduce structure- and interaction-aware sparse keypoints correspondences, and then dense correspondences for accurate tracking. By taking the best of both, our approach enables robust and generalizable manipulation across new shapes and deformations. Extensive experiments in varied scenarios validate the effectiveness of our method.

Limitations and Future Work. While our method effectively models rigid-deformable interactions, tasks with deformable-to-deformable interactions remain challenging. Defining efficient representations and modeling such interactions are difficult due to complex dynamics, entanglement and occlusion.

REFERENCES

[1] Jan Kristof Behrens, Ralph Lange, and Masoumeh Mansouri. A constraint programming approach to simultaneous task allocation and motion scheduling for industrial dual-arm manipulation tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8705–8711, 2019. i

[2] David Blanco-Mulero, Oriol Barbany, Gokhan Alcan, Adrià Colomé, Carme Torras, and Ville Kyrki. Benchmarking the sim-to-real gap in cloth manipulation. *IEEE Robotics and Automation Letters*, 9(3):2981–2988, 2024. i

[3] Ling-Chen Chen, Chi-Kai Ho, and Chung-Ta King. Keystate: Improving image-based reinforcement learning with keypoint for robot control. In *2023 IEEE International Conference on Industrial Technology (ICIT)*, pages 1–6, 2023. ii

[4] Peter R. Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation, 2018. ii, v

[5] Caelan R. Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021. ii

[6] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*, pages 24–33. PMLR, 2022. ii

[7] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. i, ii, v, vi

[8] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*, pages 1470–1477. IEEE, 2011. ii

[9] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10):1194–1227, 2013. ii

[10] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical planning in the now. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1460–1467, 2010. ii

[11] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10):1194–1227, 2013. ii

[12] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. v

[13] Fabien Lagriffoul, Dimitar Dimitrov, Julien Bidot, Alessandro Saffiotti, and Lars Karlsson. Efficiently combining task and motion planning using geometric constraints. *The International Journal of Robotics Research*, 33(14):1726–1747, 2014. ii

[14] Fabien Lagriffoul, Dimitar Dimitrov, Alessandro Saffiotti, and Lars Karlsson. Constraint propagation on interval bounds for dealing with geometric backtracking. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 957–964. IEEE, 2012. ii

[15] Jens Lambrecht, Philipp Grosenick, and Marvin Meusel. Optimizing keypoint-based single-shot camera-to-robot pose estimation through shape segmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13843–13849, 2021. ii

[16] Yitong Li, Ruihai Wu, Haoran Lu, Chuanruo Ning, Yan Shen, Guanqi Zhan, and Hao Dong. Broadcasting support relations recursively from local dynamics for object retrieval in clutters. In *Robotics: Science and Systems*, 2024. i

[17] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. ii

[18] Tomás Lozano-Pérez and Leslie Pack Kaelbling. A constraint-based method for solving sequential manipulation planning problems. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3684–3691. IEEE, 2014. ii

[19] Haoran Lu, Ruihai Wu, Yitong Li, Sijie Li, Ziyu Zhu, Chuanruo Ning, Yan Shen, Longzan Luo, Yuanpei Chen, and Hao Dong. Garmentlab: A unified simulation and benchmark for garment manipulation, 2024. v

[20] Junqi Luo, Liucun Zhu, Liang Li, and Peitao Hong. Robot visual servoing grasping based on top-down keypoint detection network. *IEEE Transactions on Instrumentation and Measurement*, 73:1–11, 2024. i

[21] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–157. Springer, 2019. i, ii

[22] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 137–144. Eurographics Association, 2012. ii

[23] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):1–8, 2012. ii

[24] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv preprint arXiv:2402.07872*, 2024. ii

[25] NVIDIA. Isaac Sim. v

[26] Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. *arXiv preprint arXiv:2307.04767*, 2024. i

[27] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014. ii

[28] Michael Posa, Scott Kuindersma, and Russ Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1366–1373. IEEE, 2016. ii

[29] Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494, 2009. ii

[30] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2016. ii

[31] John Schulman, Yan Duan, Jonathan Ho, Alex X. Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *Int. J. Robotics Res.*, 33(9):1251–1270, 2014. ii

[32] Daniel Seita, Justin Kerr, John Canny, and Ken Goldberg. Initial results on grasping and lifting physical deformable bags with a bimanual robot. In *IROS Workshop on Robotic Manipulation of Deformable Objects in Real-world Applications*, volume 2, page 3, 2021. i

[33] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022. ii

[34] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. ii

[35] Yang Tian, Jiyao Zhang, Guowei Huang, Bin Wang, Ping Wang, Jiangmiao Pang, and Hao Dong. Robokeygen: Robot pose and joint angles estimation via diffusion-based 3d keypoint generation, 2024. i

[36] Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. *arXiv preprint arXiv:2201.12716*, 2022. ii

[37] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024. i, ii

[38] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. ii

[39] Ruihai Wu, Haoran Lu, Yiyan Wang, Yubo Wang, and Hao Dong. Unigarmentmanip: A unified framework for category-level garment manipulation via dense visual correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16340–16350, 2024. ii, iii, v, vi

[40] Ruihai Wu, Chuanruo Ning, and Hao Dong. Learning foresightful dense visual affordance for deformable object manipulation, 2023. i, ii

[41] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. VAT-mart: Learning visual action trajectory proposals for manipulating 3d ARTiculated objects. In *International Conference on Learning Representations*, 2022. i

[42] Ruihai Wu, Ziyu Zhu, Yuran Wang, Yue Chen, Jiarui Wang, and Hao Dong. Garmentpile: Point-level visual affordance guided retrieval and adaptation for cluttered garments manipulation, 2025. ii

[43] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space, 2024. vi

[44] Ruinian Xu, Fu-Jen Chu, Chao Tang, Weiyu Liu, and Patricio A. Vela. An affordance keypoint detection network for robot manipulation. *IEEE Robotics and Automation Letters*, 6(2):2870–2877, 2021. ii

[45] Jun Yamada, Shaohong Zhong, Jack Collins, and Ingmar Posner. D-cubed: Latent diffusion trajectory optimisation for dexterous deformable manipulation, 2024. ii

[46] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Ayzaan Wahid, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation, 2022. i

[47] Ling Zhou, Ruilin Wang, and Liyan Zhang. Accurate robot arm attitude estimation based on multi-view images and super-resolution keypoint detection networks. *Sensors*, 24(1), 2024. ii

[48] Peng Zhou, Pai Zheng, Jiaming Qi, Chengxi Li, Hoi-Yin Lee, Anqing Duan, Liang Lu, Zhongxuan Li, Luyin Hu, and David Navarro-Alarcon. Reactive human–robot collaborative manipulation of deformable linear objects using a new topological latent control model. *Robotics and Computer-Integrated Manufacturing*, 88:102727, 2024. ii