KINO: 3D KEYPOINT-MEDIATED STATE-IMAGE REASONING FOR ROBOT POSE AND JOINT ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-view robot pose and joint estimation requires recovering an articulated state whose pose, joints, and visible parts must be spatially consistent under robot kinematics. Existing methods establish this consistency either implicitly through direct regression or explicitly through render-and-compare refinement. The former leaves the relation between image observations and 3D articulated states implicit, while the latter makes consistency checking explicit in the image plane but still requires the refinement model to infer 3D state corrections from 2D discrepancies. In this paper, we propose **KINO**, which formulates robot state estimation as latent geometric reasoning between image-induced and state-induced robot geometry. Rather than associating image evidence only with abstract pose-joint parameters, KINO maps image-induced 3D keypoints and forward-kinematics-derived state keypoints into a shared geometric token space, providing a common interface between observation and state. Dense visual evidence and canonical structural cues provide complementary context for state refinement. After each update, KINO regenerates Kinematic Keypoint Tokens from the updated pose-joint estimate through forward kinematics and feeds this geometry back into the decoder. This kinematics-driven feedback supplies the geometry implied by the current state to subsequent refinement, rather than only re-embedding state parameters. Controlled studies demonstrate the benefits of the shared geometric representation and kinematics-driven feedback. On RoboKeyGen and DREAM, KINO outperforms representative baselines, improving real-world AUC by 7.27 and 9.46 points while running at 10.5 FPS (95.53 ms per image).

1 INTRODUCTION

Robot pose and joint estimation aims to recover the 6-DoF camera-to-robot transformation and joint configurations from a single-view image. This provides the necessary 3D robot state for robot localization, human-robot interaction (Christen et al., 2023; Yang et al., 2021; Svenstrup et al., 2009), multi-robot collaboration (Li et al., 2022; Ying & Zhou, 2023; Cossette et al., 2022), and embodied AI (Huang et al., 2023; Firoozi et al., 2025; Di Palo et al., 2023). Unlike standard rigid object pose estimation, a robot is an articulated system whose global pose, joint configuration, and visible parts are tightly coupled by its kinematic structure. Therefore, estimating robot pose and joints from a single image requires reasoning about the spatial consistency between observed robot parts and the underlying pose-joint state.

Existing methods establish spatial consistency through two paradigms. Direct prediction methods such as HPE (Ban et al., 2024) and RoboTAG (Liu et al., 2026) estimate robot pose and joint parameters in a feed-forward manner, with geometric relations enforced through supervision on the predicted outputs. They are efficient, but still rely on learned observation-to-state mappings, leaving state-image consistency implicit during inference. In contrast, render-and-compare methods explicitly evaluate a candidate robot state by rendering it into the image plane and comparing it with the input observation (Labbé et al., 2021; Zhu et al., 2025). While this provides a clearer state-image association, the comparison is still performed between 2D visual evidence and a rendered projection of the 3D state, leaving the relation from image-space discrepancy to articulated-state correction to be learned by the refinement model. Repeated rendering and iterative optimization further increase

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

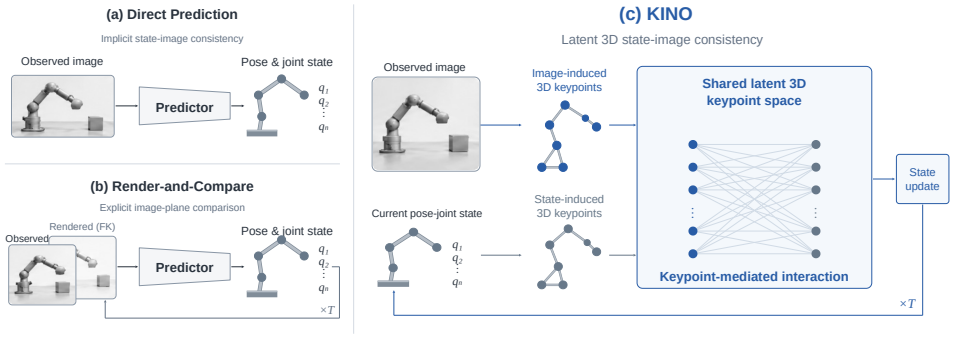


Figure 1: **State-image consistency across robot state estimation paradigms.** (**a**) Direct prediction maps image evidence directly to pose-joint parameters, leaving their geometric consistency implicit. (**b**) Render-and-compare evaluates a candidate state after projecting it into the image plane. (**c**) KINO instead relates image-induced and state-induced 3D geometry through a shared latent keypoint interface, with updated state geometry fed back across refinement steps.

computational cost. More fundamentally, both paradigms retain a representational gap between 2D visual observations and the underlying 3D articulated state. This raises a central question: *can we establish geometrically grounded state-image consistency directly in a shared 3D representation?*

To answer this question, we propose **KINO**, which reformulates single-view robot pose and joint estimation as latent geometric reasoning between image-induced and state-induced robot geometry (Fig. 1). Our key idea is to create a shared geometric interface between image observations and the current robot-state hypothesis. KINO lifts monocular observations into image-induced 3D Keypoint Tokens and converts the current pose-joint estimate into state-induced Kinematic Keypoint Tokens through forward kinematics. Both keypoint sets describe robot-part locations in the same camera frame and are embedded through a common projection. Their interaction provides a geometric basis for relating the 3D layout inferred from the image to the layout implied by the current state, rather than relying on abstract pose-joint parameters alone. Self-attention models interactions between these geometric tokens, while bidirectional cross-attention incorporates dense image evidence.

KINO augments this shared geometric interface with complementary context. 2D Visual Tokens preserve dense image evidence that may be lost in sparse keypoint representations, while Canonical Structure Tokens encode pose-independent robot structure from the lifted 3D keypoints. These tokens form a compact multi-modal space where image evidence, image-induced geometry, state-induced geometry, and canonical robot structure can interact through attention. KINO then places kinematics inside the refinement loop. At each step, the decoder updates pose and joints using observation- and state-derived geometric tokens together with visual and structural context. After each update, forward kinematics regenerates Kinematic Keypoint Tokens from the updated pose-joint estimate and feeds them into the next decoder layer. Thus, each intermediate state is converted back into its implied 3D robot layout before the next reasoning step, rather than only being re-embedded as pose-joint parameters.

Extensive experiments on RoboKeyGen and DREAM demonstrate the effectiveness of KINO. Controlled ablations separately verify the contributions of the shared keypoint representation and kinematics-driven feedback. We further compare KINO with constrained fitting using the same frozen keypoint proposals, and report multi-seed evaluations and component-wise analyses of pose and joint accuracy. KINO outperforms the representative baselines, improving real-world AUC by 7.27 and 9.46 points on RealSense and AzureKinect, respectively. Full KINO runs at 10.5 FPS (95.53 ms per image) on an NVIDIA TITAN RTX GPU, with its accuracy-latency trade-off analyzed in Appendix B.

In summary, our contributions are as follows:

- We propose **KINO**, which reformulates single-view robot pose and joint estimation as latent geometric reasoning between image-induced and state-induced 3D robot geometry. A shared 3D keypoint representation provides the interface between image evidence and a candidate articulated state.
- We design this shared geometric interface by expressing image-induced and forward-kinematics-derived state keypoints in the same camera frame and embedding them through

a common projection. Visual, state, and canonical structural cues provide complementary context for their interaction.

- We introduce kinematics-driven geometric feedback that regenerates Kinematic Keypoint Tokens after each state update. By converting each updated pose-joint estimate back into its implied 3D geometry, the next refinement step reasons from the current spatial hypothesis rather than from state parameters alone.

2 Related work

Robot Pose Estimation. Traditional camera-to-robot pose estimation often relies on hand-eye calibration and fiducial-marker-based methods (Horaud & Dornaika, 1995; Garrido-Jurado et al., 2014; Fiala, 2005; Olson, 2011). Related model-based methods have also addressed articulated-object pose detection and tracking (Pauwels et al., 2014). Recently, markerless, deep learning-based methods have attracted increasing attention, and can be broadly categorized into keypoint-based and rendering-based approaches, following paradigms widely adopted in general object pose estimation (Lin et al., 2022a;b; Labbe et al., 2020; Li et al., 2020; Jiang et al., 2023; Wen et al., 2023; Wen & Bekris, 2021). The keypoint-based method utilizes a neural network to detect the 2D keypoints at specific robot parts in the image, and then the 2D-3D correspondence is used to recover the camera-to-robot pose via a PnP-solver (Gao et al., 2003; Lepetit et al., 2009; Moreno-Noguer et al., 2007). DREAM (Lee et al., 2020) introduces a large-scale dataset with 6D pose and joint angle annotations, along with a stacked hourglass network (Newell et al., 2016) for 2D keypoint detection. Building on this, SGTAPose (Tian et al., 2023) exploits robot structural priors and temporal consistency to mitigate occlusions and ambiguities in single-view settings. CtRNet (Lu et al., 2023) addresses the sim-to-real gap via self-supervision based on robot mask discrepancies. CtRNet-X (Lu et al., 2025) further incorporates vision-language models to enable fine-grained component detection, particularly for invisible parts. Rendering-based refinement provides another paradigm for pose estimation. DeepIM (Li et al., 2020) iteratively refines object pose by matching observed and rendered images. RoboPose (Labbé et al., 2021) adopts a related render-and-compare paradigm to jointly estimate robot pose and joint angles, while MonoSE(3)-Diffusion (Zhu et al., 2025) reformulates iterative robot pose estimation as a diffusion process (Ho et al., 2020). However, robot pose estimation largely treats articulated robots as rigid objects by assuming known joint configurations, which are not always available in practical scenarios. As such, we tackle the challenging problem of jointly estimating robot pose and joint configurations.

Robot Pose & Joints Estimation. RoboPose (Labbé et al., 2021) can jointly estimate pose and joint angles via a rendering-based paradigm, but it comes at a high computational cost. To improve computational efficiency, HPE (Ban et al., 2024) regresses joints and rotation from image features and recovers translation from predicted depth and keypoints. Its self-supervised training uses a consistency loss between image-predicted and FK-derived keypoints. RoboPEPP (Goswami et al., 2025) further enhances the semantic understanding of 2D image encoders by introducing a self-supervised embedding-predictive framework. Similarly, RoboKeyGen (Tian et al., 2024) decomposes the challenging direct 2D-to-3D prediction into two stages: 2D keypoint detection followed by 3D layout prediction via an image-agnostic diffusion model. RoboTAG (Liu et al., 2026) combines image-based camera-pose prediction with joint estimation from depth-lifted 3D keypoints, and links branch predictions and kinematic reconstructions through topological consistency losses. HPE and RoboTAG enforce consistency among their predictions through training losses. In contrast, KINO models interactions between image-induced and state-induced geometry during iterative inference. A shared keypoint embedding enables these interactions. After each update, forward kinematics regenerates the state-induced geometry for the next refinement step.

3 Method

KINO estimates robot pose and joints by reasoning about state-image consistency in a latent space. Image-induced and state-induced 3D keypoints provide the shared geometric representation. Image observations are represented by image-induced 3D Keypoint Tokens (Sec. 3.1), while the iteratively updated pose-joint state is represented by state-induced Kinematic Keypoint Tokens (Sec. 3.2). Together with Canonical Structure Tokens, these tokens interact through a multi-modal attention de-

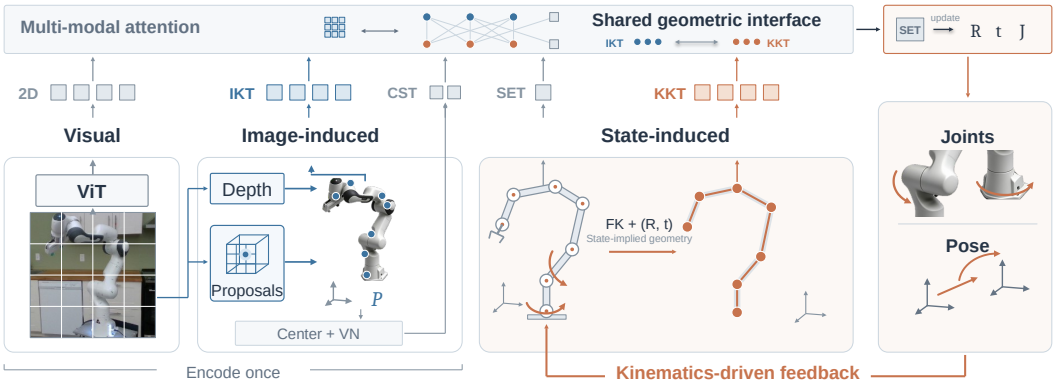


Figure 2: **Overview of KINO.** KINO couples image observations and a candidate robot state through a shared geometric interface. Image-induced 3D keypoints are embedded as Image-induced Keypoint Tokens (IKT), while forward kinematics converts the current pose-joint estimate into state-induced Kinematic Keypoint Tokens (KKT) in the same camera-frame keypoint space. 2D Visual Tokens, the State Embedding Token (SET), and Canonical Structure Tokens (CST) provide complementary visual, global-state, and structural context for multi-modal interaction and state update. After each update, forward kinematics regenerates KKT from the updated pose-joint estimate, feeding the geometry implied by the current state into the next decoder layer.

coder to iteratively predict robot rotation, translation, and joint configuration (Secs. 3.3–3.4). KINO is trained with supervised state and keypoint losses (Sec. 3.5).

3.1 Image-induced 3D Keypoint Tokenization

Given an input image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$, our goal is to estimate the camera-to-robot transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in SE(3)$ and the joint configuration $\mathbf{J} \in \mathbb{R}^{N_{\text{dof}}}$. Here, $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ denote the robot rotation and translation, and N_{dof} is the number of articulated degrees of freedom. As discussed above, directly associating monocular image features with abstract pose-joint parameters is difficult. We therefore first construct the image-induced representation of the 3D keypoint bridge, which lifts image observations into 3D keypoint tokens as well as preserves dense visual evidence for later state-image association.

Image-induced 3D Keypoint Tokens. KINO starts its state-image association from the image-induced keypoint tokens:

$$\mathbf{T}_{\text{IKT}} = \text{Embed}(\mathbf{P}) \in \mathbb{R}^{N\times C}. \tag{1}$$

Here, $\mathbf{P} \in \mathbb{R}^{N\times 3}$ denotes N camera-frame 3D keypoint proposals inferred from the input image. Specifically, Embed is a shared affine projection applied independently to each xyz coordinate. Following HPE (Ban et al., 2024), we predict the absolute root depth d and lift monocular features with a volumetric keypoint proposal network to obtain $\mathbf{P}$. The root proposal initializes the translation as $\mathbf{t} = \mathbf{P}_1$, while the full proposal set provides the image-side geometric representation whose depth, 3D keypoints, and 2D projections are supervised during Stage I (Sec. 3.5). Thus, $\mathbf{T}_{\text{IKT}}$ supplies the local 3D anchors that connect image observations to the state-induced representation.

Keypoint proposals. Given image $\mathcal{I}$, a depth network predicts the absolute root depth,

$$d = \text{DepthNet}(\mathcal{I}) \in \mathbb{R}. \tag{2}$$

In parallel, a keypoint-proposal backbone $\phi(\cdot)$ extracts dense visual features $\mathbf{F} = \phi(\mathcal{I}) \in \mathbb{R}^{H'\times W'\times C}$, where H' and W' are the downsampled spatial resolution and C is the feature dimension. Following HPE (Ban et al., 2024), a volumetric keypoint proposal head combines these features with the root-depth estimate to lift monocular observations into 3D.

Specifically, the head produces a volumetric heatmap $\mathbf{H} \in \mathbb{R}^{N\times D\times H'\times W'}$, where N is the number of robot keypoints and D is the discretized depth dimension. Each voxel represents the likelihood that a keypoint occupies the corresponding 3D location. The root-relative location of the n-th robot

keypoint is obtained by differentiable spatial expectation:

$$\tilde{\mathbf{P}}_n = \sum_{k=1}^{D} \sum_{i=1}^{H'} \sum_{j=1}^{W'} (k, i, j) \cdot \mathbf{H}_n(k, i, j). \tag{3}$$

Combining $\tilde{\mathbf{P}}$ with d gives the camera-frame proposals $\mathbf{P} \in \mathbb{R}^{N\times 3}$, whose root keypoint is used as the initial translation. These proposals are embedded by Eq. 1; during Stage I, the depth prediction, camera-frame proposals, and their 2D projections are supervised as described in Sec. 3.5.

2D Visual Tokens. While image-induced 3D keypoint tokens above provide a compact 3D representation of observable robot parts, they do not fully preserve dense appearance and contextual cues from the image. We therefore use a dedicated DINO-initialized ViT-B backbone (Dosovitskiy et al., 2021; Caron et al., 2021) with a patch size of 16 and 12 Transformer layers to extract image patch features as dense 2D Visual Tokens. After a linear projection to the common token dimension, the patch features form $\mathbf{T}_{2D} \in \mathbb{R}^{N_{2D}\times C}$. The 2D Visual Tokens complement the sparse 3D keypoint tokens by preserving full-image evidence for subsequent multi-modal interaction.

3.2 State-induced 3D Kinematic Keypoint Tokenization

The image-induced tokens describe the robot parts inferred from the monocular observation. To associate them with a candidate robot state, we also need a token representation of the robot layout implied by the current pose-joint estimate. We first encode the complete pose-joint estimate as a compact State Embedding Token (SET). Writing the rotation matrix as $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$, we concatenate its first two columns with the translation and joint configuration, and project the resulting state vector into the common token space:

$$\mathbf{T}_{\text{SET}} = \text{Embed}_{\text{state}}\left([\mathbf{r}_1; \mathbf{r}_2; \mathbf{t}; \mathbf{J}]\right) \in \mathbb{R}^{1\times C}. \tag{4}$$

At the first refinement step, $\mathbf{R}$ and $\mathbf{J}$ are initialized with their default values, while $\mathbf{t}$ is initialized by the image-induced root keypoint predicted in Sec. 3.1. Although this pose-joint embedding provides an effective global summary of the current robot state, its abstract parameters do not explicitly reveal where individual robot parts should appear in 3D space. We therefore complement SET with part-level 3D kinematic keypoint tokens before multi-modal interaction. Given the current joint estimate $\mathbf{J}$, forward kinematics maps it to robot-frame keypoints:

$$\tilde{\mathbf{P}}_K = FK(\mathbf{J}), \qquad \mathbf{P}_K = \mathbf{R}\tilde{\mathbf{P}}_K + \mathbf{t}. \tag{5}$$

Here, $\mathbf{t}$ denotes the current translation estimate, which is initialized by the image-induced root keypoint proposal at the first refinement step. Finally, we embed the camera-frame kinematic keypoints using the same keypoint embedding function as the image-induced proposals:

$$\mathbf{T}_{\text{KKT}} = \text{Embed}(\mathbf{P}_K) \in \mathbb{R}^{N\times C}. \tag{6}$$

The resulting 3D Kinematic Keypoint Tokens (KKT) represent the robot layout implied by the current state and provide the state-induced counterpart to the image-induced Keypoint Tokens. By embedding both sides in the same keypoint token space, KINO enables direct interaction between image-induced geometry and state-induced geometry. IKT and KKT are formed without explicit joint-ID embeddings or predefined token correspondences, while their interactions are learned from 3D spatial configurations together with state and structural context.

3.3 Multi-modal Context Tokenization

Canonical Structure Tokens. The image-induced 3D Keypoint Tokens and state-induced Kinematic 3D Keypoint Tokens provide local 3D geometric anchors for state-image association. However, robot pose and joint estimation also requires reasoning about the global articulated structure, such as how different robot parts are connected and constrained. Since the predicted keypoint proposals are expressed in the camera frame and entangled with the global pose, directly using them may make the structural representation sensitive to rotation. We therefore introduce Canonical Structure Tokens (CST) to encode pose-independent robot structure. Specifically, we root-center the proposals by subtracting the initial root prediction from each image-induced 3D keypoint proposal:

$$\mathbf{P}_c = \{\mathbf{P}_n - \mathbf{P}_1\}_{n=1}^{N}, \qquad \mathbf{T}_{\text{CST}} = \text{EVN}(\mathbf{P}_c) \in \mathbb{R}^{N\times C}. \tag{7}$$

where EVN follows the Vector Neurons-based construction of Neural Descriptor Fields (Deng et al., 2021; Simeonov et al., 2022). The resulting CST capture the structural relationships among robot parts while reducing the influence of global pose, providing complementary structure cues for subsequent multi-modal interaction.

Multi-modal Token Space. KINO then forms a compact multi-modal token space by collecting image evidence, image-induced geometry, state-induced geometry, and canonical structure:

$$\mathcal{T} = \{\mathbf{T}_{2D}, \mathbf{T}_{\mathrm{IKT}}, \mathbf{T}_{\mathrm{SET}}, \mathbf{T}_{\mathrm{KKT}}, \mathbf{T}_{\mathrm{CST}}\}. \tag{8}$$

Here, $\mathbf{T}_{2D}$ preserves dense image context, $\mathbf{T}_{\mathrm{IKT}}$ represents image-induced 3D geometry, $\mathbf{T}_{\mathrm{SET}}$ summarizes the current global pose-joint state, $\mathbf{T}_{\mathrm{KKT}}$ represents its implied robot layout, and $\mathbf{T}_{\mathrm{CST}}$ provides pose-independent structural cues. This token space is where KINO performs latent state-image association.

3.4 Latent State-Image Association via State Refinement

Given the multi-modal token space, KINO enters a dedicated state-refinement stage implemented by a six-layer multi-modal attention decoder. Each decoder layer performs one state-update step. Within each layer, we group the tokens into a 2D visual stream and a 3D state-geometry stream. The 2D stream contains $\mathbf{T}_{2D}$, while the 3D stream contains $\mathbf{T}_{\mathrm{SET}}$, $\mathbf{T}_{\mathrm{KKT}}$, $\mathbf{T}_{\mathrm{IKT}}$, and $\mathbf{T}_{\mathrm{CST}}$:

$$\mathbf{T}_{3D} = [\mathbf{T}_{\mathrm{SET}}; \mathbf{T}_{\mathrm{KKT}}; \mathbf{T}_{\mathrm{IKT}}; \mathbf{T}_{\mathrm{CST}}]. \tag{9}$$

Here, after each state update, the state-derived tokens $\mathbf{T}_{\mathrm{SET}}$ and $\mathbf{T}_{\mathrm{KKT}}$ are refreshed from the current pose-joint estimate, with SET re-embedded from the updated state and KKT regenerated through forward kinematics. In contrast, $\mathbf{T}_{\mathrm{IKT}}$ and $\mathbf{T}_{\mathrm{CST}}$ are encoded once from the image-induced keypoint proposals and are not regenerated during iterative refinement. Their hidden features are nevertheless updated by the self- and cross-attention operations at each decoder layer and propagated to the next layer. Each decoder stream first uses self-attention to model intra-domain dependencies:

$$\tilde{\mathbf{T}}_{2D} = \mathrm{SelfAttn}(\mathbf{T}_{2D}), \quad \tilde{\mathbf{T}}_{3D} = \mathrm{SelfAttn}(\mathbf{T}_{3D}). \tag{10}$$

It then performs bidirectional cross-attention to exchange information between image observations and 3D state-geometry tokens:

$$\hat{\mathbf{T}}_{3D} = \mathrm{CrossAttn}(\tilde{\mathbf{T}}_{3D}, \tilde{\mathbf{T}}_{2D}), \qquad \hat{\mathbf{T}}_{2D} = \mathrm{CrossAttn}(\tilde{\mathbf{T}}_{2D}, \tilde{\mathbf{T}}_{3D}). \tag{11}$$

Through this interaction, the state-induced KKT can attend to image evidence and image-induced geometry, enabling latent state-image association without repeatedly rendering the current state for image-space comparison, as in render-and-compare refinement (Li et al., 2020; Labbé et al., 2021). After multi-modal attention at layer ℓ, we denote the updated first token of the 3D stream as $\hat{\mathbf{T}}^{(\ell)}_{\mathrm{SET}} = \hat{\mathbf{T}}^{(\ell,1)}_{3D}$ and use it to predict the current robot state:

$$\mathbf{R}^{(\ell)} = \mathrm{MLP}_R(\hat{\mathbf{T}}^{(\ell)}_{\mathrm{SET}}), \qquad \mathbf{t}^{(\ell)} = \mathrm{MLP}_t(\hat{\mathbf{T}}^{(\ell)}_{\mathrm{SET}}), \qquad \mathbf{J}^{(\ell)} = \mathrm{MLP}_J(\hat{\mathbf{T}}^{(\ell)}_{\mathrm{SET}}). \tag{12}$$

The rotation head predicts a continuous 6D rotation representation (Zhou et al., 2019), which is orthonormalized via Gram–Schmidt to obtain a valid rotation matrix $\mathbf{R}^{(\ell)} \in SO(3)$. For $\ell < 6$, the updated state $(\mathbf{R}^{(\ell)}, \mathbf{t}^{(\ell)}, \mathbf{J}^{(\ell)})$ is re-embedded as SET and converted again into KKT following Sec. 3.2 before being passed to layer $\ell + 1$. The readouts at layers 1–5 serve only this state-to-token feedback path and receive no auxiliary supervision; the layer-6 readout is the final prediction. Thus, the six decoder layers correspond to six sequential state-update steps. The feedback path uses forward kinematics to regenerate 3D keypoints from each intermediate state estimate and embeds them as input to the next layer. The next state update therefore uses the 3D robot layout implied by the current estimate, rather than its parameter embedding alone.

3.5 Two-Stage Training Objectives

We optimize KINO in two stages, separating the pretraining of image-induced geometry from the training of the remaining framework.

Stage I: Root-depth and keypoint pretraining. We first pretrain the root-depth estimator and the 3D keypoint proposal network. This stage supervises the absolute root depth d, the camera-frame keypoint proposals $\mathbf{P}$, and their 2D projections $\mathbf{p}$ using

$$\mathcal{L}_{\mathrm{pre}} = \mathcal{L}_d + \lambda \mathcal{L}_P, \qquad \mathcal{L}_d = \|d - d_{\mathrm{gt}}\|_2^2, \ \mathcal{L}_P = \|\mathbf{P} - \mathbf{P}_{\mathrm{gt}}\|_2^2 + \|\mathbf{p} - \mathbf{p}_{\mathrm{gt}}\|_2^2. \tag{13}$$

Stage II: KINO framework training. After pretraining, we freeze the weights of the root-depth estimator and the 3D keypoint proposal network. The remaining KINO modules are then trained with the fixed image-induced geometry by supervising the final-layer rotation $\mathbf{R}^{(6)}$, translation $\mathbf{t}^{(6)}$, joint configuration $\mathbf{J}^{(6)}$, state-induced kinematic keypoints $\mathbf{P}_K^{(6)}$, and their 2D projections $\mathbf{p}_K^{(6)}$:

$$\mathcal{L}_{\text{KINO}} = \mathcal{L}_R + \mathcal{L}_t + \mathcal{L}_J + \lambda \mathcal{L}_K, \tag{14}$$

where $\mathcal{L}_R = \|\mathbf{R}^{(6)} - \mathbf{R}_{\text{gt}}\|_F^2$, $\mathcal{L}_t = \|\mathbf{t}^{(6)} - \mathbf{t}_{\text{gt}}\|_2^2$, $\mathcal{L}_J = \|\mathbf{J}^{(6)} - \mathbf{J}_{\text{gt}}\|_2^2$, and $\mathcal{L}_K = \|\mathbf{P}_K^{(6)} - \mathbf{P}_{\text{gt}}\|_2^2 + \|\mathbf{p}_K^{(6)} - \mathbf{p}_{\text{gt}}\|_2^2$. Here, $\mathbf{P}_{\text{gt}}$ and $\mathbf{p}_{\text{gt}}$ denote the ground-truth 3D keypoints and their 2D projections, respectively. In both stages, we set the keypoint-loss weight to $\lambda = 10$ and normalize predicted and ground-truth 2D keypoints by the input image size before computing the corresponding loss. During Stage II, only the final decoder readout receives direct supervision; earlier readouts affect the final prediction only through regenerated SET and KKT. Gradients from $\mathcal{L}_{\text{KINO}}$ update only the unfrozen KINO parameters and do not alter the pretrained depth or keypoint proposal networks.

4 Experiments

4.1 Experimental Settings

Datasets. We train KINO on the synthetic Franka Panda training split of DREAM, which provides RGB images annotated with 6-DoF pose and joint configurations (Lee et al., 2020). Evaluation uses the DREAM DR and Photo synthetic test sets together with the RealSense-Franka and AzureKinect-Franka real-world test sets from RoboKeyGen (Tian et al., 2024). The RoboKeyGen real-world test sets are more challenging than the DREAM test sets, as robots are often visually indistinct from the background or occupy only a small image region (Zhu et al., 2025).

Metrics. Following prior work (Xiang et al., 2018; Tian et al., 2024), we evaluate the complete articulated state using average distance (ADD). For image m, $\text{ADD}_m = \frac{1}{N}\sum_{n=1}^{N} \|\hat{\mathbf{P}}_{m,n} - \mathbf{P}_{\text{gt},m,n}\|_2$, and we report $\text{Mean} = \frac{1}{M}\sum_{m=1}^{M} \text{ADD}_m$. AUC is the normalized area under the empirical accuracy $\Pr(\text{ADD}_m < \tau)$ for $0 \le \tau \le 0.1\,\text{m}$, expressed as a percentage. Runtime is reported using both per-image inference latency and throughput in frames per second (FPS).

Comparison Protocol. RoboPose (Labbé et al., 2021), HPE (Ban et al., 2024), and RoboPEPP (Goswami et al., 2025) are rerun using official pretrained models trained on the same DREAM synthetic Panda split as KINO. For RoboTAG (Liu et al., 2026), whose official implementation is unavailable, we report the DR/Photo results from its original paper. RoboKeyGen (Tian et al., 2024) is not included in Table 1 because its reported results were obtained using a different training set from the baselines above, and its official implementation is unavailable. Implementation details are provided in Appendix A.

4.2 Main Results

	RealSense		AzureKinect		DR		Photo	
Method	AUC ↑	Mean ↓	AUC ↑	Mean ↓	AUC ↑	Mean ↓	AUC ↑	Mean ↓
RoboTAG†	–	–	–	–	82.5	–	84.3	–
RoboPose	37.96	0.122	44.64	0.082	82.77	0.022	79.45	0.032
HPE	46.40	0.056	40.19	0.061	82.68	0.018	81.98	0.018
RoboPEPP	54.77	0.077	50.15	0.066	84.44	0.020	82.66	0.023
KINO (Ours)	**62.04**	**0.039**	**59.61**	**0.041**	**86.45**	**0.014**	**85.83**	**0.014**

Table 1: ADD AUC (%) and Mean (m). KINO results average three training seeds (Appendix C). RoboPose, HPE, and RoboPEPP use official pretrained weights and inference code. †: reported results from the original paper, provided for cross-paper context. "–" denotes unavailable entries.

We evaluate KINO in the unknown-state setting of prior work (Labbé et al., 2021; Ban et al., 2024; Goswami et al., 2025; Liu et al., 2026), with forward kinematics available for constructing Kinematic Keypoint Tokens (KKT). As shown in Tab. 1, KINO outperforms all three rerun baselines on all four benchmarks. On RealSense and AzureKinect, KINO reaches AUC scores of 62.04 and 59.61, improving over the strongest rerun baseline, RoboPEPP, by 7.27 and 9.46 points, respectively. On DR and Photo, KINO reaches AUC scores of 86.45 and 85.83, exceeding RoboPEPP by

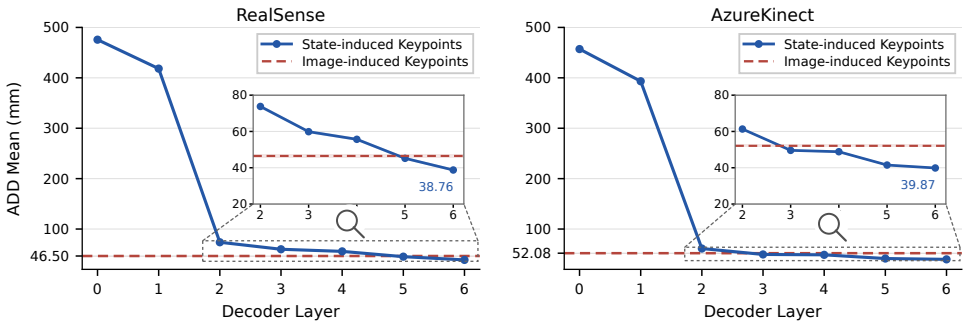


Figure 3: Decoder-layer keypoint ADD to ground truth on RealSense and AzureKinect. Blue curves show the mean ADD of state-induced keypoints; red dashed lines show the fixed image-induced-keypoint ADD. Layer 0 denotes the initialization before decoder refinement; insets enlarge Layers 2–6. Results are obtained using random seed 808.



Figure 4: **Qualitative results.** Two real-world examples from RealSense and Azure Kinect compare the robot skeleton predictions of HPE and KINO, with enlarged views of the wrist and base regions and single-frame ADD computed over all keypoints. HPE and KINO predictions are shown in orange and blue, respectively; white dashed skeletons with hollow markers indicate ground truth. More comparisons are provided in the supplementary video.

2.01 and 3.17 points, respectively. The gains support keypoint-mediated state-image association. We next examine how the state-induced geometry evolves across the six refinement steps.

For this layer-wise analysis, we use the model trained with random seed 808. At initialization and after each decoder layer, we compute the seven-keypoint ADD of the regenerated state-induced keypoints to ground truth and compare it with the fixed image-induced-keypoint ADD (Fig. 3). The state-induced error decreases over successive updates on both cameras and, by Layer 6, reaches 38.76 mm on RealSense and 39.87 mm on AzureKinect, below the fixed image-induced-keypoint ADD values of 46.50 mm and 52.08 mm, respectively. The decreasing state-induced keypoint error suggests that the refinement is not limited to reproducing the fixed image-induced keypoint estimates. This trend complements the token-removal and regeneration studies in Sec. 4.3. Figure 4 compares HPE and KINO in two real-world examples from RealSense and AzureKinect.

4.3 Ablation Studies

To enable controlled comparisons, all ablation experiments use random seed 808. Under this setting, the full KINO model achieves AUC scores of 62.59 and 60.38 on RealSense and AzureKinect, respectively. Tables 2 and 3 use matched controls to separately examine KINO's geometric token representation and state-to-token feedback, while retaining the same visual backbone and refinement decoder across the corresponding variants.

Token Families. SET is retained in all variants because it carries the pose-joint state updated at every decoder layer. As shown in Table 2, IKT is the most critical token family: removing it reduces

Variant	RS	AK	Δ
2D+SET, no refresh	12.77	25.83	–
Separate IKT/KKT emb.	58.65	57.82	-3.25
w/o $\mathbf{T}_{2D}$	60.40	56.27	-3.15
w/o CST	59.67	57.89	-2.71
w/o KKT	55.24	53.14	-7.30
w/o IKT	35.61	32.75	-27.31

Table 2: Ablation of KINO's token space on the real-world benchmarks. The first row is a direct state-regression control using only 2D Visual Tokens and SET; the second replaces the shared IKT/KKT projection with separate affine projections; the remaining rows are leave-one-family-out variants. For all rows except the first, Δ is the mean RS/AK AUC change relative to full KINO.

Variant	RS	AK	Δ
All tokens, no refresh	56.98	51.42	–
SET refresh only	56.05	53.73	+0.69
KKT refresh only	60.04	59.37	+5.51
SET+KKT refresh	**62.59**	**60.38**	+7.29

Table 3: State-to-token feedback ablation on the real-world benchmarks. Δ is the mean RS/AK AUC change relative to "All tokens, no refresh."

mean real-world AUC by 27.31 points, identifying image-induced 3D keypoints as the primary observation anchors. Removing KKT causes a 7.30-point drop, showing the importance of state-implied geometry for associating observed and hypothesized robot structure. Using separate affine projections for IKT and KKT reduces mean AUC by 3.25 points, supporting the shared embedding that places both geometries in a common token space. CST and 2D Visual Tokens provide complementary cues: removing them decreases mean AUC by 2.71 and 3.15 points, respectively, with the latter reducing performance from 61.49 to 58.34. This relatively modest drop without 2D Visual Tokens suggests that IKT conveys most image-side information needed for state estimation, while dense visual features provide additional context. Finally, the 2D+SET no-refresh control reaches only 19.30 mean AUC, showing that it needs geometric keypoint tokens.

Kinematics-driven State-to-Token Feedback. The SET-only and KKT-only variants independently refresh the corresponding state-derived tokens after each intermediate state update. With all token families retained, disabling both refresh operations gives a mean real-world AUC of 54.20. Refreshing SET alone, which iteratively feeds the updated pose-joint state back to subsequent layers without regenerating FK-based geometry, yields only a modest improvement to 54.89 (+0.69). In contrast, regenerating KKT increases the mean AUC to 59.71 (+5.51). Regenerating both SET and KKT achieves the best performance of 61.49 (+7.29). These controls use the same decoder architecture and show that regenerating KKT is more effective than refreshing SET alone. Geometric regeneration is therefore the main contributor to the gain from state-to-token feedback.

Additional Analyses. Appendix B compares inference speed; Appendix C reports training-seed stability; Appendix D evaluates additional robot morphologies; and Appendix E reports joint-angle, rotation, and translation errors, with improvements over HPE in all three metrics on both real-world test sets. Appendix F compares KINO with constrained fitting of the same frozen keypoint proposals (Table 8); KINO leads the eight-start fit by 6.29 AUC on RealSense and 11.50 on AzureKinect.

5 Conclusion

We introduced **KINO**, which reformulates single-view robot pose and joint estimation as reasoning about state-image consistency in a latent space. The shared 3D keypoint representation enables direct interaction between image-induced and state-induced geometry. Kinematics-driven feedback supplies the updated state's 3D geometry to subsequent refinement. Experiments on RoboKeyGen and DREAM demonstrate improved accuracy over representative baselines. Ablation studies confirm the contributions of the shared geometric representation and kinematics-driven feedback. KINO also outperforms constrained fitting based on the same frozen keypoint proposals.

AI USE STATEMENT

In preparing this manuscript, we used ChatGPT for language editing, translation, LaTeX formatting, and code assistance. ChatGPT was not used to generate or determine the core research ideas, theoretical claims, methodology, experimental design, data, reported results, or their interpretation. All AI-assisted outputs were reviewed and verified by the authors. The authors take full responsibility for the final content of this submission.

REFERENCES

Shikun Ban, Juling Fan, Xiaoxuan Ma, Wentao Zhu, Yu Qiao, and Yizhou Wang. Real-time holistic robot pose estimation with unknown states. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Sammy Christen, Wei Yang, Claudia Pérez-D'Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. Learning human-to-robot handovers from point clouds. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Charles Champagne Cossette, Mohammed Ayman Shalaby, David Saussié, Jérôme Le Ny, and James Richard Forbes. Optimal multi-robot formations for relative pose estimation using range measurements. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Norman Di Palo, Arunkumar Byravan, Leonard Hasenclever, Markus Wulfmeier, Nicolas Heess, and Martin Riedmiller. Towards a unified agent with foundation models. In *ICLR 2023 Workshop on Reincarnating Reinforcement Learning*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representation (ICLR)*, 2021.

M. Fiala. Artag, a fiducial marker system using digital techniques. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research (IJRR)*, 44 (5):701–739, 2025. doi: 10.1177/02783649241281508.

Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2003.

Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco Madrid-Cuevas, and Manuel Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition (PR)*, 2014.

Raktim Gautam Goswami, Prashanth Krishnamurthy, Yann LeCun, and Farshad Khorrami. RoboPEPP: Vision-based robot pose and joint angle estimation through embedding predictive pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Radu Horaud and Fadi Dornaika. Hand-eye calibration. *The International Journal of Robotics Research (IJRR)*, 1995.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

Haobo Jiang, Mathieu Salzmann, Zheng Dang, Jin Xie, and Jian Yang. Se(3) diffusion model-based point cloud registration for robust 6d object pose estimation. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Single-view robot pose and joint angle estimation via render & compare. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Timothy E. Lee, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Oliver Kroemer, Dieter Fox, and Stan Birchfield. Camera-to-robot pose estimation from a single image. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2020.

Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision (IJCV)*, 2009.

Shushuai Li, Christophe De Wagter, and Guido C. H. E. De Croon. Self-supervised monocular multi-robot relative localization with efficient deep neural networks. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6d pose estimation. *International Journal of Computer Vision (IJCV)*, 128(3):657–678, 2020. doi: 10.1007/s11263-019-01250-9.

Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A. Vela, and Stan Birchfield. Single-stage keypoint-based category-level object pose estimation from an RGB image. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022a.

Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A. Vela, and Stan Birchfield. Keypoint-based category-level object pose tracking from an RGB sequence with uncertainty estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022b.

Yifan Liu, Fangneng Zhan, Wanhua Li, Haowen Sun, Katerina Fragkiadaki, and Hanspeter Pfister. RoboTAG: End-to-end robot configuration estimation via topological alignment graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jingpei Lu, Florian Richter, and Michael C. Yip. Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jingpei Lu, Zekai Liang, Tristin Xie, Florian Richter, Shan Lin, Sainan Liu, and Michael C Yip. CtRNet-X: Camera-to-robot pose estimation in real-world conditions using a single camera. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Accurate non-iterative o(n) solution to the pnp problem. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2011.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Karl Pauwels, Leonardo Rubio, and Eduardo Ros. Real-time model-based articulated object pose detection and tracking with variable rigidity constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B. Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: SE(3)-equivariant object representations for manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

Mikael Svenstrup, Søren Tranberg Hansen, Hans Jørgen Andersen, and Thomas Bak. Pose estimation and adaptive robot behaviour for human-robot interaction. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

Yang Tian, Jiyao Zhang, Zekai Yin, and Hao Dong. Robot structure prior guided temporal attention for camera-to-robot pose estimation from image sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Yang Tian, Jiyao Zhang, Guowei Huang, Bin Wang, Ping Wang, Jiangmiao Pang, and Hao Dong. Robokeygen: Robot pose and joint angles estimation via diffusion-based 3d keypoint generation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

B Wen and Kostas E Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. BundleSDF: Neural 6-DoF tracking and 3D reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Proceedings of the Robotics: Science and Systems (RSS)*, 2018.

Wei Yang, Chris Paxton, Arsalan Mousavian, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. Reactive human-to-robot handovers of arbitrary objects. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Zhenjun Ying and Qinghua Zhou. CREPES2: A dynamic position and pose estimation system. In *Proceedings of IEEE International Conference on Control, Robotics and Cybernetics (CRC)*, 2023.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Kangjian Zhu, Haobo Jiang, Yigong Zhang, Jianjun Qian, Jian Yang, and Jin Xie. MonoSE(3)-Diffusion: A monocular SE(3) diffusion framework for robust camera-to-robot pose estimation. *IEEE Robotics and Automation Letters (RA-L)*, 2025.

A Implementation Details

We train KINO in two stages. Stage I pretrains the root-depth estimator and 3D keypoint proposal network for 100 epochs using Adam (Kingma & Ba, 2015), zero weight decay, a learning rate of 1×10^{-4}, and a batch size of 64. Stage II freezes these networks and trains the remaining modules for 80 epochs with a batch size of 48; the learning rate linearly warms from 2×10^{-5} to 1×10^{-4} over the first 5 epochs, stays fixed through epoch 50, and then decays by 0.95 per epoch. We set $\lambda = 10$ in both stages and follow HPE (Ban et al., 2024) for image preprocessing and augmentation. The root-depth and keypoint-proposal networks use HRNet-W32 and ResNet-50, respectively, while the image encoder is a DINO-initialized ViT-B/16 (Caron et al., 2021) with 768-dimensional embeddings, 12 layers, 12 heads, and patch size 16. IKT and KKT share an affine projection from $\mathbb{R}^3$ to $\mathbb{R}^{1024}$, applied independently to each keypoint without a hidden layer or nonlinearity. SET linearly projects $[\mathbf{r}_1; \mathbf{r}_2; \mathbf{t}; \mathbf{J}]$ to the same 1024-dimensional space, where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the first two rotation-matrix columns forming the 6D rotation representation. The refinement decoder has six 1024-dimensional, eight-head attention layers, each performing one state update. Our experiments use PyTorch (Paszke et al., 2019), CUDA 11.7, and a single NVIDIA TITAN GPU.

B Inference Speed

Tab. 4 separates the efficiency reported in the HPE paper from our local measurements because runtime is hardware dependent. On an NVIDIA Tesla V100S with batch size 1, HPE reports 44.3 ms per image (22.6 FPS), compared with 570.8 ms (1.8 FPS) for RoboPose (Labbé et al., 2021; Ban et al., 2024). On our NVIDIA TITAN RTX, we remeasure HPE and KINO under the same local runtime environment. HPE requires 54.36 ms per image (18.40 FPS), while full KINO (SET+KKT refresh) requires 95.53 ms per image (10.47 FPS). Full KINO improves ADD AUC over HPE by 3.77 and 3.85 points on DR and Photo, and by 15.64 and 19.42 points on RealSense and AzureKinect, respectively, using the three-seed averages in Tab. 1.

For a lower-latency operating point, the "All tokens, no refresh" variant in Tab. 3 retains all token families and the six-layer decoder but disables SET re-embedding and KKT regeneration between layers. It requires 71.49 ms per image (13.99 FPS), reducing latency by 25.2% relative to full KINO. With seed 808, this variant achieves ADD AUC scores of 56.98 and 51.42 on RealSense and AzureKinect, exceeding HPE's 46.40 and 40.19 by 10.58 and 11.23 percentage points, respectively (10.91 points on average). Full refresh adds 24.04 ms and improves mean real-world AUC by a further 7.29 points under the same seed-808 ablation, quantifying the accuracy-latency trade-off of state-to-token feedback.

Method	GPU	Time (ms)	FPS
RoboPose (reported)	V100S	570.8	1.8
HPE (reported)	V100S	44.3	22.6
HPE (local)	TITAN RTX	54.36	18.40
KINO (all tokens, no refresh)	TITAN RTX	71.49	13.99
KINO (full refresh)	TITAN RTX	95.53	10.47

Table 4: Inference efficiency. Reported values are taken from HPE (Ban et al., 2024) and were measured on an NVIDIA Tesla V100S with batch size 1. Local measurements are obtained on our NVIDIA TITAN RTX under the same runtime environment, with HPE and both KINO variants remeasured locally. Results across the V100S and TITAN RTX groups are provided for context and are not directly comparable.

These are per-image inference timings, not end-to-end control latencies; neither KINO variant meets a 30-Hz visual-update requirement on this hardware. For applications that tolerate lower-rate visual updates, a deployment could run KINO asynchronously from a higher-rate controller, process only the latest frame to avoid queue buildup, and use timestamp-aware state prediction between visual updates. Such integration would still require task-specific validation of end-to-end latency and closed-loop stability; it is not evaluated here.

C Stability across Training Seeds

To assess sensitivity to optimization randomness, we train KINO independently with seeds 808, 809, and 810 using the same experimental configuration. Tab. 5 reports the ADD AUC and ADD Mean obtained by each run across the four evaluation datasets. Using the unrounded scores, the largest AUC range is 1.45 percentage points on AzureKinect, while the ranges on RealSense, DR, and Photo are 1.25, 0.84, and 0.81 points, respectively. This agreement across both synthetic and real domains indicates that KINO is stable under the tested training-seed variation.

Dataset	Metric	Seed 808	Seed 809	Seed 810	Average
DR	AUC (%)	86.96	86.28	86.12	**86.45**
DR	Mean (m)	0.0136	0.0143	0.0145	**0.0141**
Photo	AUC (%)	86.35	85.60	85.54	**85.83**
Photo	Mean (m)	0.0140	0.0147	0.0148	**0.0145**
RS	AUC (%)	62.59	62.20	61.34	**62.04**
RS	Mean (m)	0.0388	0.0392	0.0402	**0.0394**
AK	AUC (%)	60.38	59.52	58.94	**59.61**
AK	Mean (m)	0.0399	0.0407	0.0413	**0.0406**

Table 5: Per-seed ADD results for three independently trained KINO runs. AUC is reported in percent and Mean in meters; RS and AK denote RealSense and AzureKinect, respectively. All entries are rounded for display; Average is the arithmetic mean of the unrounded per-seed values.

D Applicability to Additional Robot Morphologies

We further test whether the advantage of KINO extends beyond the Panda morphology by independently training robot-specific models for the 7-DoF Kuka and 15-DoF Baxter on their corresponding URDFs and DREAM synthetic training sets. Table 6 compares KINO with HPE on three robot-domain pairs: KINO improves ADD AUC by 4.9–6.2 points, showing consistent gains across morphologies. HPE values are reported results on the corresponding DREAM test sets.

Method	Kuka DR	Kuka Photo	Baxter DR
HPE	75.1	73.9	58.8
KINO	**80.6**	**80.1**	**63.7**

Table 6: ADD AUC (%) on additional DREAM robots.

E Joint-Angle, Rotation, and Translation Accuracy

To complement the aggregate geometric evaluation by ADD, we separately evaluate joint configuration, global rotation, and XYZ translation on the DREAM Franka Panda DR and Photo test sets and the RoboKeyGen RealSense-Franka and AzureKinect-Franka test sets. Joint MAE averages absolute angular errors over joints 1–6 and all test images, while rotation error measures the mean angular error of the estimated global robot orientation; both are reported in degrees. XYZ translation error measures the error of the predicted camera-frame 3D translation and is reported in millimeters. We evaluate full KINO (SET+KKT refresh) trained with seed 808 and compare it with HPE (Ban et al., 2024).

As shown in Tab. 7, KINO reduces all three errors relative to HPE on both real-world test sets. Joint MAE decreases from 19.16° to 11.51° on RealSense and from 16.04° to 12.41° on AzureKinect, corresponding to relative reductions of 39.9% and 22.6%, respectively. Rotation error decreases by 34.0% and 32.3%, respectively, while XYZ translation error decreases from 41.86 to 36.08 mm on RealSense and from 35.38 to 34.70 mm on AzureKinect, with a smaller translation gain on the latter. These component-wise results complement the ADD gains in Tab. 1 and demonstrate more accurate

Dataset	Method	Joint MAE (°) ↓	Rotation (°) ↓	Translation (mm) ↓
DR	HPE	**4.4417**	3.2466	10.7528
DR	KINO	4.7469	**2.8670**	**9.3970**
Photo	HPE	**4.4534**	3.3578	10.7014
Photo	KINO	4.5670	**3.1488**	**9.3230**
RealSense	HPE	19.1621	3.6895	41.8623
RealSense	KINO	**11.5132**	**2.4352**	**36.0810**
AzureKinect	HPE	16.0358	5.0918	35.3831
AzureKinect	KINO	**12.4089**	**3.4468**	**34.6982**

Table 7: Joint-angle, rotation, and XYZ translation errors on the four Franka Panda test sets (lower is better). Joint MAE is averaged over joints 1–6; KINO uses SET+KKT refresh and training seed 808. The lower error within each dataset is highlighted in bold for each metric.

recovery of both global pose and joint configuration on real images, rather than an improvement in aggregate keypoint geometry alone.

On DR and Photo, KINO also reduces rotation and translation errors, whereas joint MAE is slightly higher than HPE by 0.31° and 0.11°, respectively. Thus, the substantial joint-angle improvements occur on the real-world test sets; the synthetic results show better global pose accuracy with slightly higher joint-angle errors.

F Constrained Fitting of Frozen Keypoint Proposals

F.1 Comparison Protocol and Results

An improvement over image-induced keypoint proposals alone does not distinguish learned refinement from fitting those proposals to a kinematic model. We therefore evaluate a direct baseline using *exactly the same frozen camera-frame proposals* $\mathbf{P}$ as KINO.

Constrained Fitting. For each image, we minimize

$$E_{\text{fit}}(\mathbf{R}, \mathbf{t}, \mathbf{J}; \mathbf{P}) = \sum_{n=1}^{N} \|\mathbf{R}\,\mathrm{FK}(\mathbf{J})_n + \mathbf{t} - \mathbf{P}_n\|_2^2, \tag{15}$$

subject to $\mathbf{R} \in \mathrm{SO}(3)$ and component-wise URDF joint bounds $\mathbf{J}_{\min} \le \mathbf{J} \le \mathbf{J}_{\max}$, with unrestricted $\mathbf{t} \in \mathbb{R}^3$. For each candidate $\mathbf{J}$, proper rigid alignment solves $\mathbf{R}, \mathbf{t}$ analytically; bounded nonlinear least squares then optimizes the joint variables. We evaluate one and eight starts, selecting the latter's output solely by final proposal residual, not ground-truth error. The following subsections give the reduced objective and selection rule.

	RealSense		AzureKinect	
Method	AUC ↑	Mean ↓	AUC ↑	Mean ↓
Raw proposals	55.4060	0.046499	48.7035	0.052078
Geometric fitting, one start	56.1042	0.045825	48.7881	0.051981
Geometric fitting, eight starts	56.3034	0.045618	48.8842	0.051879
KINO	**62.5928**	**0.038757**	**60.3837**	**0.039867**

Table 8: Identical-proposal geometric fitting. ADD AUC (%) and Mean (m). Raw proposals are evaluated directly; fitted outputs and KINO are evaluated through their estimated states. KINO reports the matched-proposal evaluation, not Table 1's three-seed average.

Comparison with KINO. On RealSense and AzureKinect, single-start fitting gains 0.70 and 0.08 AUC points over raw proposals; eight starts gain 0.90 and 0.18 points, respectively (Table 8). KINO exceeds the eight-start fit by a further 6.29 and 11.50 points and reduces ADD Mean by 6.861 and 12.012 mm, corresponding to relative reductions of 15.0% and 23.2%. The fitting baseline minimizes distance to $\mathbf{P}$ under kinematic constraints, whereas KINO learns its updates from state and

geometric supervision, with additional visual and structural context. A state close to noisy proposals need not be closest to ground truth; the tested fitting procedure does not reproduce the complete estimator's accuracy.

Scope of the Comparison. This comparison controls proposal quality, not downstream supervision or access to dense visual features, and does not isolate refresh alone. Table 3 provides that complementary matched-decoder evidence. Eight local starts do not certify a global optimum, nor does the comparison rule out alternative fitting formulations.

F.2 Eliminating Rigid Pose for Fixed Joints

The fitting baseline uses exactly the same frozen image-induced proposals $\mathbf{P}$ as KINO. For a candidate joint configuration $\mathbf{J}$, let $\mathbf{Q}_n(\mathbf{J}) = \mathrm{FK}(\mathbf{J})_n$ and define

$$\bar{\mathbf{Q}}(\mathbf{J}) = \frac{1}{N}\sum_n \mathbf{Q}_n(\mathbf{J}), \qquad \bar{\mathbf{P}} = \frac{1}{N}\sum_n \mathbf{P}_n. \tag{16}$$

The optimal translation for a fixed rotation is $\mathbf{t} = \bar{\mathbf{P}} - \mathbf{R}\bar{\mathbf{Q}}(\mathbf{J})$. The remaining rigid alignment is constrained to a proper rotation:

$$\mathbf{R}^*(\mathbf{J}) \in \operatorname*{arg\,min}_{\mathbf{R}\in \mathrm{SO}(3)} \sum_n \|\mathbf{R}(\mathbf{Q}_n(\mathbf{J}) - \bar{\mathbf{Q}}(\mathbf{J})) - (\mathbf{P}_n - \bar{\mathbf{P}})\|_2^2, \tag{17}$$

$$\mathbf{t}^*(\mathbf{J}) = \bar{\mathbf{P}} - \mathbf{R}^*(\mathbf{J})\bar{\mathbf{Q}}(\mathbf{J}). \tag{18}$$

Solving this subproblem analytically removes rotation and translation from the joint search. The proper-rotation constraint excludes reflections; no scale variable is fitted.

F.3 Bounded Joint Optimization and Proposal-Only Selection

After rigid alignment, define the joint residual and reduced objective

$$\begin{aligned} \mathbf{r}(\mathbf{J};\mathbf{P}) &= \mathrm{stack}_{n=1}^{N}[\mathbf{R}^*(\mathbf{J})\mathbf{Q}_n(\mathbf{J}) + \mathbf{t}^*(\mathbf{J}) - \mathbf{P}_n], \\ \bar{E}_{\mathrm{fit}}(\mathbf{J};\mathbf{P}) &= \|\mathbf{r}(\mathbf{J};\mathbf{P})\|_2^2. \end{aligned} \tag{19}$$

Bounded nonlinear least squares optimizes $\mathbf{J}$ subject to the component-wise URDF bounds $\mathbf{J}_{\min} \le \mathbf{J} \le \mathbf{J}_{\max}$. Single-start fitting returns one solution. Eight-start fitting optimizes from eight initial joint configurations and selects the candidate with the smallest final proposal residual:

$$a^* = \operatorname*{arg\,min}_{a\in\{1,\dots,8\}} \bar{E}_{\mathrm{fit}}(\hat{\mathbf{J}}^{[a]};\mathbf{P}). \tag{20}$$

The returned state is $\hat{\mathbf{J}} = \hat{\mathbf{J}}^{[a^*]}$, $\hat{\mathbf{R}} = \mathbf{R}^*(\hat{\mathbf{J}})$, and $\hat{\mathbf{t}} = \mathbf{t}^*(\hat{\mathbf{J}})$. Selection does not use ground-truth ADD, AUC, or state error. Annotations are used only to evaluate the returned geometry under the metrics in Section 4. Eight starts constitute a multi-start local optimization baseline, not a certificate of global optimality.

F.4 Controls and Interpretation

All four rows of Table 8 use identical proposals. The raw reference evaluates $\mathbf{P}$ directly; the fitted variants and KINO evaluate the camera-frame keypoints reconstructed from their estimated pose and joints via forward kinematics. The fitting residual uses indexed proposal/FK correspondences. Eight-start fitting reduces ADD Mean over the raw proposals by 0.881 mm on RealSense and 0.199 mm on AzureKinect. KINO reduces it by a further 6.861 and 12.012 mm (15.0% and 23.2% relative to the fit). The comparison controls proposal quality, not all downstream information or learning. Fitting uses the fixed coordinate objective and kinematic constraints; KINO additionally uses supervised updates and visual/structural context. Table 8 consequently tests this fitting alternative against the complete estimator, while Table 3 separately tests geometric refresh within a matched decoder.