Probing Effective and Efficient Category-Level Articulated Object Pose Perception

Li Zhang, Xianhui Meng, Liu Liu, Haonan Jiang, Jianan Wang, Rujing Wang, Cewu Lu, *IEEE Member*, Jun Liu, *IEEE Senior Member*, Hong Zhang, *IEEE Life Fellow*.

Abstract—Category-level articulated object pose perception—encompassing both static pose estimation and dynamic pose tracking—is critical for embodied AI systems interacting with complex environments. Due to the inherent complexity and diverse motion structures of articulated objects, existing methods often exhibit limitations in adequately modeling kinematic constraints, handling self-occlusions, and meeting optimization requirements. Building upon EfficientCAPER [1], this work introduces CAPER++, a unified framework addressing these limitations through three key innovations: first, a joint-centric hierarchical model decomposes objects into a root part and constrained parts linked by joints, explicitly embedding kinematic constraints for geometrically consistent pose recovery. Second, an SE(3) manifold formulation leverages Lie algebra in the tangent space for singularity-free rotation representation and stable optimization, replacing error-prone direct regression. Third, for tracking, a proxy canonicalization strategy reformulates pose updates as SE(3) increment predictions relative to keyframes, enhanced by a dynamic keyframe mechanism to suppress drift. Extensive experiments on synthetic (ArtImage, PM-Videos), semi-synthetic (ReArtMix, ReArt-Videos), and real-world (RobotArm, RobotArm-Videos) benchmarks demonstrate state-of-the-art accuracy and robustness. CAPER++ achieves real-time inference (50 FPS) without post-processing, significantly advancing category-level articulated perception for real-world applications. Codes and datasets are available at project website: https://sites.google.com/view/caperplusplus.

Index Terms—Articulated Object, Pose Perception, Embodied AI

1 Introduction

Articulated object pose perception—encompassing both static pose estimation and dynamic pose tracking-is a cornerstone capability for embodied AI systems interacting with complex environments. Pose estimation aims to infer the joint configurations of articulated structures (e.g., doors, tools, or robotic limbs) from perceptual data, while pose tracking focuses on continuously updating these configurations over time, despite occlusion or motion ambiguity. The overarching goal is to enable machines to reason about and manipulate articulated objects with human-like fluency, a critical requirement for applications ranging from robotic manipulation [2], [3] and autonomous navigation [4], [5], [6] to augmented reality [7], [8]. Advances in this domain not only enhance the functional versatility of embodied agents but also deepen their understanding of physical interactions, bridging the gap between passive perception and actionable intelligence.

While instance-level 6D pose perception [9]—where models recognize known objects with precise CAD geometry—has seen remarkable progress, real-world embodied AI demands a more generalized paradigm: category-level 6D pose perception. This task requires inferring an object's 3D rotation, 3D translation, and 3D scale without instance-specific prior knowledge, relying solely on semantic category priors to handle unseen instances at test time. Such capability is inherently more challenging due to intra-category shape and appearance variations [10], partial observability, and the need for scale-invariant reasoning—issues exacerbated by the scarcity of densely annotated real-world training data. Yet, overcoming these challenges unlocks broader applicability: from robotic systems manipulating novel household objects to augmented reality interfaces adapting to diverse environments, category-level pose perception eliminates the impractical need for exhaustive instance-specific training, paving the way for scalable and adaptive embodied intelligence.

Recent years have witnessed growing interest in category-level articulated object pose perception, with representative approaches like A-NCSH [11] adopted from NOCS [12] for articulated object pose estimation and CAPTRA [13] for articulated object pose tracking. These methods introduce ***intra-class*** or ***intra-part*** representations in a canonical space, enabling pose recovery by estimating part-wise correspondences. While effective, current solutions suffer from several critical limitations:

- **Articulation Modeling.** Most methods [11] adopt a *rigid part-level* manner (Fig. 1 (a_1)), independently estimating each part's pose while ignoring kinematic constraints, leading to physically implausible configurations.
- **Pose Estimation.** Existing methods rely on *direct regression* of pose parameters [14], which is prone to suboptimal solutions (Fig. 1 (b_1)). Alternatively, *dense prediction* [12] strategies have been employed to improve accuracy (Fig. 1 (b_2)), but they often depend on complex post-processing procedures, resulting in significant computational overhead that hinders real-time deployment.
- **Pose Tracking.** Traditional tracking methods [15], [16]

- *Li Zhang and Liu Liu are with Hefei University of Technology, Hefei, 230026, China. Haonan Jiang is with Zhejiang University of Technology, Hangzhou, 310000. Jianan Wang is with Astribot, Shenzhen, 518000. Cewu Lu is with Shanghai Jiao Tong University, Shanghai, 200000. Hong Zhang is with Southern University of Science and Technology, Shenzhen, 518000. (e-mail: liuliu@hfut.edu.cn, 211122030127@zjut.edu.cn, wendyjnwang@gmail.com, lucewu@sjtu.edu.cn, hzhang@sustech.edu.cn.)*
- *Li Zhang, Xianhui Meng, Jun Liu, and Rujing Wang are with the University of Science and Technology of China, Hefei, 230026, China (e-mail:zanly20@mail.ustc.edu.cn, mengxh@mail.ustc.edu.cn, junliu@ustc.edu.cn, rjwang@iim.ac.cn).*

Manuscript received July 09, 2025. (Corresponding author: Liu Liu).

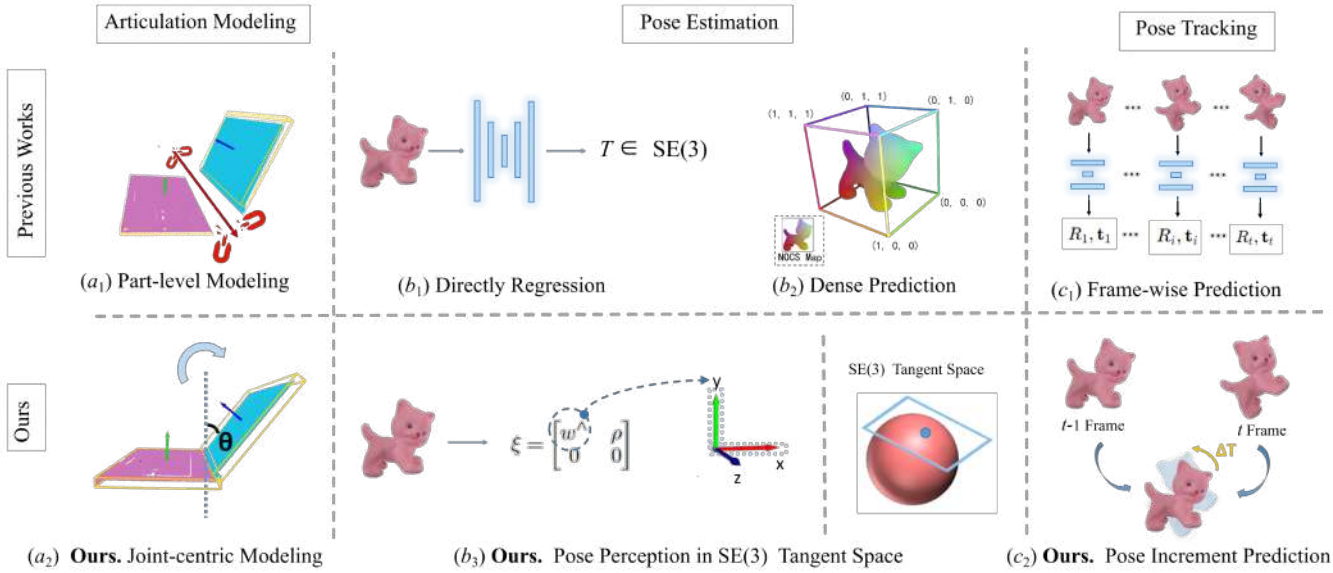

Fig. 1. **Novelty** of This Work. **(a) Articulation Modeling.** Traditional approaches often approximate articulated objects as a set of independent rigid parts, neglecting their inherent kinematic structure and thus frequently producing physically inconsistent predictions. In contrast, our method explicitly integrates kinematic constraints and employs a hierarchical modeling strategy to achieve more physically plausible articulated pose reasoning. **(b) Pose Estimation.** Existing methods either rely on direct regression, which tends to result in unstable predictions, or adopt dense prediction schemes, which incur substantial post-processing overhead. To address this, we introduce a decoupled rotation prediction strategy within the SE(3) tangent space, significantly improving both estimation accuracy and computational efficiency. **(c) Pose Tracking.** A straightforward method for pose tracking is frame-by-frame independent prediction, failing to exploit temporal dependencies across sequential frames. In this work, we incorporate a pose-increment learning mechanism that enables online, real-time, and high-precision tracking of articulated objects.

often depend on *dense per-frame predictions* (Fig. 1 (c_1)), resulting in excessive computation that hampers real-time performance for point cloud streams, limiting their practicality in dynamic scenarios.

Our Motivation. 1) intuitively, articulated object pose perception can be grounded in such a fundamental observation, in other words, the rigid parts of any articulated object can be categorized into two distinct types: (i) **root parts**, which admit unconstrained 6D motion (e.g., a cabinet's main body) with only one such part existing per object, and (ii) **constrained parts**, whose motion is restricted to predefined connected joints (e.g., doors/drawers) following physically plausible kinematics (with arbitrary counts allowed). This taxonomy enables **explicit exploitation of joint constraints** for geometrically consistent pose modeling.

2) Inspired by prior work in human pose estimation [17], we observe that representing poses on the SE(3) manifold using the associated Lie algebra $\mathfrak{se}(3)$ can effectively mitigate issues arising from direct parameterization in Euclidean space, such as the violation of rotation matrix orthogonality (i.e., $R^T R \neq I$). Compared to Euler angles, the Lie algebra avoids the gimbal lock problem. Unlike quaternion representations, it eliminates the ambiguity caused by sign non-uniqueness (i.e., $\mathbf{q}$ and $-\mathbf{q}$ representing the same rotation), which often leads to training instability and inference ambiguity. In summary, Lie algebra based representation is promising in enhancing numerical stability during optimization while maintaining physical consistency.

3) Efficient object tracking fundamentally relies on modeling motion coherence, which is the physical property that objects exhibit continuity across space and time. Conventional frame-by-frame estimation approaches neglect this critical temporal prior by treating each frame as an independent observation. As a result, they are prone to trajectory jitter, cumulative error, and reduced robustness in the presence of occlusion or sensor noise. More importantly, such methods fail to explicitly capture the typically small and structured incremental changes between consecutive frames. Recent advances in domains such as autonomous driving [18], video stabilization and deshaking [19], and person re-identification (Re-ID) [20] have demonstrated the effectiveness of incorporating kinematic priors, offering valuable insights for designing inter-frame prior learning strategies in this work.

Our Solution. 1) To address the limitation of existing part-level methods that ignore the intrinsic kinematic constraints between multi-parts, we propose a hierarchical articulation modeling strategy, as illustrated in Fig. 1 (a_2). In our formulation, an articulated object is decomposed into the root part and the constrained parts, as shown in Fig. 2. Specifically, we first estimate the 6D pose of the root part and then progressively infer the joint parameters and joint state of each connected constrained part according to the kinematic topology. This step-by-step process explicitly incorporates joint constraints to recover physically plausible per-part poses across the entire articulated structure.

2) To address the second limitation, this work presents a pose modeling framework that strictly adheres to the geometric structure of SE(3) (Fig. 1 (b_3)). Unlike conventional methods that approximate SE(3) as an Euclidean space, our approach performs pose estimation and tracking directly in its tangent space, thereby enforcing geometric consistency and mitigating non-linear errors during transformation learning. Specifically, we decouple the SE(3) transformation into translation and rotation components, modeling them independently. The rotational component is first represented using two orthogonal directional vectors and then mapped into the Lie algebra form in the tangent space of SE(3).

3) In this work, we extend our previous method [1] to handle continuous 6D pose tracking task. To account for the temporal continuity often ignored in prior methods, our core idea is to introduce a proxy-canonicalization strategy, which reformulates the tracking task as pose increment prediction relative to the keyframe (Fig. 1(c_2)). This effectively reduces the frame-to-

frame jitter commonly observed in single-frame estimations. In addition, we develop a dynamic keyframe selection mechanism that adaptively selects keyframes to mitigate drift, significantly improving the stability and robustness of long-term tracking.

This work constitutes a substantial advancement over our preliminary version presented at NeurIPS 2024 [1]. Compared to the conference paper, this work introduces several significant improvements and enhancements: (1) To address the potential violation of orthogonality constraints in decoupled rotation direction vectors, we explicitly incorporate orthogonality regularization and adopt $\mathfrak{se}(3)$ Lie algebra for rotation representation and optimization, thereby improving geometric consistency and numerical stability; (2) To mitigate issues in the previous work where joint parameters were directly regressed or part scale was overlooked, we propose a scale-aware heatmap voting scheme for more robust joint parameter prediction; (3) We further extend CAPER++ to the task of articulated pose tracking and conduct comprehensive evaluations in synthetic, semi-synthetic, and real-world scenarios to validate its generalization and robustness in dynamic environments; (4) A series of experiments and visual analyses are conducted to provide deeper insights into the underlying mechanisms and effectiveness of CAPER++ in both static pose estimation and dynamic tracking tasks; (5) Lastly, we present a systematic review of related literature on articulated object pose estimation, pose tracking, and pose perception on SE(3) manifolds, highlighting the strengths and limitations of existing methods and providing theoretical insights and practical guidance for subsequent studies.

In summary, the contributions can be concluded as follows:

- We present CAPER++, a unified, efficient, and robust framework for category-level articulated object 6D pose estimation and pose tracking, operating end-to-end without optimization or post-processing.
- We introduce a joint-centric formulation for articulated pose estimation that predicts the root-part pose and joint states jointly, enabling effective exploitation of kinematic constraints. An orthogonal decoupling strategy and an $\mathfrak{se}(3)$-based rotation representation are employed to improve numerical stability and physical interpretability, with effectiveness consistently demonstrated in experiments.
- We extend the framework to articulated pose tracking by modeling inter-frame pose increments through a proxy normalization strategy, which enhances temporal coherence. A dynamic keyframe selection mechanism is further introduced to reduce long-term error accumulation, leading to improved accuracy and stability in extensive tracking evaluations.

2 Preliminaries: SE(3) Manifolds and $\mathfrak{se}(3)$ Lie Algebra Tangent Space

Lie algebra provides a linearized representation of Lie groups by describing their local, infinitesimal behavior near the identity element. It offers a compact six-dimensional vector space formulation for both rotation and translation. Through exponential and logarithmic mappings, it enables efficient conversion between the tangent space and the group manifold, facilitating local pose updates in a linear space. First of all, let's give a simple introduction about a special Lie group named SE(3) [21]. It forms the mathematical foundation for representing rigid-body motions in 3D space, combining rotations and translations into a unified framework. At its core, SE(3) consists of 4×4 homogeneous transformation matrices of the form:

$$T = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{1}$$

where the rotation component $R \in \mathrm{SO}(3)$ satisfies the orthonormality constraints $R^T R = I$ and $\det(R) = 1$, while $\mathbf{t}$ represents translation. Crucially, SE(3) exhibits the structure of a smooth manifold that locally resembles Euclidean space but globally enforces the geometric constraints of rigid transformations. This manifold property is fundamental to modern pose estimation, as it inherently prevents degenerate solutions that violate physical motion constraints. Note that throughout this work, SE(3) is used to denote the SE(3) manifold or SE(3) transformations, while $\mathfrak{se}(3)$ refers specifically to the corresponding Lie algebra.

The local linearization of SE(3) is achieved through its associated Lie algebra $\mathfrak{se}(3)$, which describes infinitesimal motions around the identity transformation. The Lie algebra elements, known as twists, take the form of 4×4 matrices parameterized by angular and linear velocities:

$$\xi^\wedge = \begin{bmatrix} w^\wedge & \rho \\ 0 & 0 \end{bmatrix}, \quad \text{where} \quad w^\wedge = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \tag{2}$$

where $w^\wedge \in \mathfrak{so}(3)$ is the skew-symmetric matrix representation of the angular velocity vector $w = [w_1, w_2, w_3]^T \in \mathbb{R}^3$, $\wedge$ means skew-symmetric mapping, *i.e.*, vector to skew-symmetric matrix operator. For computational purposes, twists are typically represented as 6D vector $\xi = (w, \rho)^T$, combining rotational and translational components into a minimal parametrization. The connection between the Lie algebra and the Lie group is established through the exponential map $\exp : \mathfrak{se}(3) \to \mathrm{SE}(3)$, which converts these infinitesimal motions into finite transformations. This mapping can be efficiently computed using the Rodriguez formula for the rotational part while maintaining the group structure.

The practical significance of this mathematical framework becomes evident in pose estimation and tracking applications. Unlike Euler angles that suffer from gimbal lock [22] or quaternions [23] with sign ambiguity, the SE(3) manifolds representation provides a singularity-free parametrization of rigid motions. In addition, the tangent space $\mathfrak{se}(3)$ offers a linearized domain for optimization, where gradient-based methods can operate without violating the underlying geometric constraints. Recent works [17], [24], [25], [26] have shown that the tangent space representation is necessary and efficient for solving optimization problems on SE(3) manifolds.

3 Related Work

3.1 Category-level Articulation Pose Estimation

Category-level articulation pose estimation marks a paradigm shift from traditional instance-level modeling by focusing on generalization across unseen instances within a given object category [27]. This approach not only challenges the long-standing assumption of fixed geometries but also calls for learning shape-invariant kinematic reasoning, thereby paving the way for recent advances in articulation understanding via deep learning. Compared to rigid object pose estimation, this task aims to infer the 6D pose of each rigid part within an articulated object, where the

parts are linked by joints exhibiting various kinematic constraints. The overarching goal is to recover per-part poses that accurately reflect the articulation state of the whole object.

Deep learning has significantly advanced this area, with convolutional neural networks (CNNs) demonstrating strong capabilities in learning spatial patterns from raw input. For instance, Wei et al. [28] proposed a multi-stage CNN architecture for human pose estimation, achieving state-of-the-art performance on datasets such as MPII Human Pose [29] and COCO [30]. However, most early efforts [31], [32] were instance-specific and struggled to generalize to novel object configurations, limiting their applicability to real-world scenarios involving diverse categories. To overcome this, recent works have targeted the more challenging category-level setting. Li et al. [11] introduced A-NCSH, adapting normalized coordinate space [12] to articulated objects. Liu et al. [33] extended this direction by incorporating part-pair reasoning into real-world applications. Most recently, Zhang et al. [34] proposed a unified framework that bridges category-level pose estimation for both rigid and articulated objects.

Despite these advancements, existing methods often ignore ***the constraints of the kinematic structure*** *and may struggle under real-world challenges such as occlusion and variability.*

3.2 Category-level Articulation Pose Tracking

Category-level articulation pose tracking aims to bridge instance-agnostic generalization with dynamic scene understanding, where the variability of unseen articulated objects and their kinematic structures poses a significant challenge [13]. This task inherits complexities from both articulated pose estimation and multi-part tracking, requiring models to be robust to shape variation, part topology, and motion diversity. While recent advances in category-level articulated pose estimation (e.g., A-NCSH [11], OMAD [14]) have introduced joint-aware representations to support pose inference for unseen instances, these methods are primarily designed for single-frame prediction and lack temporal coherence [15], [16]. Consequently, tracking articulated objects over time demands further exploration into temporal articulation modeling that captures kinematic constraints, part-wise motion, and cross-frame continuity.

Articulation pose tracking extends static pose estimation by incorporating temporal consistency [13], aiming to continuously estimate the 6D poses of multiple rigid parts under kinematic constraints across sequential frames. Unlike rigid object tracking, which assumes a single SE(3) transformation between frames, articulation tracking must account for joint-driven part motions [35], such as those governed by revolute or prismatic joints. While early rigid tracking methods—from ICP-based alignment to learning-based approaches like FlowNet3D [36]—laid the groundwork for dense motion estimation, they fall short when handling independently moving parts in articulated objects. This shift toward articulation-aware tracking introduces new challenges, including motion propagation through kinematic priors [37], [38], robustness to occlusion, and real-time efficiency by avoiding redundant per-frame regressions [38], [39].

Despite progress, prior studies often neglect ***the motion continuity*** *across consecutive frames or insufficiently utilize the spatiotemporal priors embedded in adjacent frames. In this work, we introduce a dynamic keyframe mechanism that exploits inter-frame priors within* ***proxy canonical space****, thereby enhancing the performance of articulated object pose tracking.*

3.3 Pose Perception on SE(3) Manifolds

The group of 3D rigid-body transformations, SE(3), forms a Lie group that jointly models rotation and translation within a unified geometric structure, making it well suited for object pose representation [40], [41]. Compared to traditional parameterizations such as Euler angles, quaternions, or 6D vectors—which often suffer from singularities, discontinuities, or weakened geometric consistency [42], [43], [44], [45], the SE(3) manifold provides a globally consistent and differentiable space that supports interpolation, averaging, and optimization directly on the manifold. Within this framework, Lie algebra has been widely adopted across multiple research paradigms, though with distinct functional roles. In classical robotics and optimization-centric formulations, Lie algebra is primarily used to linearize local pose perturbations, enabling unconstrained iterative optimization while preserving manifold constraints [46], [47], [48], [49]. Similar principles underpin filtering and state estimation on Lie groups, while recent learning-based approaches incorporate manifold awareness through SE(3)-aware loss functions, geodesic distances [50], [51], [52], or tangent-space gradient computation to improve numerical stability [53], [54], [55].

From a modeling perspective, prior work on pose perception over SE(3) can be broadly grouped into complementary directions. One line [56], [57], [58] formulates pose estimation and tracking directly on Lie groups with explicit kinematic modeling. A related but distinct body of work [59], [59], [60], [61] focuses on pose representations and motion contexts, systematically analyzing how parameterization choices affect learning dynamics and temporal prediction [62], [63]. Despite progress, many existing works still treat SE(3) as a Euclidean space, ignoring its curved geometry and consequently leading to suboptimal pose estimation and tracking performance. In contrast, operating in the tangent space of SE(3) enables principled local linearization, facilitates stable gradient-based optimization, and preserves manifold constraints by construction, which is essential for accurate and consistent pose perception in complex scenarios. *From this perspective, our method can be viewed as an extension of EfficientCAPER [1]. While inheriting its efficient inference characteristics, we further generalize it to real-time pose tracking by modeling incremental motions in the tangent space of SE(3) rather than in Euclidean space. This design enables temporally coherent tracking while maintaining both computational efficiency and geometric consistency.*

4 Problem Statement

To achieve a robust algorithm for the **C**ategory-level **A**rticulation **P**ose **P**erception task (**C-APP**), our core idea is to introduce a two-step end-to-end pose estimator without any post-processing or optimization operation. Concretely, for pose estimation task, given the partial observation $\mathcal{P} \in \mathbb{R}^{N\times 3}$ with K rigid parts, our CAPER++ network performs predictions under unknown CAD models for: (1) per-part 3D rotation $R^{(k)} \in SO(3)$. (2) per-part 3D translation $\mathbf{t}^{(k)}$. (3) per-part 3D scale $\boldsymbol{s}^{(k)}$. In total, these together constitute the final pose estimation result: $T = \{R^{(k)}, \mathbf{t}^{(k)}, \boldsymbol{s}^{(k)}\}_{k=1}^{K}$ (***output***), where $\{R^{(k)}, \mathbf{t}^{(k)}\}_{k=1}^{K} \in SE(3)$. For pose tracking task, given the live point cloud stream $\{\mathcal{P}_t\}_{t\geq 0}$ as ***input***, where t denotes the t frame in videos. Our ***target*** is to track the per-part pose $T_t^{(k)} = \{R_t^{(k)}, \mathbf{t}_t^{(k)}, \boldsymbol{s}_t^{(k)}\}_{k=1}^{K}$ at all frames $t > 0$ in an online manner, given a live stream

of point clouds $\{\mathcal{P}_t\}_{t\geq 0}$ containing the per-part pose $T_0^{(k)} = \{R_0^{(k)}, \mathbf{t}_0^{(k)}, \boldsymbol{s}_0^{(k)}\}_{k=1}^K$ at the 0 frame.

Here, we formulate a new paradigm for the C-APP task named **CAPER++**, whose pipeline can be illustrated as follows: under the proposed joint-centric modeling strategy, we first predict the 3D rotation $R_{\mathbf{root}}$ and 3D translation $\mathbf{t}_{\mathbf{root}}$ for root part, where the 3D rotation is encoded as a decoupled rotation representation, and they are transformed by Lie algebra which is in the tangent space of SE(3). After the canonicalization, we conduct the prediction of 3D rotation $R_{\mathbf{cons}}^{(k)}$ and per-part 3D translation $\mathbf{t}_{\mathbf{cons}}^{(k)}$ for each constrained part, which are represented as joint state $\boldsymbol{\theta}_{\mathbf{cons}}^{(k)}$. In total, the rotation and translation together constitute an articulation pose estimation result $\{R^{(k)}, \mathbf{t}^{(k)}, \boldsymbol{s}^{(k)}\}_{k=1}^K$, where $\{R^{(k)}\}_{k=1}^K = \{R_{\mathbf{root}}, \{R_{\mathbf{cons}}^{(k)}\}_{k=2}^K\}$, $\{\mathbf{t}^{(k)}\}_{k=1}^K = \{\mathbf{t}_{\mathbf{root}}, \{\mathbf{t}_{\mathbf{cons}}^{(k)}\}_{k=2}^K\}$ and $\{\boldsymbol{s}^{(k)}\}_{k=1}^K = \{\boldsymbol{s}_{\mathbf{root}}, \{\boldsymbol{s}_{\mathbf{cons}}^{(k)}\}_{k=2}^K\}$. Besides, CAPER++ also predicts per-point part segmentation δ (optional).

When extending CAPER++ to pose tracking task, our core idea is to fully leverage inter-frame priors by reformulating the per-frame absolute pose prediction problem as the estimation of relative pose increments between adjacent frames (i.e., $t-1$ and t frame). Specifically, we apply the pose estimated from the previous frame $T_{t-1}^{(k)}$ to inverse-transform the current frame's point cloud $(T_{t-1}^{(k)})^{-1} \cdot \mathcal{P}_t^{(k)}$, thereby aligning it into a *canonical proxy space* point cloud $\ddot{\mathcal{P}}_t^{(k)}$ and eliminating the influence of global transformations. To mitigate the impact of accumulated errors on long-term tracking stability, we further introduce a dynamic keyframe selection mechanism that resets the keyframes, thereby enhancing the overall robustness and accuracy of the system.

We assume that there is only one root part in an articulated object, and the corresponding index is 1. **Note that we use $'$ to represent the variables in canonical space (e.g., $\mathcal{P}'$), $\ddot{}$ to represent the variables in proxy canonical space (e.g., $\ddot{\mathcal{P}}$), which is distinguished from variables in camera space (e.g., $\mathcal{P}$). To avoid ambiguity and better clarify the variables, we use the symbol $*$ to indicate the GT variables, and $\hat{}$ to indicate the predicted variables.**

5 METHOD

Unlike part-level methods that overlook kinematic dependencies and are prone to occlusion-related errors, this work adopts a *joint-centric* perspective, formulating pose estimation as the prediction of joint states. In this view, the motion of rigid parts is directly driven by joint configurations under kinematic constraints. We categorize the rigid parts of articulation into two types: *root part* and *constrained parts*, as illustrated in Fig. 2. Root parts can move arbitrarily in 3D space, while constrained parts are limited to motions along predefined joint axes, determined by the underlying kinematic structure. The motion of constrained parts is governed by the specific joint type. Following the articulation models proposed in A-NCSH [11] and OMAD [14], we define two joint types:

(1) Revolute Joint: This joint enables rotational motion within the articulated object. The joint state, denoted by $\boldsymbol{\theta}_r$, is represented by the relative rotation angle compared to the rest state. The joint parameters are $\boldsymbol{\phi}_r = (\boldsymbol{u}_r, \boldsymbol{q}_r)$, where $\boldsymbol{u}_r$ represents the direction of the joint axis and $\boldsymbol{q}_r$ denotes the pivot point position of the joint axis in the canonical coordinate space.

(2) Prismatic Joint: This joint allows the constrained part to move linearly along the joint direction. The joint state, denoted by $\boldsymbol{\theta}_p$, is represented by the relative distance compared to the rest state. The joint parameters are defined as $\boldsymbol{\phi}_p = (\boldsymbol{u}_p)$, where $\boldsymbol{u}_p$ represents the direction of the joint axis.

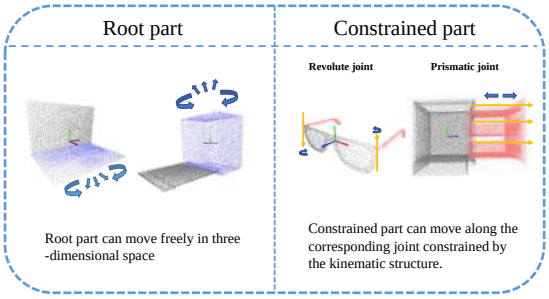

Fig. 2. **Root Part and Constrained Part in Articulated Objects**. These two kinds of parts are connected by joints.

By adopting the joint-centric pose modeling approach, we can establish a one-to-one correspondence between parts and joints. Therefore, the pose of the entire articulated object, which consists of K parts, can be described by a sequence of joint states $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^{(k)}\}_{k=2}^K$ for the constrained parts and the pose $\{R_{\mathbf{root}}, \mathbf{t}_{\mathbf{root}}\}$ for the root part.

The overall pipeline of CAPER++ is shown in Fig. 3. Given a point cloud of an articulated object, our method estimates part-wise poses in two steps. First, hybrid-scope features [64] are extracted using an HS-Encoder [64], and passed to one translation decoder and two rotation decoders to predict the translation and rotation of a root part. Then, the input point cloud is canonicalized using the estimated pose of the root part, improving robustness. Then, the second HS-Encoder extracts features that are used by four subbranches to estimate joint state, joint parameters, scale, and part segmentation. Finally, the poses of constrained parts are recovered based on the predicted joint state and the root part's pose. Note that the input point cloud can be obtained via back-projection from RGB-D images.

5.1 Root Part Pose Estimation

5.1.1 Rotation Estimation

Previous works [65], [66] were originally proposed for estimating the 6D pose of rigid objects from point cloud. For the root part pose estimation, we extend this approach by using the point cloud from all rigid parts of the articulated object, rather than only the root part (as validated in the ablation study Tab. 7). This strategy offers two key advantages: (1) it eliminates the need for part segmentation, thus enhancing the efficiency of the method, and (2) it allows information from other constrained parts to help estimate the pose of the root part, especially in cases of strong symmetry.

Given the point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$, we design an encoder-decoder architecture consisting of an HS-Encoder with four HS-layers and two rotational decoder with four 1-D convolution layers. Features are extracted from each HS-layer after encoding. To obtain a global feature that captures the overall information, we apply max pooling to the point cloud dimension of the last layer's output. This global feature is then concatenated with the outputs of each HS-layer to form the final encoder output.

To estimate the rotation matrix $R_{\mathbf{root}}$, we decompose it into three directional vectors: x, y, and z. Two of these vectors are selected based on the object's symmetry to better guide the network in learning shape priors. For instance, as illustrated in Fig. 4, for a laptop that is symmetric with respect to the y-z plane, we select the y- and z-axis directions; for a box, we choose

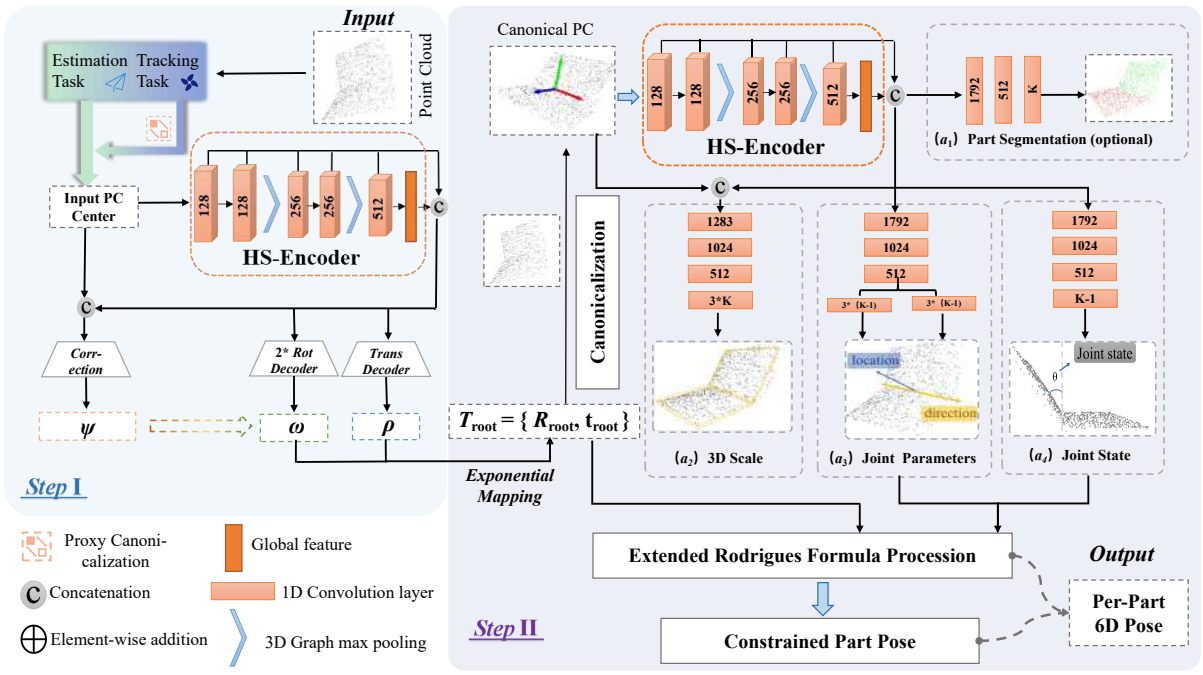

Fig. 3. **The Proposed CAPER++ Framework** is designed to support both pose estimation and pose tracking for articulated objects. In the pose estimation task, the input is a single-frame point cloud, from which the network directly predicts the pose of the target articulated object. In the pose tracking task, it takes a sequence of point clouds as input. The framework iteratively updates the pose based on the previous frame's estimate, producing a continuous stream of poses over time to enable robust object tracking. Our method is built upon a two-step framework: Step I estimates the pose of the root part (Sec. 5.1), followed by Step II, which estimates the pose of the constrained parts (Sec. 5.2). This articulated modeling approach is further extended to the pose tracking task, as described in Sec. 5.3.

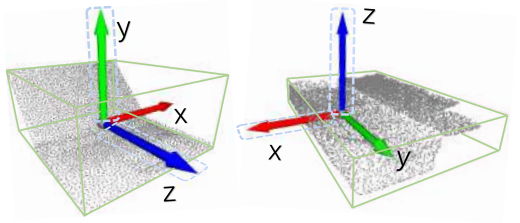

Fig. 4. **Rotation Decoupled Strategies.** We select the corresponding two axes according to the symmetry priors.

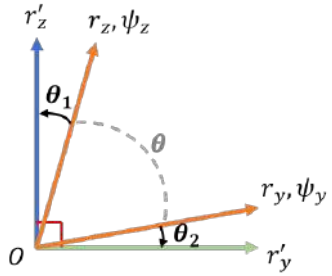

Fig. 5. **The Illustration of the Proposed Vertical Correction Mechanism.** r_y,r_z are the initial predicted rotation direction vectors, ψ_y,ψ_z are their corresponding correction factors. $r_y^{'}$,$r_z^{'}$ are the perpendicular direction vectors which are obtained by maximizing Eq.3.

the x- and z-axis directions. Therefore, two selected vectors are generated and normalized to represent predicted plane normals in the decoding step.

It is evident that the three direction vectors derived from a rotation matrix must be mutually orthogonal. However, in practice, the two direction vectors predicted by our previous work Efficient-CAPER [1] may violate this orthogonality constraint. To address this issue, we additionally introduce a correction mechanism that applies correction factors to refine the predicted direction vectors. This mechanism enhances robustness in the final 3D rotation recovery process.

Specifically, taking the laptop as an example, the predicted r_z and r_y may not be orthogonal (assuming an acute angle between them). As shown in Fig. 5, to enforce orthogonality, a certain angle adjustment is needed. This raises the question of which direction vector should be prioritized for adjustment. This is where the correction factor comes into play. Based on this, we propose ***Vertical Correction Mechanism***. The correction factor for each direction vector determines the extent to which the respective direction vector needs to be adjusted. This correction can be represented by the following optimization function:

$$\begin{aligned} \theta_1^*, \theta_2^* = \text{argmax} \ \ & (-\log(\psi_z \theta_1^2 + \psi_y \theta_2^2)) \\ s.t. \ \ & \theta + \theta_1 + \theta_2 = \pi/2, \end{aligned} \tag{3}$$

where θ denotes the angle between r_z and r_y. (θ_1, θ_2) are the residual angles. (ψ_z, ψ_y) represent the correction factors of r_z and r_y. with the optimized (θ_1^*, θ_2^*), the calibrated plane normals $r_y^{'}$ and $r_z^{'}$ can be calculated. Then the remaining direction vector is obtained as $r_x^{'} = r_y^{'} \times r_z^{'}$. Eventually, we compose the three vectors into the rotation matrix $R = [r_x^{'}, r_y^{'}, r_z^{'}]$.

5.1.2 Translation Estimation

For the translation $\mathbf{t}$, we first zero-center the input point cloud $\mathcal{P}$ by subtracting its mean, resulting in $\widetilde{\mathcal{P}}$. Next, we construct a decoder similar to the rotation decoder, consisting of 1-D convolution layers. Following [67], we estimate the residual between the mean of $\mathcal{P}$ and the ground truth translation. The feature is formed by concatenating the outputs of each HS-layer with the zero-centered point cloud. Using this feature, we decode to obtain

the residual translation $\mathbf{t}^*_{\mathbf{root}}$. The final translation of the root part is then computed as $\mathbf{t}_{\mathbf{root}} = \mathbf{t}^*_{\mathbf{root}} + \widetilde{\mathcal{P}}$.

Thus, the 6D pose of the root part is represented by the combination of rotation and translation, denoted as $T_{\mathbf{root}} = \{R_{\mathbf{root}}, \mathbf{t}_{\mathbf{root}}\}$.

5.2 Constrained Part Pose Estimation

Before proceeding to the second step of pose estimation for the constrained parts, we first canonicalize the input point cloud using the predicted pose of the root part, $T_{\mathbf{root}}$. This canonicalization process offers several advantages: (1) it transforms the joint state estimation task from camera space to a canonical space, which is more conducive to neural network learning, and (2) it mitigates the effects of varying joint configurations in the movable rigid parts, enabling the canonicalized point cloud to provide both shape and kinematic priors, significantly aiding the regression of joint states.

In the second stage of CAPER++, we predict the poses of constrained parts $R^{(k)}\mathbf{cons}k = 2^K$, along with their scales $\boldsymbol{s}^{(k)}\mathbf{cons}k = 2^K$, segmentation labels δ, and joint parameters $(\boldsymbol{u}^{(k)}, \boldsymbol{q}^{(k)})_{k=2}^K$. To this end, we construct an encoder-decoder architecture that partially shares parameters with the first stage. Specifically, we reuse the HS-Encoder and design four decoders composed of 1D convolutional layers, batch normalization, dropout, and ReLU activations. The network concludes with four parallel branches for predicting joint parameters, joint states, part scales, and segmentation labels.

5.2.1 Joint Parameter Branch

To recover the pose for each constrained part, we need to estimate the joint parameters $(\boldsymbol{u}, \boldsymbol{q})$. Note that our CAPER++ predicts joint parameters in the canonicalized articulation space rather than normalized coordinate space. Concretely, a *scale-aware heatmap voting scheme* is proposed, which is more accurate for the prediction of joint parameters (Fig. 6).

During training, we define the offset as the projected distance D_i for each point p_i from each canonicalized point p_i to the joint $(\boldsymbol{u}, \boldsymbol{q})$. The ground truth of offset can be formulated by Eq. 4, while heatmap h_i represents the likelihood of point p_i contributing to the joint. It is noted that accounting for the scale discrepancies among different parts and across different objects, we empirically restrict the computation of heatmap values to points whose off-axis distances do not exceed 0.2 times (Eq. 5) the farthest distance ($D_{max} = \max\{D_i\}_{i=1}^N$) , which serves as an approximate scale reference for each part.

$$D_i^* = \|(p_i - \boldsymbol{q}^*) \times \boldsymbol{u}^*\|_2 \tag{4}$$

$$h_i^* = \begin{cases} \exp\{-D_i^*\}, & \text{if } D_i^* < 0.2 \cdot D_{max} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

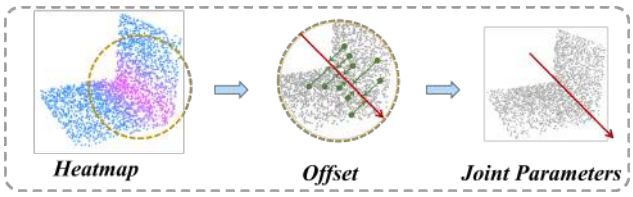

Fig. 6. **Illustration for Joint Parameter Prediction.** The heatmap uses color intensity to represent the likelihood of a point serving as a pivot—deeper red denotes higher probability, while deeper blue indicates lower probability.

During inference, the pivot prediction consists of two heads serving for offset prediction and heatmap prediction. The final predicted pivot is the weighted result by Eq. 6. In a similar spirit, the joint direction prediction consists of one head: a joint direction head that regresses the joint direction $\hat{\boldsymbol{u}}$ (as shown in Eq. 7) for both revolute and prismatic joints.

$$\hat{\boldsymbol{q}} = \frac{1}{N}\sum_{i=1}^{N} \hat{h}_i(p_i + \hat{D}_i) \tag{6}$$

$$\hat{\boldsymbol{u}} = \frac{1}{N}\sum_{i=1}^{N} \{\hat{\boldsymbol{u}}_i \cdot \hat{h}_i\} \tag{7}$$

Note that for prismatic joints, we only use the joint direction head since $\boldsymbol{q}$ is not defined for prismatic motion.

5.2.2 Joint State Branch

To estimate the poses of the constrained parts, we build a prediction branch that estimates the joint state $\boldsymbol{\theta}_r^{(k)}$ for each part. The joint state branch concludes with a 1-D convolution layer that outputs the final result, consisting of $K-1$ channels corresponding to the $K-1$ constrained parts in the articulated object. During inference for revolute joints, given the predicted joint parameters $\boldsymbol{\phi}_r^{(k)} = (\boldsymbol{u}_r^{(k)}, \boldsymbol{q}_r^{(k)})$ and joint states $\boldsymbol{\theta}_r^{(k)}$, we use the extended Rodrigues rotation formula to convert the joint state into the rotation matrix $R_{\mathbf{cons}}^{(k)}$ for the k-th part:

$$\begin{aligned} R_{\mathbf{cons}}^{'(k)} = &\cos\boldsymbol{\theta}_r^{(k)} U_r^{(k)} + (1 - \cos\boldsymbol{\theta}_r^{(k)})\cdot \\ &(U_r^{(k)} \cdot Q_r^{(k)})Q_r^{(k)} + \sin\boldsymbol{\theta}_r^{(k)} Q_r^{(k)} \end{aligned} \tag{8}$$

where $Q_r^{(k)}$ is a skew-symmetric matrix formed from the pivot point position $\boldsymbol{q}_r^{(k)}$, and $U_r^{(k)}$ is the normalized direction of the joint axis $\boldsymbol{u}_r^{(k)}$. The resulting matrix $R_{\mathbf{cons}}^{'(k)}$ is a 4×3 matrix, which is then combined with the row [0, 0, 0, 1] to form the homogeneous matrix $T_{\mathbf{cons}}^{'(k)}$, representing the pose of the k-th constrained part.

For prismatic joints, given the joint parameters $(\boldsymbol{u}_p^{(k)})$ and predicted joint state $\boldsymbol{\theta}_p^{(k)}$, we calculate the relative homogeneous matrix as the pose for the k-th part:

$$T_{\mathbf{cons}}^{'(k)} = \begin{bmatrix} \mathbf{I} & \boldsymbol{\theta}_p^{(k)}\boldsymbol{u}_p^{(k)} \\ 0 & 1 \end{bmatrix}$$

where $\mathbf{I}$ is the identity matrix. Finally, the poses of the constrained parts are obtained by $T_{\mathbf{cons}}^{(k)} = T_{\mathbf{root}} \cdot T_{\mathbf{cons}}^{'(k)}$.

5.2.3 Part Scale Branch

To predict the scale $\boldsymbol{s}^{(k)}$ for all parts of the articulated object, we employ a regression scheme in the part scale branch. This branch predicts the residual size $\mathring{\boldsymbol{s}}^{(k)} = \boldsymbol{s}^{(k)} - \widetilde{\boldsymbol{s}}^{(k)}$, where $\widetilde{\boldsymbol{s}}^{(k)}$ is the pre-defined mean size of the k-th part. The final scale is then determined by $\boldsymbol{s}^{(k)} = \mathring{\boldsymbol{s}}^{(k)} + \widetilde{\boldsymbol{s}}^{(k)}$.

In the articulated pose canonicalization module, the canonicalized point cloud $\mathcal{P}'$ is computed as the product of the inverse transformation $T_{\mathbf{root}}$ and the point cloud $\mathcal{P}^{(k)}$:

$$\mathcal{P}^{'(k)} = (T_{\mathbf{root}})^{-1}\mathcal{P}^{(k)} = (R_{\mathbf{root}})^{-1}(\mathcal{P}^{(k)} - \mathbf{t}_{\mathbf{root}})$$

By canonicalizing the input point cloud, the second-step regression model takes the canonicalized point cloud as input to estimate joint parameters and state. This increases the sensitivity of

the neural network to the joint parameters and state, significantly improving regression performance.

5.3 Extending To Pose Tracking Task

In our conference version [1], we proposed the EfficientCAPER method, which centers on a decoupled formulation of pose representation and articulated object modeling. The decoupled rotation representation significantly avoids the computational overhead of dense predictions [11], [12] in conventional approaches, thereby enabling real-time inference at up to 50 FPS. This efficiency makes the method particularly well-suited for tasks such as pose tracking, where fast inference is critical.

Regarding articulated pose tracking, most existing methods adopt a *frame-by-frame* formulation, which often incurs computational inefficiency due to redundant per-frame inference. Several approaches [13], [68], [69] attempt to exploit temporal coherence by incorporating inter-frame information. However, their formulations typically rely on Euclidean pose parameterizations. This choice gives rise to two inherent limitations: (i) geometrically invalid poses may emerge during optimization, such as non-orthogonal rotation matrices, and (ii) singularities associated with parameterized rotations, including gimbal lock, remain unavoidable.

From a geometric standpoint, articulated pose tracking is intrinsically defined on the SE(3) manifold, where rigid motions naturally preserve pose validity. This observation motivates us to formulate articulated pose tracking as incremental motion estimation on SE(3) via its Lie algebra. By representing inter-frame pose variations in the tangent space, the tracking problem admits locally linear updates while maintaining global manifold constraints through the exponential map. As a result, pose validity is guaranteed by construction, and pose increments can be accumulated in a numerically stable manner over time. This manifold-aware formulation constitutes the theoretical foundation of our tracking design.

5.3.1 Proxy Canonicalization

In the task of single-frame pose estimation, we typically refer to the rest pose in the canonical space and predict the absolute pose of the current frame relative to its rest pose. However, to switch from predicting absolute poses to predicting pose increments, our key idea is to transform the point cloud of the current frame and incorporate information from the previous frame to achieve incremental prediction. Details can be found in Fig. 7.

Since the predicted poses are all relative to the rest pose, we first transform the current frame $\mathcal{P}_t^{(k)}$ (gray object in Fig. 7) to the proxy canonical space (blue object in Fig. 7) using the inverse pose matrix $(T_{t-1}^{(k)})^{-1}$ from previous frame (orange object in Fig. 7). This process can be formulated as Eq.9:

$$\ddot{\mathcal{P}}_t^{(k)} = (T_{t-1}^{(k)})^{-1} \cdot \mathcal{P}_t^{(k)} = (R_{t-1}^{(k)})^{-1}(\mathcal{P}_t^{(k)} - \mathbf{t}_{t-1}^{(k)}). \quad (9)$$

We refer to this transformation as *proxy canonicalization*. The point cloud $\ddot{\mathcal{P}}_t^{(k)}$ after proxy canonicalization is termed as the point cloud in the *proxy canonical space*.

Based on this, the incremental pose $\Delta T_t^{(k)}$ between $t-1$ frame and t frame can be determined (blue dotted line in Fig. 7), whose ground truth can be formulated as:

$$\Delta T_t^{(k)*} = (T_{t-1}^{(k)})^{-1} \cdot T_t^{(k)} \quad (10)$$

By this means, the predicted pose is considered to be equivalent to the pose increment between the previous frame and the current frame. Therefore, the t frame's point cloud can be formulated by:

$$\mathcal{P}_t^{(k)} = \Delta\hat{T}_t^{(k)} \cdot \mathcal{P}_{t-1}^{(k)} \quad (11)$$

5.3.2 Pose Increment Learning on SE(3) Manifolds

The Transformation in the SE(3) Tangent Space. On the SE(3) manifolds, the Eq. 10 and Eq. 11 can be reformulated via Lie algebra:

$$(\Delta\xi_t^{(k)*})^\wedge = \{(\xi_{t-1}^{(k)})^{-1}\}^\wedge \otimes (\xi_t^{(k)})^\wedge, \quad \xi^\wedge \in \mathfrak{se}(3) \quad (12)$$

$$\mathcal{P}_t^{(k)} = (\Delta\hat{\xi}_t^{(k)})^\wedge \otimes \mathcal{P}_{t-1}^{(k)} \quad (13)$$

where $(\xi_t^{(k)})^\wedge$ is the Lie algebra representation of the t frame's pose and it is in the $\mathfrak{se}(3)$ domain which is the tangent space of SE(3) manifolds, $\otimes$ denotes the rigid transformation of the point cloud under Lie algebra $(\Delta\xi_t^{(k)})^\wedge$.

Mathematically, Eq. 10 and Eq. 12 can be unified under the exponential map operation and formulated as an optimization problem on the tangent space of SE(3), aiming to estimate the optimal relative transformation.

$$\Delta T_t^{(k)} = \exp\left\{(\Delta\xi_t^{(k)})^\wedge\right\} = \begin{bmatrix} R_t^{(k)} & \mathbf{t}_t^{(k)} \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3) \quad (14)$$

Here, $(\Delta\xi_t^{(k)})^\wedge$ is parameterized in the tangent space of SE(3), represented as $(\Delta\xi_t^{(k)})^\wedge = (w^\wedge, \rho)^T \in \mathfrak{se}(3)$, where $w \in \mathbb{R}^3$ and $\rho \in \mathbb{R}^3$ correspond to the rotational and translational components, respectively. R is the rotation matrix derived from $w^\wedge$. $\mathbf{t}$ is the translation vector derived from ρ. This formulation ensures that the pose update strictly adheres to the manifold constraints, avoiding degenerate solutions that violate rigid-body motion properties.

Dynamic Keyframe Selection. In articulated object tracking tasks, only the pose of the initial frame is available as ground truth. When subsequent frames rely solely on the estimated pose from the preceding frame, prediction errors tend to accumulate over time, leading to significant degradation in tracking accuracy. To mitigate this issue, we propose a *keyframe-based strategy* that explicitly suppresses error accumulation. In this framework, poses of keyframes are obtained through direct estimation, while poses of non-keyframes are predicted via learned inter-frame pose increments, conditioned on the prior pose of the most recent keyframe. Specifically, keyframes are selected at fixed intervals L, corresponding to frame indices $0, L, 2L, \ldots$. The pose increment for each non-keyframe is then computed relative to its corresponding keyframe, rather than the immediately preceding frame, as formulated in Eq. 15. Within each segment defined by consecutive keyframes, tracking is performed incrementally. When transitioning between segments, the system is re-initialized using the directly estimated keyframe pose. This design effectively constrains error propagation, enhancing both tracking stability and system robustness over long sequences.

$$\exp\left\{(\Delta\xi_t^{(k)})^\wedge\right\} = \left(T_{t-L}^{(k)}\right)^{-1} \cdot T_t^{(k)} \quad (15)$$

where $T_{t-L}^{(k)}$ and $T_t^{(k)}$ denote the keyframe and current frame poses, respectively.

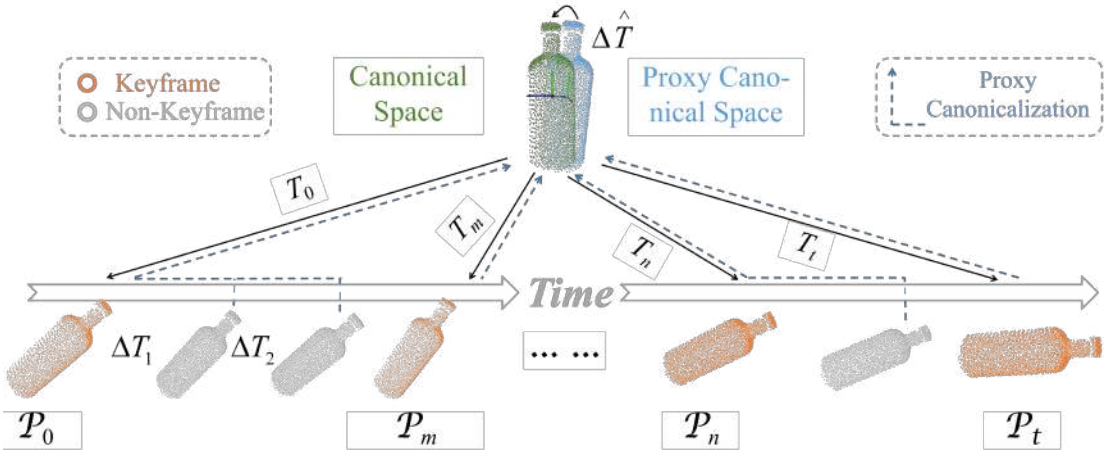

Fig. 7. **The Illustration of Proxy Canonicalization**. The objects in blue are in proxy canonical space, while the objects in green are in canonical space. The orange is objects in the keyframe, The objects in orange and gray are utilized to indicate the camera space. The solid black lines signify the transformation from camera space to the canonical space. The dashed blue lines illustrate the transformation from camera space to the proxy canonical space. It is important to note that, during keyframes, proxy canonicalization is equivalent to canonicalization. Note that all the ΔT is the pose increment reference to its keyframe.

In practical applications, we observe that the strategy for selecting keyframes has a substantial impact on pose tracking accuracy. Although selecting keyframes at fixed intervals can improve tracking stability to some extent, this approach suffers from inherent limitations: when the interval is too large, incremental errors within a segment tend to accumulate; conversely, if the interval is too small, the method degrades into frame-by-frame independent prediction, failing to leverage temporal redundancy effectively. To address this issue, we propose a dynamic keyframe selection mechanism that allows the model to adaptively determine the keyframe interval based on the current tracking state. While we continue to denote the keyframe interval as L, it is no longer a fixed constant but is dynamically adjusted by the model. Specifically, we design an energy function that quantifies the effectiveness of the current keyframe by jointly considering the predicted pose and the actual observed point cloud.

Mathematically, keyframe selection is determined by minimizing an energy function $\mathfrak{E}$.

$$\mathfrak{E}_t = \frac{1}{K}\Sigma_{k=1}^{K}\{D(\mathcal{P}_t^{(k)}, \quad \Delta\hat{T}_t^{(k)} \cdot \mathcal{P}_{t-L}^{(k)})\} \tag{16}$$

Here, $\Delta\hat{T}_t^{(k)}$ is specially used for the predicted SE(3) transformation between $t-L$ and t frame. $\mathcal{P}_t^{(k)}$ and $\mathcal{P}_{t-L}^{(k)}$ represent the observed point cloud of t frame and $t-1$ frame. $D(\cdot,\cdot)$ is the chamfer distance error. In practice, the energy exceeds a predefined threshold $\phi = 0.01$, which indicates degraded tracking performance, prompting an immediate keyframe update to maintain overall prediction stability and accuracy.

The overall articulation tracking procedure for the frame stream is summarized in Algorithm 1.

Algorithm 1 The Pipeline of CAPER++ Applied in Pose Tracking Task.

Input: The frame stream $\{\mathcal{P}_t^{(k)}\}_{t\geq 0}$ of all the k parts in the camera space. The pose of initial frame $T_0^{(k)}$ which can be mapped to Lie algebra element $(\xi_0^{(k)})^\wedge$.
Output: Per-part pose $(\hat{T}_t^{(k)})$ and scale $\hat{s}_t^{(k)}$ for all the $t>0$ frames.
for each frame $\mathcal{P}_t^{(k)} \in \{\mathcal{P}_t^{(k)}\}_{t>0}$ **do**
  Proxy canonicalization process $\left\{(\xi_{t-L}^{(k)})^\wedge\right\}^{-1} \otimes \mathcal{P}_t^{(k)}$.
  Compute pose increment $(\Delta\hat{\xi}_t^{(k)})^\wedge$ and scales $\hat{\mathbf{s}}_t^{(k)}$.
  Accumulate Lie algebra element increments $(\hat{\xi}_t^{(k)})^\wedge = (\hat{\xi}_{t-L}^{(k)})^\wedge + (\Delta\hat{\xi}_t^{(k)})^\wedge$.
  Map the Lie algebra element into Lie group $\hat{T}_t^{(k)} = exp\left\{(\hat{\xi}_t^{(k)})^\wedge\right\}$.
  if Energy function $\mathfrak{E}_t < \phi$ **then**
    Update keyframe.
  end if
  Output the predicted pose $\hat{T}_t^{(k)}$ and scale $\hat{\mathbf{s}}_t^{(k)}$ for the current frame.
end for

5.4 Loss Function

As previously discussed, pose tracking is closely related to pose estimation. While estimation regresses the absolute pose in the camera coordinate system, tracking employs a proxy canonicalization strategy that transforms the current frame's point cloud into the reference keyframe's coordinate system, enabling prediction of the relative pose increment. Both tasks thus share a consistent loss formulation, differing only in supervision: estimation uses absolute pose labels, whereas tracking relies on relative pose increments. In essence, pose tracking can be viewed as conditional pose estimation anchored to a specific keyframe.

For step I, the L1 loss function is applied to supervise the (r_m, r_n) vector and translation $\mathbf{t}$ regression:

$$\mathcal{L}_{rot} = \|r_m - \hat{r}_m\|_1 + \|r_n - \hat{r}_n\|_1, \mathcal{L}_{trans} = \left\|t - \hat{t}\right\|_1 \tag{17}$$

where m,n represent the possible selected direction vectors. To train the correction factor ψ that is used for (r_m, r_n) voting, the loss function $\mathcal{L}_{conf}$ is defined as:

$$\mathcal{L}_{conf} = \sum_{\substack{i\in\{m,n\}\\ j\in\{1,2\}}} \left\|\psi_j - \exp(|r_i - \hat{r}_i|^2)\right\|_1 \tag{18}$$

For step II, we utilize four loss functions to supervise the joint parameter, joint state, part scale, and part segmentation branches. Firstly for joint parameter branch, the loss function $\mathcal{L}_{para}$ aims to

optimize the projection distance from predicted joint location $\hat{\boldsymbol{q}}^{(k)}$ to joint $(\boldsymbol{u}^{(k)}, \boldsymbol{q}^{(k)})$:

$$\mathcal{L}_{loc} = \frac{1}{K-1}\sum_{k=2}^{K} \mathrm{Proj}(\hat{\boldsymbol{q}}^{(k)}, \boldsymbol{u}^{(k)}, \boldsymbol{q}^{(k)}) \tag{19}$$

and we use the L1 loss to supervise the joint direction $\mathcal{L}_{dir}$. In terms of the joint state branch, since the joint states in the annotation can be positive or negative, directly regressing these joint states is quite difficult. So we design the joint state loss function $\mathcal{L}_{stat}$ with a positive penalty factor $\alpha = 20$:

$$\mathcal{L}_{stat} = \begin{cases} \frac{1}{K-1}\sum\limits_{k=2}^{K} \|\boldsymbol{\theta}^{(k)} - \hat{\boldsymbol{\theta}}^{(k)}\|_1, & \text{if} \quad \boldsymbol{\theta}^{(k)} \cdot \hat{\boldsymbol{\theta}}^{(k)} > 0 \\ \frac{1}{K-1}\sum\limits_{k=2}^{K} \alpha\|\boldsymbol{\theta}^{(k)} - \hat{\boldsymbol{\theta}}^{(k)}\|_1, & \text{if} \quad \boldsymbol{\theta}^{(k)} \cdot \hat{\boldsymbol{\theta}}^{(k)} < 0 \end{cases} \tag{20}$$

In this loss function, when the joint state of the k-th joint is opposite to the ground truth in annotation, we apply a penalty factor α to the loss of that particular joint which encourages the neural network to learn more correctly. Next, we employ L1 loss for part scale branch $\mathcal{L}_{sca}$ and Cross-Entropy loss for part segmentation branch $\mathcal{L}_{seg}$.

In particular, for the pose tracking task, we use an additional L1 loss to supervise the $\mathfrak{se}(3)$ Lie algebra:

$$\mathcal{L}_{se} = \|\rho - \hat{\rho}\|_1 + \|w - \hat{w}\|_1 \tag{21}$$

Finally, the total training losses $\mathcal{L}_I$ for step I and $\mathcal{L}_{II}$ for step II are both the weighted sum of the losses from root part pose estimation and constrained part pose estimation modules:

$$\mathcal{L}_I = \lambda_{rot}\mathcal{L}_{rot} + \lambda_{trans}\mathcal{L}_{trans} + \lambda_{conf}\mathcal{L}_{conf} + \lambda_{se}\mathcal{L}_{se} \tag{22}$$

$$\mathcal{L}_{II} = \lambda_{loc}\mathcal{L}_{loc} + \lambda_{dir}\mathcal{L}_{dir} + \lambda_{stat}\mathcal{L}_{stat} + \lambda_{sca}\mathcal{L}_{sca} + \lambda_{seg}\mathcal{L}_{seg} \tag{23}$$

where the weight multiplication factors $\lambda_{rot}, \lambda_{trans}, \lambda_{conf}, \lambda_{loc}, \lambda_{dir}, \lambda_{stat}, \lambda_{sca}, \lambda_{seg}, \lambda_{se}$ are set to be 8.0, 10.0, 1.0, 1.0, 1.0, 6.0, 1.0, 0.1, 1.0. The optimal hyperparameters are mined by a cross-validation method.

6 Experiments

6.1 Experimental Settings

6.1.1 Datasets

Datasets For Pose Estimation. Following [1], we evaluate CAPER++ on three public datasets ranging from synthetic, semi-synthetic, and real-world scenarios: ArtImage [14], ReArt-Mix [33], and RobotArm [33]:

(1) Synthetic Dataset: ArtImage is a synthetic articulated object estimation dataset derived from PartNet Mobility [70]. This dataset includes five categories of articulated objects: Laptops, Eyeglasses, Dishwashers, Scissors, and Drawers.

(2) Semi-synthetic Dataset: ReArtMix. To further validate the effectiveness of our method on semi-synthetic dataset, we also assess CAPER++ on the ReArtMix dataset. ReArtMix is a semi-synthetic dataset specifically designed for articulated object estimation tasks, combining real-world scenes with synthetic objects.

(3) Real-world Scenario: RobotArm. To demonstrate the generality and robustness of our method in real-world scenarios, we train and test CAPER++ on the 7-part RobotArm dataset. The RobotArm dataset provides real-world data of robotic arm movements, which is crucial for assessing the model's performance in practical applications.

Datasets For Pose Tracking. Following the setup in [13], we construct a synthetic dataset for category-level articulated object pose tracking based on the PartNet-Mobility dataset [70]. Since the original work did not release the data generation scripts or detailed configuration parameters, we were unable to directly reproduce its dataset. To ensure the scientific rigor and comparability of our experimental settings, we rebuild a synthetic tracking dataset using a similar strategy, referred to as PM-Videos. In addition, to comprehensively evaluate the spatiotemporal consistency and generalization ability of the proposed model under challenging conditions such as motion continuity, self-occlusion, and sensor noise, we further introduce two more demanding datasets: ReArt-Videos, constructed from semi-synthetic scenes provided by [33], and RobotArm-Videos, collected in real-world settings. These datasets span a range of scenarios from simulation to reality, enabling systematic evaluation of the proposed method's robustness and adaptability across diverse environments.

(1) Synthetic Dataset: PM-Videos. Firstly, we introduce a synthetic articulated object tracking dataset, PM-Videos, based on objects from PartNet-Mobility [70]. The dataset is generated using real-time rendering in the Unity engine and includes five object categories—laptop, eyeglasses, dishwasher, scissors, and drawer, as referenced in ArtImage. Each category contains 300 videos and approximately 9K frames. Each video contains approximately 30 frames. For training, we use 270 videos per category, while the remaining 30 videos are reserved for testing. The video generation process is as follows: mesh models from PartNet-Mobility serves as the input geometry, and joint angles are progressively adjusted in ascending order within Unity to simulate smooth object articulation. During this process, we record synchronized multi-modal outputs, including joint configurations, depth maps (640×640), and RGB images (640×640) for each frame. The range of joint motion is constrained by the physical properties of real-world objects; for instance, joint angles span from -90° to 90° for laptops and from -90° to 0° for eyeglasses. This monotonic adjustment ensures temporal consistency and results in coherent motion sequences for each articulated object.

(2) Semi-synthetic Dataset: ReArt-Videos. Secondly, we construct a semi-synthetic dataset for articulated object tracking based on the ReArt-48 repository, using the SAMERT technique [33], which yields over 100 videos per category. Similar to PM-Videos, each video in ReArt-Videos consists of approximately 30 frames. It includes five categories—box, cutter, drawer, scissor, and stapler. For training, we use 80 videos per category, while the remaining 20 Videos are reserved for testing. Besides, in our dataset, joint configurations, depth maps (1280×720), and RGB images (1280×720) were recorded.

(3) Real-world Scenario: RobotArm-Videos. This dataset is derived from [33]. It is worth noting that the RobotArm dataset for the static pose estimation task is constructed by extracting static frames from the video streams collected in [33], and is primarily used for static pose estimation tasks. In contrast, the RobotArm-Videos dataset preserves the original continuous video sequences, making it suitable for evaluating the performance of pose tracking algorithms in dynamic scenarios. Each video in RobotArm-Videos consists of 200 frames. For training, we use 80 videos per category, while the remaining 20 videos are reserved for testing. Besides, in

TABLE 1
Comparison with State-of-the-arts on the ArtImage Dataset. We validate our CAPER++ on categories Laptop, Eyeglasses, Dishwasher, Scissors and Drawer that contains 2, 3, 2, 2, 4 parts respectively. ↓ means the lower the better and ↑ means the upper the better.

Category	Method	Per-part Pose			Inference Time per Image (s) ↓
		Rotation Error (°) ↓	Translation Error (m) ↓	3D IOU (%) ↑	
Laptop	A-NCSH [11]	5.3, 5.4	0.054, **0.043**	56.7, 40.2	9.00
	ContactArt [27]	5.2, 4.1	0.058, 0.059	45.5, 30.1	1.51
	ArtPERL [71]	4.9, 4.7	**0.053**, 0.066	64.6, 50.4	0.89
	EfficientCAPER [1]	3.5, 4.0	0.065, 0.071	88.2, 87.9	0.02
	Lie-X [56]	5.1, 5.3	0.068, 0.069	65.3, 53.5	2.33
	ScorePose [57]	3.6, 4.2	0.068, 0.068	83.4, 84.3	0.54
	CAPER++ (Ours)	**3.1**, **2.7**	0.063, 0.064	**88.4**, **88.3**	**0.02**
Eyeglasses	A-NCSH [11]	3.7, 22.3, 23.2	0.049, 0.313, 0.324	52.5, 40.2, 39.6	11.91
	ContactArt [27]	4.8, 7.3, 7.2	0.061, 0.105, 0.321	23.8, 28.5, 27.4	2.03
	ArtPERL [71]	4.1, 6.2, 6.0	0.047, **0.095**, 0.091	58.6, 46.5, 51.7	1.01
	EfficientCAPER [1]	3.1, 7.2, 5.8	0.038, 0.109, 0.089	91.8, 82.3, 84.3	0.02
	Lie-X [56]	4.5, 8.5, 9.1	0.059, 0.123, 0.134	61.8, 48.9, 53.6	2.74
	ScorePose [57]	3.2, 6.9, 6.0	0.041, 0.111, 0.098	90.1, 81.3, 81.2	0.63
	CAPER++ (Ours)	**2.8**, **5.4**, **5.5**	**0.036**, 0.103, **0.085**	**92.0**, **84.8**, **86.3**	**0.02**
Dishwasher	A-NCSH [11]	4.0, 4.8	0.059, 0.123	84.3, 56.2	5.48
	ContactArt [27]	5.8, 6.0	0.104, 0.138	67.5, 40.2	1.32
	ArtPERL [71]	3.9, 4.3	0.055, 0.079	89.3, 67.6	0.93
	EfficientCAPER [1]	2.5, 3.1	0.052, 0.085	91.2, 83.9	0.02
	Lie-X [56]	4.3, 4.9	0.067, 0.099	81.3, 82.3	2.34
	ScorePose [57]	2.9, 3.3	0.058, 0.080	89.3, 84.5	0.53
	CAPER++ (Ours)	**2.3**, **2.8**	**0.050**, **0.073**	**92.8**, **90.2**	**0.02**
Scissors	A-NCSH [11]	**2.0**, 2.9	0.035, **0.025**	46.5, 44.8	6.54
	ContactArt [27]	3.7, 3.3	0.042, 0.036	39.6, 40.2	1.31
	ArtPERL [71]	2.2, **2.6**	**0.031**, 0.042	40.9, 46.3	0.84
	EfficientCAPER [1]	5.0, 5.1	0.048, 0.099	74.2, 69.2	0.02
	Lie-X [56]	4.8, 4.9	0.053, 0.098	42.3, 48.3	2.32
	ScorePose [57]	4.8, 5.0	0.050, 0.095	72.3, 68.2	0.71
	CAPER++ (Ours)	4.3, 4.4	0.044, 0.090	**77.1**, **70.4**	**0.02**
Drawer	A-NCSH [11]	2.8, 3.5, 3.9, 2.9	0.045, 0.155, 0.157, 0.075	90.2, 81.5, 78.4, 82.7	16.52
	ContactArt [27]	4.2, 4.2, 4.2, 4.2	0.118, 0.139, 0.132, 0.105	76.8, 75.4, 77.2, 72.3	1.51
	ArtPERL [71]	3.5, 3.5, 3.5, 3.5	0.061, 0.112, 0.121, 0.104	84.8, 78.6, 79.0, 81.2	1.14
	EfficientCAPER [1]	2.0, 2.0, 2.0, 2.0	0.041, 0.086, 0.090, 0.080	93.2, 87.1, 87.1, 88.4	0.02
	Lie-X [56]	3.4, 3.7, 3.7, 3.4	0.063, 0.121, 0.135, 0.109	83.5, 77.5, 78.3, 82.1	1.03
	ScorePose [57]	1.9, 1.9, 1.9, 1.9	0.048, 0.101, 0.099, 0.083	91.2, 85.3, 85.1, 84.6	3.69
	CAPER++ (Ours)	**1.3**, **1.3**, **1.3**, **1.3**	**0.040**, **0.079**, **0.081**, **0.064**	**93.4**, **88.3**, **88.8**, **90.5**	**0.02**

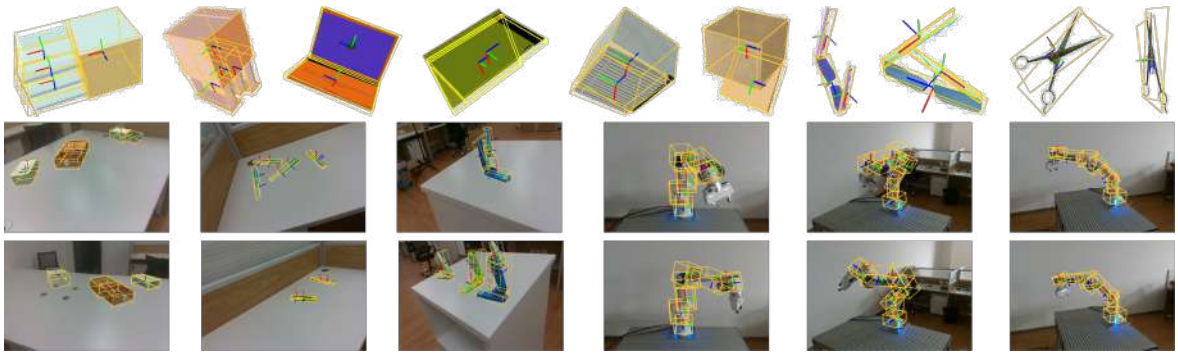

Fig. 8. **Qualitative Results on ArtImage (top), ReArtMix (bottom left), and RobotArm (bottom right) Dataset.**

our dataset, joint configurations, depth maps (640×480), and RGB images (640×480) were recorded.

6.1.2 Metrics, Implementation Details and Baselines

To validate the performance of the CAPER++, we adopt degree error (°) for 3D rotation, distance error (m) for 3D translation, 3D IOU (%) for scale, and per-image inference time for estimating speed. During the data pre-processing, input point clouds are sampled into 1,024 points as the network inputs. We have adopted the Ranger optimizer and the "flat and anneal" learning rate strategy. CAPER++ is trained from scratch rather than being initialized from the pre-trained weights of EfficientCAPER and the total training epoch is 250. We put two trained models in the same script for inference to directly obtain per-part pose. All the experiments are implemented on an NVIDIA GeForce RTX 3090 GPU with 24GB memory. Note that the metrics used for tracking evaluation are the same.

In this study, we select the following methods as comparative baselines: A-NCSH [11], ContactArt [27], CAPTRA [13], GAPS [72], ArtPERL [71], Lie-X [56], and ScorePose [57]. Specifically, A-NCSH [11] is the earliest representative approach proposed for category-level articulation pose estimation. CAPTRA [13] extends this line of work to pose tracking, serving as a foundational baseline for the tracking task. Meanwhile, ContactArt [27], GAPS [72], and ArtPERL [71] are recent state-of-the-art methods that reflect the latest advances in this field. Besides, Lie-X [56] and ScorePose [57] are classical and strong baselines based on Lie algebra. To ensure a fair comparison, we apply minimal task-specific adaptations to certain methods that are not directly compatible with our evaluation setting, while preserving their core design principles.

6.2 Performance on Pose Estimation Task

(1) Comparison of SOTA Methods on Synthetic Scenarios. We evaluate CAPER++ on the synthetic ArtImage dataset, which comprises articulated objects derived from PartNet-Mobility. Quantitative results are reported in Tab. 1. Overall, CAPER++

TABLE 2
Pose Estimation Results on ReArtMix Dataset.

Category	Method	Per-part Pose		
		Rotation Error (°)	Translation Error (m)	3D IOU(%)
Box	ReArtNet [33]	3.4, 3.6	0.025, 0.034	52.1, 49.3
	ScorePose [57]	2.7, 2.5	0.018, 0.027	75.3, 56.8
	CAPER++	**0.8**, **1.5**	**0.004**, **0.012**	**96.8**, **90.1**
Stapler	ReArtNet [33]	4.3, 5.9	0.029, 0.027	38.1, 35.5
	ScorePose [57]	3.7, 4.8	0.021, 0.023	67.8, 66.3
	CAPER++	**0.9**, **1.4**	**0.006**, **0.018**	**94.8**, **87.5**
Cutter	ReArtNet [33]	3.0, 3.2	0.016, 0.017	35.5, 34.2
	ScorePose [57]	2.6, 2.6	0.013, 0.008	75.3, 66.8
	CAPER++	**1.5**, **1.5**	**0.003**, **0.004**	**97.2**, **95.9**
Scissors	ReArtNet [33]	5.6, 5.3	0.011, 0.013	26.3, 24.3
	ScorePose [57]	3.7, 4.8	0.018, 0.014	53.5, 49.8
	CAPER++	**2.0**, **3.9**	**0.005**, **0.012**	**93.5**, **89.2**
Drawer	ReArtNet [33]	3.1, 3.2	0.021, 0.019	51.5, 50.2
	ScorePose [57]	2.1, 2.0	0.017, 0.015	68.3, 69.7
	CAPER++	**0.7**, **0.7**	**0.006**, **0.007**	**95.3**, **94.2**

TABLE 3
Pose Estimation Results on 7-part RobotArm Dataset.

Per-part Rotation Error (°)							
Part ID	1	2	3	4	5	6	7
A-NCSH [11]	7.8	7.9	10.3	10.5	11.2	16.4	23.5
ScorePose [57]	2.3	6.2	8.3	8.9	9.5	11.5	16.3
CAPER++	**0.04**	**4.9**	**6.8**	**6.5**	**7.2**	**8.8**	**11.7**
Per-part Translation Error (m)							
Part ID	1	2	3	4	5	6	7
A-NCSH [11]	**0.012**	0.044	0.067	0.066	0.079	0.236	0.403
ScorePose [57]	0.015	0.032	0.052	0.056	0.084	0.138	0.263
CAPER++	0.013	**0.018**	**0.039**	**0.047**	**0.070**	**0.109**	**0.194**

TABLE 4
Pose Tracking Results on ReArt-Videos Dataset.

Category	Method	Per-part Pose		
		Rotation Error (°)	Translation Error (m)	3D IOU(%)
Box	ReArtNet [33]	9.3, 9.5	0.041, 0.051	41.3, 38.4
	ScorePose [57]	6.8, 6.9	0.032, 0.037	66.8, 68.3
	CAPER++	**4.6, 4.7**	**0.008, 0.007**	**87.3, 80.1**
Stapler	ReArtNet [33]	11.2, 11.8	0.045, 0.047	27.1, 24.8
	ScorePose [57]	7.8, 7.9	0.032, 0.033	45.3, 47.6
	CAPER++	**5.6, 6.5**	**0.007, 0.007**	**83.1, 79.3**
Cutter	ReArtNet [33]	10.1, 10.3	0.033, 0.034	24.6, 23.8
	ScorePose [57]	7.1, 7.0	0.027, 0.031	51.3, 50.7
	CAPER++	**4.9, 4.9**	**0.017, 0.029**	**88.6, 83.2**
Scissors	ReArtNet [33]	13.8, 14.9	0.030, 0.033	17.3, 15.3
	ScorePose [57]	11.2, 13.1	0.025, 0.026	38.9, 42.5
	CAPER++	**8.3, 10.7**	**0.005, 0.005**	**86.3, 81.3**
Drawer	ReArtNet [33]	13.5, 13.5	0.042, 0.038	42.3, 42.6
	ScorePose [57]	10.5, 10.8	0.032, 0.034	61.8, 63.7
	CAPER++	**8.3, 8.3**	**0.022, 0.019**	**89.3, 85.1**

TABLE 5
Pose Tracking Results on RobotArm-Videos Dataset.

Per-part Rotation Error (°)							
Part ID	1	2	3	4	5	6	7
A-NCSH [11]	9.9	11.2	16.8	23.5	29.3	35.3	41.5
ScorePose [57]	2.9	4.5	8.9	13.5	17.5	21.5	25.3
CAPER++	**0.09**	**0.78**	**5.7**	**7.6**	**12.3**	**16.8**	**19.3**
Per-part Translation Error (m)							
Part ID	1	2	3	4	5	6	7
A-NCSH [11]	0.010	0.041	0.062	0.065	0.129	0.198	0.378
ScorePose [57]	0.008	0.029	0.058	0.065	0.093	0.141	0.198
CAPER++	**0.002**	**0.016**	**0.043**	**0.061**	**0.089**	**0.102**	**0.139**

achieves either the best or second-best rotation accuracy across the majority of object categories. In particular, on the *Dishwasher* category, our method attains rotation errors as low as 2.3° and **2.8**°, indicating strong robustness under articulated motion. Compared with representative Lie-algebra-based approaches (e.g., Lie-X [56]), CAPER++ reduces the rotation error by approximately **2**° and the translation error by **0.017**m. This improvement is primarily attributed to the proposed joint-centric representation, which effectively mitigates the adverse effects of self-occlusion. For the *Drawer* category, CAPER++ also exhibits superior translation accuracy, achieving errors of **0.040**m, **0.079**m,**0.081**m, and **0.064**m under different configurations. These results demonstrate the effectiveness of our framework in precisely estimating the state and orientation of prismatic joints. In addition, CAPER++ consistently improves 3D scale estimation, yielding higher 3D IoU scores than baseline methods across nearly all categories. From an efficiency perspective, CAPER++ eliminates conventional post-processing pipelines that rely on per-point NOCS prediction followed by Umeyama alignment and RANSAC-based refinement [11], [12]. Instead, object poses are directly regressed in a single forward pass, without auxiliary optimization or retrieval procedures. As a result, the proposed method supports real-time inference with an average runtime of approximately **0.02** s per frame. Qualitative comparisons are provided in Fig. 8.

(2) Generalization Capacity on Semi-Synthetic Scenarios. We further evaluate CAPER++ on ReArtMix, a dataset consisting of semi-synthetic articulated object scenarios. Quantitative results are summarized in Tab. 2. Compared with representative baseline methods, including ReArtNet [33] and ScorePose [57], CAPER++ consistently outperforms prior approaches across all object categories, with particularly notable gains in rotation accuracy. For the Cutter category, our method achieves highly precise translation estimation, yielding errors as low as 0.003m and 0.004m, respectively. These results further validate the effectiveness of the proposed framework in handling articulated structures under semi-synthetic settings. Qualitative comparisons are provided in Fig. 8.

(3) Generalization Capacity on Real-world Scenarios. To assess performance under real-world conditions, we further train and evaluate CAPER++ on the 7-part RobotArm dataset. This benchmark introduces additional challenges arising from the robot's multi-depth kinematic structure, where estimation errors tend to accumulate and propagate from the base to the end effector. Despite these difficulties, CAPER++ demonstrates strong robustness and stable accuracy across objects with diverse structural complexity, as evidenced by the quantitative results in Tab. 3 and the qualitative comparisons in Fig. 8.

6.3 Performance on Pose Tracking Task

(1) Comparison of SOTA Methods on Synthetic Scenarios. The proposed CAPER++ yields substantial gains in articulated object part pose tracking. As summarized in Tab. 6, it achieves state-of-the-art performance on both primary evaluation metrics, namely rotation and translation errors. For complex articulated objects such as *Dishwasher*, CAPER++ reduces the base-part rotation error to **3.5**° and **4.6**°, corresponding to a relative improvement of **34.3%–43.5%** over the strongest baseline, CAPTRA. Translation accuracy is also consistently improved, with an average error of **0.065**m. Compared with a representative Lie-algebra-based method (Lie-X [56]), CAPER++ further reduces the rotation error by approximately **1.8**° and the translation error by **0.035**m, highlighting its superior pose estimation accuracy. In terms of spatial alignment, CAPER++ achieves strong 3D Intersection-over-Union performance, attaining the best average IoU of **82.8%** on the *Drawer* category, substantially outperforming prior methods. The notable reduction in rotation error is primarily attributed to the proposed *vertical correction mechanism*, which enhances rotational stability under occlusion and kinematic constraints. From an efficiency perspective, the lightweight design of CAPER++ enables real-time inference at **50.0** FPS, delivering a **40**×**–50**× speedup over traditional optimization-based trackers such as ContactArt. These results demonstrate that CAPER++ effectively balances accuracy and efficiency, making it well-suited for real-world articulated object tracking. Qualitative results are provided in Fig. 9.

TABLE 6
Comparison with State-of-the-arts on the PM-Videos Dataset. We validate our CAPER++ on categories Laptop, Eyeglasses, Dishwasher, Scissors, and Drawer that contains 2, 3, 2, 2, 4 parts respectively. ↓ means the lower the better and ↑ means the upper the better.

Category	Method	Per-part Pose			Frame Per Second (FPS) ↑
		Rotation Error (°) ↓	Translation Error (m) ↓	3D IOU (%) ↑	
Laptop	A-NCSH [11]	8.5, 9.2	0.084, 0.103	40.8, 28.3	0.6
	ContactArt [27]	10.8, 16.2	0.131, 0.174	30.2, 17.3	1.4
	CAPTRA [13]	5.9, 5.3	0.080, 0.063	70.1, 43.5	10.0
	GAPS [72]	8.7, 8.9	0.092, 0.096	54.3, 39.3	2.9
	Lie-X [56]	9.3, 13.2	0.121, 0.153	47.3, 35.3	0.4
	ScorePose [57]	6.3, 6.8	0.088, 0.075	66.5, 40.9	2.0
	CAPER++ (Ours)	**4.3**, **5.0**	**0.063**, **0.058**	**74.2**, **51.3**	**50.0**
Eyeglasses	A-NCSH [11]	7.6, 24.8, 26.6	0.079, 0.324, 0.319	40.5, 29.3, 28.4	0.4
	ContactArt [27]	14.3, 26.5, 29.6	0.154, 0.198, 0.196	17.3, 13.6, 12.6	1.0
	CAPTRA [13]	4.5, 12.6, 13.1	0.054, 0.097, 0.084	53.1, 41.2, 39.8	7.1
	GAPS [72]	8.5, 9.3, 9.6	0.105, 0.123, 0.118	48.2, 36.7, 35.6	1.2
	Lie-X [56]	10.3, 15.3, 18.3	0.137, 0.163, 0.157	33.3, 25.9, 25.8	0.4
	ScorePose [57]	9.4, 12.5, 13.8	0.119, 0.148, 0.131	40.5, 30.1, 29.6	1.7
	CAPER++ (Ours)	**2.6**, **4.2**, **4.3**	**0.031**, **0.047**, **0.050**	**64.5**, **61.3**, **61.2**	**50.0**
Dishwasher	A-NCSH [11]	5.0, 5.7	0.074, 0.119	64.5, 43.8	0.6
	ContactArt [27]	7.8, 12.4	0.196, 0.234	44.3, 24.3	1.5
	CAPTRA [13]	4.6, 5.4	0.055, 0.089	85.3, 61.2	9.1
	GAPS [72]	6.2, 7.0	0.126, 0.207	80.3, 54.3	2.8
	Lie-X [56]	6.8, 8.9	0.170, 0.212	61.2, 39.8	0.4
	ScorePose [57]	5.3, 6.6	0.081, 0.137	82.3, 58.3	2.0
	CAPER++ (Ours)	**3.5**, **4.6**	**0.049**, **0.080**	**87.6**, **72.1**	**50.0**
Scissors	A-NCSH [11]	5.0, 5.7	0.041, 0.057	32.3, 32.8	0.8
	ContactArt [27]	16.8, 14.5	0.185, 0.167	23.6, 21.0	2.0
	CAPTRA [13]	4.1, 4.7	0.032, 0.039	43.2, 42.8	8.3
	GAPS [72]	6.1, 6.6	0.055, 0.069	41.3, 40.2	3.5
	Lie-X [56]	11.3, 10.1	0.131, 0.123	33.2, 31.8	0.4
	ScorePose [57]	5.3, 5.7	0.039, 0.051	42.1, 41.3	1.4
	CAPER++ (Ours)	**3.7**, **4.4**	**0.025**, **0.033**	**44.5**, **46.3**	**50.0**
Drawer	A-NCSH [11]	8.6, 9.8, 11.5, 8.5	0.088, 0.255, 0.257, 0.175	66.5, 62.3, 58.6, 61.3	0.3
	ContactArt [27]	12.5, 19.8, 19.6, 20.1	0.234, 0.342, 0.338, 0.337	51.3, 50.6, 47.6, 49.6	1.0
	CAPTRA [13]	4.8, 6.5, 6.3, 6.0	0.112, 0.185, 0.177, 0.156	89.5, 80.1, 76.7, 78.2	4.0
	GAPS [72]	6.5, 6.5, 6.5, 6.5	0.168, 0.242, 0.243, 0.239	86.3, 76.5, 75.6, 77.6	1.6
	Lie-X [56]	8.7, 7.8, 8.9, 7.2	0.173, 0.281, 0.289, 0.273	68.9, 61.8, 59.6, 54.9	0.3
	ScorePose [57]	5.3, 6.1, 6.3, 6.3	0.139, 0.216, 0.218, 0.193	88.3, 78.3, 75.9, 78.1	1.0
	CAPER++ (Ours)	**4.5**, **4.5**, **4.5**, **4.5**	**0.098**, **0.170**, **0.158**, **0.153**	**90.1**, **81.1**, **77.9**, **82.1**	**50.0**

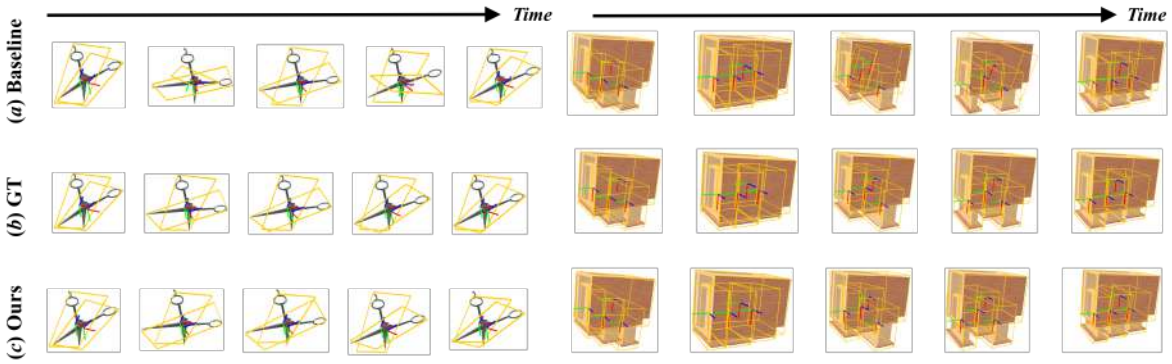

Fig. 9. **Qualitative Results on PM-Videos**. The left is a revolute joint case: *scissor*, while the right is a prismatic joint case: *drawer*.

(2) Generalization Capacity on Semi-Synthetic Scenarios. We evaluate the proposed articulated object pose tracking method on the semi-synthetic ReArt-Videos dataset. The quantitative results are reported in Tab. 4. For the *Box* category, our approach attains low average rotation errors of **4.6**° and **4.7**°, together with translation errors of **0.008**m and **0.007**m, respectively, demonstrating strong accuracy and robustness. These results validate the effectiveness of our tracking strategy, which explicitly exploits inter-frame priors through pose-increment prediction. Qualitative visualizations are provided in Fig. 10 (top).

(3) Generalization Capacity on Real-world Scenarios. To assess tracking performance under real-world scenarios, we train and evaluate CAPER++ on the RobotArm-Videos dataset. Quantitative results are summarized in Tab. 5, showing that CAPER++ achieves reliable 6D pose tracking accuracy in both rotation and translation. The observed robustness is largely attributed to the proposed dynamic keyframe selection strategy, which effectively mitigates error accumulation over long sequences. Nevertheless, the intrinsic multi-level depth hierarchy of robotic arms poses additional challenges and inevitably impacts tracking accuracy. We further validate our method on real-world video sequences, with qualitative results illustrated in Fig. 10 (bottom).

6.4 Ablation Study

Note that (1)-(2) are the ablation study of the pose estimation task, while (3)-(5) are the ablation study of the pose tracking task. For the pose estimation task, we trained and evaluated our model on the ArtImage dataset. To further assess the generalization capability of CAPER++ in complex scenarios, we conducted evaluations on two more challenging datasets—ReArt-Videos and RobotArm-Videos for the pose tracking task.

(1) Single Root Part Input *v.s.* Whole Object Input. To further assess the impact of constrained part information on root part pose estimation, we conducted an additional set of experiments,

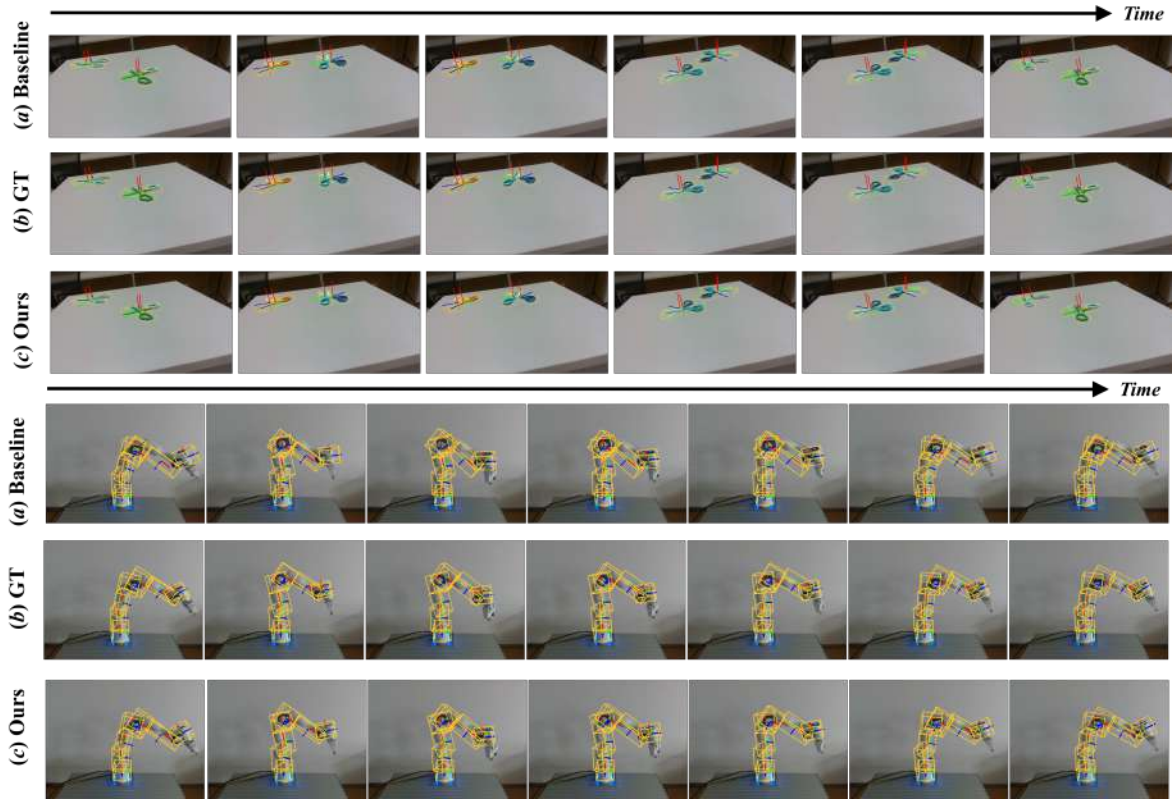

Fig. 10. **Qualitative Results on ReArt-Videos (top) and RobotArm-Videos (bottom)**. Visualizations of long-term tracking sequences and egocentric results from robotic experiments are available on our project website.

TABLE 7
Experiments Comparing Single Root Part as Input Between Both Root Part and Constrained Parts as Input. Note that we only investigate this analysis in the first step of our CAPER++.

Category	Input Parts	Rotation Error (°)	Translation Error (m)
Laptop	root	3.8	0.087
	root+constrained	**2.9**	**0.063**
Eyeglasses	root	5.7	0.103
	root+constrained	**4.5**	**0.074**
Dishwasher	root	20.4	0.143
	root+constrained	**2.5**	**0.064**
Scissors	root	6.7	0.088
	root+constrained	**4.3**	**0.067**
Drawer	root	3.9	0.099
	root+constrained	**1.3**	**0.066**

with results summarized in Tab. 7. Incorporating constrained parts led to a noticeable reduction in rotation error, as the network benefits from a more complete geometric context. This improvement is particularly pronounced in the Dishwasher category, where a 90% gain was observed. While a slight increase in translation error was noted—likely due to shifts in the root part's original position after merging with constrained components—the overall trade-off remains acceptable. *These findings highlight the importance of providing the whole geometry of the full articulation, enabling our framework to infer the poses of all parts concurrently from a single input.*

(2) Self-Occlusion Analysis. To evaluate the robustness of CAPER++ under self-occlusion, we partitioned the test samples of the Drawer category into three subsets according to occlusion severity, *as this issue is particularly prevalent in such objects.* Occlusion level is defined as the ratio of visible points to the total number of points on the part, following a similar protocol to A-NCSH [11]. Specifically, the three visibility intervals considered are: (0%, 40%), (40%, 80%), and (80%, 100%). We employ the mean rotation and translation errors to uniformly

TABLE 8
Ablation Study of Self-Occlusion. Pose errors for *Drawer* category with varying occlusion levels.

Occlusion Level	Rotation Error (°)	Translation Error (m)
0%-40%	1.9	0.057
40%-80%	2.0	0.062
80%-100%	2.0	0.095

assess pose estimation performance across subsets. As shown in Tab. 8, our joint-centric framework maintains consistent rotation accuracy regardless of occlusion degree. For visibility levels up to 80%, translation error remains low, and even in highly occluded scenarios (80%–100%), the model still performs reliably with a translation error of just $0.095m$. These results demonstrate the method's strong resilience to self-occlusion. *The key reason lies in the effectiveness of our proposed joint-centric modeling strategy in alleviating perception challenges caused by occlusion. Specifically, when a rigid part becomes difficult to perceive due to self-occlusion,* ***its corresponding joint state (i.e., the joint angle) typically remains unaffected and clearly observable****, thereby enabling stable and reliable estimation.*

(3) Lie Algebra. To evaluate the impact of Lie algebra representation on pose modeling performance, we conduct an ablation study in this section. As shown in Tab. 9, representing poses in the $\mathfrak{se}(3)$ space significantly reduces both rotation and translation errors. For the *Scissors* category, the rotation error decreases by 2.3° and 1.8°, while the translation error drops by $0.005m$ and $0.007m$, respectively. We attribute *this improvement to the Lie algebra's ability to provide* ***a minimal and continuous representation of rigid transformations within the*** $\mathfrak{se}(3)$ ***space***, thereby avoiding issues such as rotation orthogonality violations or geometric distortions commonly encountered in Euclidean space. In contrast, Euclidean modeling (direct two-axis modeling) fails to

TABLE 9
Ablation Study of Lie Algebra. Note that "Euc" treats the pose (i.e., the SE(3) transformation) as a vector in Euclidean space for direct modeling, whereas "Lie" models the transformation based on the $\mathfrak{se}(3)$ Lie algebra using the exponential map.

Category	SE(3) Manifolds	Per-part Pose	
		Rotation Error (°)	Translation Error (m)
Box	Euc	6.1 6.2	0.015 0.014
	Lie	**4.6 4.7**	**0.008 0.007**
Stapler	Euc	6.6 7.5	0.016 0.016
	Lie	**5.6 6.5**	**0.007 0.007**
Cutter	Euc	6.9 6.9	0.025 0.038
	Lie	**4.9 4.9**	**0.017 0.029**
Scissors	Euc	10.6 12.5	0.010 0.012
	Lie	**8.3 10.7**	**0.005 0.005**
Drawer	Euc	9.7 9.7	0.041 0.039
	Lie	**8.3 8.3**	**0.022 0.019**

capture the nonlinear structure of the SE(3) manifold, often leading to estimation bias. By preserving the geometric consistency of transformations, the Lie algebra-based formulation enhances both the physical plausibility and the optimization stability of pose modeling.

(4) Tracking Manner. To evaluate the performance difference between incremental prediction and conventional frame-by-frame prediction, we conduct an ablation study comparing their performance on the RobotArm dataset. Specifically, we test both frame-by-frame prediction and keyframe-based incremental prediction to analyze their respective advantages and limitations. The corresponding results are summarized in Tab. 10. *The No Keyframe setting yields the highest errors, highlighting that error accumulation severely degrades pose estimation performance when keyframes are absent.* Compared with tracking methods that do not utilize keyframes, algorithms adopting a fixed keyframe strategy exhibit significantly reduced pose errors on the terminal joints, with an error reduction of nearly 40% (**23.2**° *vs* 38.1° in rotation error and **0.212**m *vs* 0.407m in translation error).

(5) Keyframe Selection. As previously discussed, the selection of keyframes is crucial as it directly influences the incremental prediction of subsequent frames. To investigate this impact, we conduct an ablation study comparing fixed keyframes with dynamically selected keyframes. The corresponding results are summarized in Tab. 10. The Fixed keyframe setup partially mitigates this issue but remains inferior to the dynamic keyframe strategy. For instance, in the last part of robotarm, the rotation errors with fixed keyframes reach 23.2° and 0.212m, whereas the dynamic keyframe method reduces them to only **19.3**° and **0.139**m, respectively. *Experimental results demonstrate that the proposed dynamic joint-frame selection algorithm not only effectively suppresses structural errors arising from hierarchical joint accumulation in* ***multi-hinged structures****, but also mitigates temporal drift within frame sequences, thereby enhancing overall tracking accuracy and stability.*

TABLE 10
Ablation Study of Tracking Manner. Note that "N" represents "No Keyframe", "F" represents "Fixed Keyframe", while "D" represents "Dynamic Keyframe" (Ours).

Per-part Rotation Error (°)							
Part ID	1	2	3	4	5	6	7
N	0.3	2.9	8.5	13.6	19.7	25.3	38.1
F	0.1	1.3	6.5	9.8	14.6	18.3	23.2
D	**0.1**	**0.7**	**5.7**	**7.6**	**12.3**	**16.8**	**19.3**
Per-part Translation Error (m)							
Part ID	1	2	3	4	5	6	7
N	0.005	0.043	0.082	0.138	0.185	0.263	0.407
F	0.004	0.028	0.056	0.079	0.108	0.142	0.212
D	**0.002**	**0.016**	**0.043**	**0.061**	**0.089**	**0.102**	**0.139**

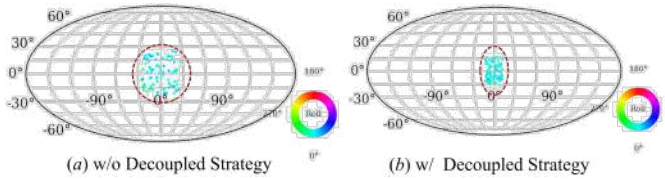

Fig. 11. **Visualization of SO(3) Distribution.** In this SO(3) distribution map, the longitude, latitude, and color respectively represent the prediction errors in yaw, pitch, and roll. The left figure shows the prediction errors of the conventional method on the base part of the laptop, while the right figure illustrates the prediction errors of the decomposed strategy on the same part. To highlight the advantage of our approach, the errors have been magnified by a factor of 3 in the visualization.

7 Limitation and Discussion

Application in Multi-root Systems. In this work, we assume that an articulated object contains a single root part, with the remaining parts connected through constrained joints. This assumption is consistent with the structural characteristics of articulated objects commonly encountered in daily environments and existing benchmark datasets. It should be noted that this formulation primarily defines the scope of the current study, rather than imposing a fundamental constraint on the underlying modeling paradigm. Conceptually, more complex articulated systems with multiple root parts, such as robots with floating bases and complex mobility, can be modeled by introducing free joints between root parts and constrained parts. Depending on the dominant motion coupling, such free joints can be characterized by relative translational or rotational motion, while ***continuing to adopt the proposed joint-centric representation without modifying the core pose estimation or tracking pipeline***. A systematic investigation of such extensions, however, relies on appropriate datasets and evaluation protocols, and is therefore beyond the scope of this work. We leave this direction for future research.

Object Segmentation. In this work, we employ an off-the-shelf object segmenter [73] to obtain object-level point cloud inputs for articulated pose estimation and tracking. When the segmentation results are inaccurate, the extracted point clouds for individual objects or parts may become incomplete or contaminated by noise, which can adversely affect the quality of downstream pose predictions. Improving the accuracy and robustness of the upstream segmenter is therefore expected to enhance the cleanliness of the input data and alleviate error propagation in the overall pipeline. We believe that advances in object segmentation techniques will further strengthen the reliability of articulated pose perception systems, and we leave the integration of more accurate segmenters as future work.

Visualization of SO(3) Distribution. We employ a decomposed rotation strategy that grounds pose estimation on a reliable anchor axis derived from object surface normals. For instance, for the root part of a laptop, the surface normal, which is perpendicular to the planar surface, is geometrically well defined and highly predictable. This axis is therefore used as a stable reference to guide full 6D pose estimation. As illustrated in Fig. 11, ablation results show that conventional methods exhibit similar error magnitudes across all three rotational axes, yielding an approximately uniform error distribution. In contrast, our approach explicitly exploits the reliability of the anchor direction, reducing the rotation error along the dominant axis to below 4 degrees, which is substantially lower

than that of baseline methods. Moreover, more accurate estimation of this primary axis also constrains the remaining axes, leading to a more compact and ellipsoidal error distribution. These results confirm that incorporating structural priors enables more accurate and stable pose estimation.

8 Conclusion

In this work, we propose CAPER++, a unified framework for efficient category-level articulation pose perception. A hierarchical strategy decomposes objects into a root part and constrained parts, enabling joint prediction of pose and joint states under explicit kinematic constraints. Robustness is enhanced via a vertical correction mechanism, Lie algebra-based rotation representation to preserve orthogonality, and a scale-aware heatmap voting scheme for stable joint axis estimation. For pose tracking, CAPER++ models inter-frame pose increments via proxy canonicalization and mitigates drift using dynamic keyframe selection. Extensive evaluations on synthetic, semi-synthetic, and real-world datasets confirm its state-of-the-art performance and generalization.

References

[1] X. Yu, H. Jiang, L. Zhang, L. Y. Wu, L. Ou, and L. Liu, "Efficientcaper: An end-to-end framework for fast and robust category-level articulated object pose estimation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 31 968–31 989, 2024.

[2] R. Shao, T. Wu, J. Wu, L. Nie, and Z. Liu, "Detecting and grounding multi-modal media manipulation and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[3] Y. Ji, H. Tan, J. Shi, X. Hao, Y. Zhang, H. Zhang, P. Wang, M. Zhao, Y. Mu, P. An *et al.*, "Robobrain: A unified brain model for robotic manipulation from abstract to concrete," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 1724–1734.

[4] J. Wu, Y. Zhou, H. Yang, Z. Huang, and C. Lv, "Human-guided reinforcement learning with sim-to-real transfer for autonomous navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 14 745–14 759, 2023.

[5] D. Baril, S.-P. Deschênes, O. Gamache, M. Vaidis, D. LaRocque, J. Laconte, V. Kubelka, P. Giguère, and F. Pomerleau, "Kilometer-scale autonomous navigation in subarctic forests: challenges and lessons learned," *Field Robotics*, vol. 2, pp. 1628–1660, 2022.

[6] A. Badki, O. Gallo, J. Kautz, and P. Sen, "Binary ttc: A temporal geofence for autonomous navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 946–12 955.

[7] S. Ghosh, A. Dhall, M. Hayat, J. Knibbe, and Q. Ji, "Automatic gaze analysis: A survey of deep learning based approaches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 1, pp. 61–84, 2023.

[8] J. Xiong, E.-L. Hsiang, Z. He, T. Zhan, and S.-T. Wu, "Augmented reality and virtual reality displays: emerging technologies and future perspectives," *Light: Science & Applications*, vol. 10, no. 1, p. 216, 2021.

[9] Y. You, W. He, J. Liu, H. Xiong, W. Wang, and C. Lu, "Cppf++: uncertainty-aware sim2real object pose estimation by vote aggregation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[10] G. Wang, F. Manhardt, X. Liu, X. Ji, and F. Tombari, "Occlusion-aware self-supervised monocular 6d object pose estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 3, pp. 1788–1803, 2021.

[11] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3706–3715.

[12] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2642–2651.

[13] Y. Weng, H. Wang, Q. Zhou, Y. Qin, Y. Duan, Q. Fan, B. Chen, H. Su, and L. J. Guibas, "Captra: Category-level pose tracking for rigid and articulated objects from point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 209–13 218.

[14] H. Xue, L. Liu, W. Xu, H. Fu, and C. Lu, "Omad: Object model with articulated deformations for pose estimation and retrieval," *arXiv preprint arXiv:2112.07334*, 2021.

[15] P. Zhang, T. Zhuo, H. Huang, K. Chen, B. Zhang, and M. Kankanhalli, "Robust tracking based on h-cnn with low-resource sampling and scaling by frame-wise motion localization," *Multimedia Tools and Applications*, vol. 77, pp. 18 781–18 800, 2018.

[16] C. Sun, M. Tappen, and H. Foroosh, "Feature-independent action spotting without human localization, segmentation or frame-wise tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2689–2696.

[17] A. Elgammal and C.-S. Lee, "Inferring 3d body pose from silhouettes using activity manifold learning," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2. IEEE, 2004, pp. II–II.

[18] Y. Shan, B. Zheng, L. Chen, L. Chen, and D. Chen, "A reinforcement learning-based adaptive path tracking approach for autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10 581–10 595, 2020.

[19] H. Qu, L. Song, and G. Xue, "Shaking video synthesis for video stabilization performance assessment," in *2013 Visual Communications and Image Processing (VCIP)*. IEEE, 2013, pp. 1–6.

[20] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 6, pp. 2872–2893, 2021.

[21] M. Lu and F. Li, "Survey on lie group machine learning," *Big Data Mining and Analytics*, vol. 3, no. 4, pp. 235–258, 2020.

[22] D. S. Brezov, C. D. Mladenova, and I. M. Mladenov, "New perspective on the gimbal lock problem," in *AIP Conference Proceedings*, vol. 1570, no. 1. American Institute of Physics, 2013, pp. 367–374.

[23] F. Zhang, "Quaternions and matrices of quaternions," *Linear algebra and its applications*, vol. 251, pp. 21–57, 1997.

[24] J. L. Blanco-Claraco, "A tutorial on se(3) transformation parameterizations and on-manifold optimization," *arXiv preprint arXiv:2103.15980*, 2021.

[25] Y. Wu and M. Carricato, "Persistent manifolds of the special euclidean group se (3): a review," *Computer Aided Geometric Design*, vol. 79, p. 101872, 2020.

[26] B. S. McCann, A. Anderson, M. Nazari, D. Canales, and R. Beeson, "On-manifold pose optimization on se (3) for spacecraft coverage maximization," in *AAS/AIAA Astrodynamics Specialists Conference*, 2023.

[27] Z. Zhu, J. Wang, Y. Qin, D. Sun, V. Jampani, and X. Wang, "Contactart: Learning 3d interaction priors for category-level articulated object and hand poses estimation," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 201–212.

[28] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[29] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, 2014, pp. 740–755.

[31] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2334–2343.

[32] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[33] L. Liu, H. Xue, W. Xu, H. Fu, and C. Lu, "Toward real-world category-level articulation pose estimation," *IEEE Transactions on Image Processing*, vol. 31, pp. 1072–1083, 2022.

[34] L. Zhang, W. Meng, Y. Zhong, B. Kong, M. Xu, J. Du, X. Wang, R. Wang, and L. Liu, "U-cope: Taking a further step to universal 9d category-level object pose estimation," in *European Conference on Computer Vision*. Springer, 2024, pp. 254–270.

[35] X. Zhang, G. Zhang, K. Nakamura, and S. Ueha, "A robot finger joint driven by hybrid multi-dof piezoelectric ultrasonic motor," *Sensors and Actuators A: Physical*, vol. 169, no. 1, pp. 206–210, 2011.

[36] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 529–537.

[37] M. Priorelli, G. Pezzulo, and I. P. Stoianov, "Deep kinematic inference affords efficient and scalable control of bodily movements," *Proceedings of the National Academy of Sciences*, vol. 120, no. 51, p. e2309058120, 2023.

[38] L. Meyer, L. Klüpfel, M. Durner, and R. Triebel, "Robust probabilistic robot arm keypoint detection exploiting kinematic knowledge," in *IROS 2022 Workshop Probabilistic Robotics in the Age of Deep Learning*, 2022.

[39] V. Drouard, R. Horaud, A. Deleforge, S. Ba, and G. Evangelidis, "Robust head-pose estimation based on partially-latent mixture of linear regressions," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1428–1440, 2017.

[40] M.-D. Hua, M. Zamani, J. Trumpf, R. Mahony, and T. Hamel, "Observer design on the special euclidean group se (3)," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 8169–8175.

[41] J. Park and W.-K. Chung, "Geometric integration on euclidean group with application to articulated multibody systems," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 850–863, 2005.

[42] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.

[43] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on robot learning*. PMLR, 2022, pp. 894–906.

[44] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," *arXiv preprint arXiv:2110.14217*, 2021.

[45] X. Hao and Z. Zhang, "An iterative algorithm for camera pose estimation with lie group representation," in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2017, pp. 187–192.

[46] K. Kanatani, "Lie algebra method for pose optimization computation," *Machine Vision and Navigation*, pp. 293–319, 2020.

[47] Y. Ni, X. Wang, and L. Yin, "Relative pose estimation for multiple cameras using lie algebra optimization," *Applied Optics*, vol. 58, no. 11, pp. 2963–2972, 2019.

[48] Z. Wang, Z. He, and J. Huang, "Lie algebra-based multivalidation model for 3d human pose estimation," in *International Conference on Computer Application and Information Security (ICCAIS 2024)*, vol. 13562. SPIE, 2025, pp. 294–302.

[49] O. Tuzel, F. Porikli, and P. Meer, "Learning on lie groups for invariant detection and tracking," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[50] W. Li, J. Liu, Y. Wang, W. Hao, D. Ren, and L. Chen, "Dl-posenet: A differential lightweight network for pose regression over se(3)," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 834–16 840.

[51] A. Pokhrel, M. Nazeri, A. Datar, and X. Xiao, "Cahsor: Competence-aware high-speed off-road ground navigation in se(3)," *IEEE Robotics and Automation Letters*, 2024.

[52] H. Zhu, Z. Zheng, W. Zheng, and R. Nevatia, "Cat-nerf: Constancy-aware tx2former for dynamic body modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6619–6628.

[53] Z. Teed and J. Deng, "Tangent space backpropagation for 3d transformation groups," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 338–10 347.

[54] Z.-Z. Sun, S.-J. Ran, and G. Su, "Tangent-space gradient optimization of tensor network for machine learning," *Physical Review E*, vol. 102, no. 1, p. 012152, 2020.

[55] H. Xiaofei and L. Binbin, "Tangent space learning and generalization," *Frontiers of Electrical and Electronic Engineering in China*, vol. 6, no. 1, pp. 27–42, 2011.

[56] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng, "Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups," *International Journal of Computer Vision*, vol. 123, no. 3, pp. 454–478, 2017.

[57] T.-C. Hsiao, H.-W. Chen, H.-K. Yang, and C.-Y. Lee, "Confronting ambiguity in 6d object pose estimation via score-based diffusion on se (3)," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 352–362.

[58] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3456–3462.

[59] G. Liao, C. Zheng, L. Cheng, H. Xie, S. Huang, J. Liao, H. Li, and L. Liu, "Hiposer: 3d human pose estimation with hierarchical shared learning at parts-level using inertial measurement units," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 13, 2025, pp. 14 309–14 317.

[60] V. Joukov, J. Ćesić, K. Westermann, I. Marković, I. Petrović, and D. Kulić, "Estimation and observability analysis of human motion on lie groups," *IEEE transactions on cybernetics*, vol. 50, no. 3, pp. 1321–1332, 2019.

[61] S. Zou, Y. Mu, W. Ji, Z.-A. Wang, X. Zuo, S. Wang, W. Si, and L. Cheng, "Highly efficient 3d human pose tracking from events with spiking spatiotemporal transformer," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[62] Z. Liu, S. Wu, S. Jin, S. Ji, Q. Liu, S. Lu, and L. Cheng, "Investigating pose representations and motion contexts modeling for 3d motion prediction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 681–697, 2022.

[63] Z. Li, W. Ye, D. Wang, F. X. Creighton, R. H. Taylor, G. Venkatesh, and M. Unberath, "Temporally consistent online depth estimation in dynamic scenes," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 3018–3027.

[64] L. Zheng, C. Wang, Y. Sun, E. Dasgupta, H. Chen, A. Leonardis, W. Zhang, and H. J. Chang, "Hs-pose: Hybrid scope feature extraction for category-level object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 163–17 173.

[65] Y. Di, R. Zhang, Z. Lou, F. Manhardt, X. Ji, N. Navab, and F. Tombari, "Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6781–6791.

[66] Z.-H. Lin, S.-Y. Huang, and Y.-C. F. Wang, "Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1800–1809.

[67] W. Chen, X. Jia, H. J. Chang, J. Duan, L. Shen, and A. Leonardis, "Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1581–1590.

[68] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu, "6-pack: Category-level 6d pose tracker with anchor-based keypoints," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 059–10 066.

[69] Z. Liu, Q. Wang, D. Liu, and J. Tan, "Pa-pose: Partial point cloud fusion based on reliable alignment for 6d pose tracking," *Pattern Recognition*, vol. 148, p. 110151, 2024.

[70] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 097–11 107.

[71] L. Liu, J. Du, H. Wu, X. Yang, Z. Liu, R. Hong, and M. Wang, "Category-level articulated object 9d pose estimation via reinforcement learning," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 728–736.

[72] Q. Yu, C. Hao, X. Yuan, L. Zhang, L. Liu, Y. Huo, R. Agarwal, and C. Lu, "Generalizable articulated object perception with superpoints," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[73] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

Li Zhang is currently pursuing the Ph.D. degree in computer science and application with the University of Science and Technology of China, Hefei. His research interests include computer vision, machine learning, and embodied AI. In particular, he is interested in the rigid, articulated, and non-rigid object pose perception. He has published several top-tier conference papers at ECCV, NeurIPS, ICML etc.

Xianhui Meng received the B.E. degree from the Southeast University (SEU), Nanjing, China, in 2024. He is currently pursuing Information and Communication Engineering at the University of Science and Technology of China (USTC), Hefei, China. His research interests include Embodied Intelligence, 3D vision, computer vision and deep learning.

Liu Liu received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015, and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2020. He was a Postdoctoral Researcher with Shanghai Jiao Tong University, China, from 2020 to 2022. He is currently an Associate Professor with Hefei University of Technology. His research interests include computer vision, deep learning, and embodied AI.

Haonan Jiang received the B.E. degree in Measurement and Control Technology and Instrument from Qingdao University of Science and Technology, Qingdao, China, in 2022. He is currently pursuing the M.S. degree at the Zhejiang University of Technology. His current research interests include computer vision and embodied AI.

Jianan Wang is the Chief Researcher in AI cognition at Astribot. Previously, she received the bachelor's degree from the Chinese University of Hong Kong and the MSc degree from the University of Oxford. She has previously worked with DeepMind and the International Digital Economy Academy (IDEA). Her research interests include computer vision, deep learning, and machine learning theory, with a recent focus on generative AI and robotics.

Rujing Wang received the B.E. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 1987, and M.S. degree in electronic engineering from Dalian University of Technology, Dalian, China, in 1990, and Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in 2005. He is currently working with the Institute of Intelligent Machinery of the Chinese Academy of Sciences as Professor and Researcher. His main research interests include intelligent agriculture, agricultural Internet of things, agricultural knowledge engineering.

Cewu Lu (Member, IEEE) is professor with Shanghai Jiao Tong University, distinguished professor of the Changjiang Scholars Program, and recipient of the Scientific Exploration Award. In 2016, he was selected as a high-level overseas young talent, and in 2018, he was named one of the 35 Innovators Under 35 in China by MIT Technology Review (MIT TR35). In 2019, he received the Qiushi Outstanding Young Scholar Award, and in 2020, he was the third contributor to the Shanghai Science and Technology Progress Special Award. In 2022, he received the Ministry of Education Youth Science Award. His research interests fall mainly in computer vision, deep learning, and robotics.

Jun Liu (Senior Member, IEEE) received the B.S. degree in mathematics from Wuhan University of Technology, Wuhan, China, in 2006, the M.S. degree in mathematics from Chinese Academy of Sciences, China, in 2009, and the Ph.D. degree in electrical engineering from Xidian University, Xi'an, China, in 2012. From July 2012 to December 2012, he was a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. From January 2013 to September 2014, he was a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA. He is currently an Associate Professor with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China. His research interests include statistical signal processing, image processing, and machine learning.

Hong Zhang (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from Purdue University in 1986. He was a Professor with the Department of Computing Science, University of Alberta, Canada, for over 30 years, before he joined the Southern University of Science and Technology (SUSTech), China, in 2020, where he is currently the Chair Professor. His research interests include robotics, computer vision, and image processing. He is a fellow of Canadian Academy of Engineering. He served as the Editor-in-Chief for the IROS Conference Paper Review Board (2020-2022) and is currently a member of the IEEE Robotics and Automation Society Administrative Committee (2023-2025).