STRUCTURE-GUIDED ALTERNATING INFERENCE FOR MONOCULAR ROBOT STATE REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating the pose and joint angles of an articulated robot from a single RGB image supports spatial reasoning without access to its internal measurements. Existing approaches use learned visual representations to predict the robot state, sometimes with iterative refinement. However, articulation and global pose can produce similar image changes, making it difficult to determine which variable should correct a remaining discrepancy. To address this, we introduce Structure-Guided Alternating Inference (SGAI), a training-free method that formulates state refinement as bounded geometric optimization using robot geometry and perceptual cues extracted by visual foundation models. Specifically, SGAI solves separate articulation and pose subproblems while holding the other block fixed, preventing compensating changes between the two blocks within each solve. Point-to-link, silhouette, and projection anchor constraints assign geometric evidence to the appropriate updates and limit corrections driven by imperfect depth and foreground predictions. Re-rendering and renewed geometric associations allow each block to use the shape or placement corrected by the other. Experiments on DREAM show improved 3D localization in most tested settings across two initializers and three robot models, using a shared configuration without additional training or initializer- or robot-specific retuning.

1 INTRODUCTION

Recovering a robot's state from an external RGB camera enables spatial reasoning without access to joint measurements. The state consists of joint angles Φ and a robot-to-camera transformation T_R^C: the former determines the articulated shape, and the latter places that shape in the scene. Estimating them together is difficult because self-occlusion and perspective projection can make different three-dimensional states look similar (Labbé et al., 2021). A useful estimate must therefore explain the image while remaining consistent with the robot's geometry.

Learning-based methods provide strong initial estimates. DREAM recovers pose from detected keypoints when joint angles are known (Lee et al., 2020); RoboPose (Labbé et al., 2021), holistic estimation (Ban et al., 2024), and RoboPEPP (Goswami et al., 2025) also address unknown articulation. RoboTAG organizes robot and camera variables through a graph for cross-branch consistency learning (Liu et al., 2026). Their predictions can nevertheless retain spatial errors. This raises an inference problem: given an initial state, how should we optimize its joint angles and pose to better explain the observations while respecting the robot model?

The challenge is to decide which state variable should explain a discrepancy. A joint rotation can be compensated by an opposing rigid motion, so reducing an image-space residual does not necessarily identify the required articulation and pose corrections. Visual foundation models offer dense depth and foreground cues for refinement, but these predictions are imperfect: depth can be biased, support can be incomplete, and observations near adjacent links can be ambiguously associated (Lin et al., 2025; Mei et al., 2026). Refinement consequently needs a rule for assigning evidence to the articulated structure and for controlling how that evidence changes the state.

To address this, we propose *Structure-Guided Alternating Inference (SGAI)*, which formulates refinement as two bounded optimization subproblems and connects their solutions through geometric feedback (Figure 1). With pose fixed, the articulation step asks how the robot's shape should change at its current placement. With articulation fixed, the pose step asks how that shape should move as a

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.

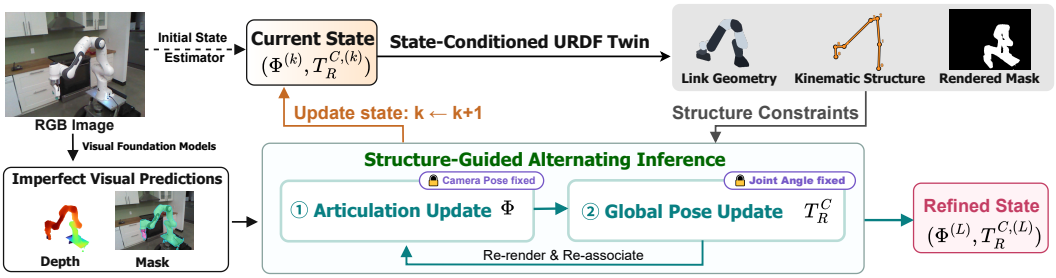


Figure 1: **Structure-guided refinement of monocular robot state estimates.** SGAI uses robot geometry and visual foundation-model cues to refine an existing estimate of articulation and global pose. The illustrated loop separates their updates while connecting successive corrections through geometric feedback. One refinement configuration is shared across the tested initializers and robot models.

whole. Fixing one block excludes compensating articulation–pose changes within a solve. The updated shape then guides pose alignment, and the updated placement guides point-to-link association in the next cycle. A local projection-Jacobian diagnostic examines the ambiguity behind this choice; controlled ablations compare it with simultaneous updates. The two objectives are built from three complementary constraints, which make depth and mask predictions usable as evidence without requiring either to be exact. Re-rendering refreshes the geometric context after state changes, while joint limits, sensitivity-dependent ranges, and local acceptance rules bound the corrections. The visual predictions themselves remain fixed throughout refinement. Experiments show that SGAI uses one refinement configuration for all baseline methods across three robot models in DREAM. SGAI thus offers a structure-guided refinement framework for monocular robot state estimation, enabling existing estimators to benefit from robot geometry and foundation-model perception using one fixed configuration.

Our contributions are threefold:

1) **Structure-guided state correction.** Three consistency constraints assign imperfect depth and foreground evidence to articulated shape and global placement, enabling bounded corrections without learning another state-update network.
2) **Alternating inference with geometric feedback.** Conditional updates exclude within-step articulation–pose compensation, while refreshed geometry connects successive corrections.
3) **Shared refinement with two initializers and three robot models on DREAM.** One fixed set of objective weights and solver settings improves 3D localization in 6/9 settings for RoboPose and 7/9 for RoboPEPP, with small decreases elsewhere.

2 Related Work

Camera-to-robot pose estimation. With known joint configurations, image keypoints can be paired with forward-kinematic locations for pose recovery, as in early synthetic-data and few-shot methods (Lambrecht & Kästner, 2019; Lambrecht, 2019) and DREAM (Lee et al., 2020). Subsequent work improves keypoint selection (Lu et al., 2022), shape support (Lambrecht et al., 2021), partial-visibility handling (Zhong et al., 2023), feature fusion (Han et al., 2024), and temporal reasoning (Tian et al., 2023). CtRNet and CtRNet-X address transfer to real images and partially visible robots (Lu et al., 2023b; 2025); MonoSE(3)-Diffusion learns iterative pose denoising (Zhu et al., 2025). PK-ROKED explicitly models uncertain kinematic priors (Klüpfel et al., 2025). These methods motivate geometric pose recovery, but our refinement must also correct unknown articulation rather than rely on an externally supplied joint configuration.

Articulation and robot state estimation. Depth-based approaches estimate robot joints through part classification (Bohg et al., 2014), decoupled heatmaps (Simoni et al., 2022), or temporal prediction (Simoni et al., 2025). For monocular RGB, RoboPose learns articulated render-and-compare updates (Labbé et al., 2021), holistic estimation predicts complementary state quantities (Ban et al., 2024), and RoboKeyGen generates 3D keypoints by diffusion (Tian et al., 2024). Distance-

geometric joint recovery (Bilić et al., 2023) and GISR combine kinematic structure with learned estimation; GISR adds iterative silhouette-based refinement (Bilić et al., 2024). RoboPEPP improves representation learning through embedding prediction (Goswami et al., 2025), while RoboTAG uses a topological graph to connect robot and camera variables (Liu et al., 2026). SGAI complements these learned state estimators: it operates on their output state without accessing internal features or retraining, and uses alternating geometric corrections to address residual articulation–pose ambiguity.

Geometric refinement and robot rendering. Render-and-compare inference underlies rigid-object methods such as DeepIM and FoundationPose (Li et al., 2018; Wen et al., 2024). Differentiable rendering (Liu et al., 2019; Ravi et al., 2020) also supports robot pose and shape estimation (Lu et al., 2023a; Liang et al., 2025), automatic calibration with EasyHeC and its training-free extension EasyHeC++ (Chen et al., 2023; Hong et al., 2024), and stereo robot tracking (Hao et al., 2026). Hydra instead registers RGB-D observations for hand-eye calibration (Huber et al., 2025). Dr. Robot and Prof. Robot develop differentiable robot representations for visual state and action optimization (Liu et al., 2024; Ruan et al., 2025). SGAI uses the perception capabilities of Depth Anything 3 and RobotSeg (Lin et al., 2025; Mei et al., 2026) to obtain auxiliary evidence from a single RGB image. Structure-specific conditional objectives then use this evidence to refine unknown articulation and pose.

3 METHODOLOGY

3.1 PROBLEM DESCRIPTION AND METHOD OVERVIEW

The input is a single RGB image I, camera intrinsics K, a robot model $\mathcal{S}$, and an initial state $(\Phi^{(0)}, T_R^{C,(0)})$ from an existing estimator. The unknowns are the n joint angles $\Phi \in \mathbb{R}^n$ and the robot-to-camera transformation $T_R^C = (R, \mathbf{t}) \in SE(3)$, where $R \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ map a point from the robot base frame to the camera frame as $\mathbf{x}^C = R\mathbf{x}^R + \mathbf{t}$. The URDF model $\mathcal{S}$ supplies link meshes, forward kinematics, and admissible joint intervals.

We seek to refine articulation and global pose jointly so that the estimated robot agrees with the image while respecting its kinematic structure:

$$(\Phi^\star, T_R^{C,\star}) \in \underset{\Phi \in \mathcal{Q},\ T_R^C \in SE(3)}{\arg\min} \mathcal{E}(\Phi, T_R^C; I, K, \mathcal{S}), \tag{1}$$

Here $\mathcal{Q} = \prod_{j=1}^n [\Phi_j^{\min}, \Phi_j^{\max}]$ is the URDF-admissible joint domain, and $\mathcal{E}$ denotes visual-geometric consistency. This formulation defines the joint estimation goal; SGAI pursues it through the conditional objectives introduced in Section 3.1.

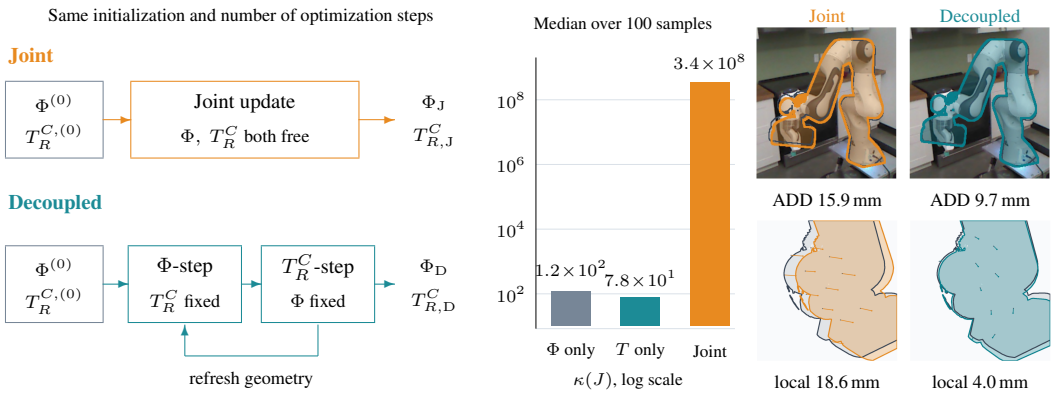
Why alternating inference? Equation 1 involves an $(n+6)$-dimensional search over articulation and pose. The difficulty is not only the dimension: a joint rotation and an opposing rigid motion can produce nearly identical image projections, leaving the required correction ambiguous. We examine this coupling through a local projection diagnostic on seven Panda URDF nodes. Stack their image coordinates into $\mathbf{u} \in \mathbb{R}^{14}$ and the six joint and six pose increments into $\delta\boldsymbol{\theta}$. Using the per-parameter update ranges $\mathbf{b}$ to define $B = \mathrm{diag}(\mathbf{b})$ and $\delta\boldsymbol{\theta} = B\boldsymbol{\eta}$, the scaled Jacobian is

$$J = \left.\frac{\partial \mathbf{u}}{\partial \boldsymbol{\eta}}\right|_0 = \left.\frac{\partial \mathbf{u}}{\partial \delta\boldsymbol{\theta}}\right|_0 B = [J_\Phi\ J_T]. \tag{2}$$

A mixed near-null direction satisfies

$$J_\Phi \mathbf{v}_\Phi + J_T \mathbf{v}_T \approx 0, \qquad \begin{bmatrix}\mathbf{v}_\Phi \\ \mathbf{v}_T\end{bmatrix} \neq 0. \tag{3}$$

Such a direction allows articulation and pose to compensate for one another without substantially changing the projection. As shown in Figure 2, across randomly selected 100 samples, the median condition number is 3.4×10^8 for the combined Jacobian, compared with 1.2×10^2 and 7.8×10^1 for the articulation and pose blocks. Holding one block fixed removes access to this mixed direction within a solve and reduces the number of free variables to n or six. This motivates alternating updates that determine the shape at a given placement, then align the corrected shape as a rigid whole. Appendix A.6 gives the diagnostic protocol.



(a) Compared settings **(b)** Conditioning **(c)** Camera / side views

Figure 2: **Comparison between joint and decoupled strategies.** **(a)** Both strategies start from the same state and use the same number of optimization steps. **(b)** Median condition numbers of trust-region-scaled projection Jacobians over randomly selected 100 samples (log scale). **(c)** Camera overlays (top) and side views (bottom) for one frame. Orange denotes joint updates, teal decoupled updates, and dark gray the ground-truth boundary. Table 3 reports the dataset-level comparison.

Method overview. Building on this decoupling, SGAI constructs two conditional objectives from the robot model and image-derived evidence (Figure 3). We use the perception capabilities of visual foundation models to extract auxiliary geometric and foreground cues from the RGB image (Lin et al., 2025; Mei et al., 2026), represented by camera-frame points $\mathcal{P} = \{\mathbf{y}_a \in \mathbb{R}^3\}$ and foreground support M. These predictions remain fixed throughout state optimization.

Three constraints organize the evidence: point-to-link consistency $\mathcal{L}_{\text{link}}$, silhouette consistency comprising overlap $\mathcal{L}_{\text{sil}}$ and boundary coverage $\mathcal{L}_{\text{bd}}$, and projection anchor consistency $\mathcal{L}_{\text{anc}}$. With a local articulation regularizer $\mathcal{L}_{\text{prior}}$ and nonnegative weights λ, the objectives are

$$\mathcal{L}_{\text{art}} = \lambda_{\text{link}}\mathcal{L}_{\text{link}} + \lambda_{\text{sil}}\mathcal{L}_{\text{sil}} + \lambda_{\text{bd}}\mathcal{L}_{\text{bd}} + \lambda_{\text{prior}}\mathcal{L}_{\text{prior}}, \tag{4}$$

$$\mathcal{L}_{\text{pose}} = \lambda_{\text{bd}}^{T}\mathcal{L}_{\text{bd}} + \lambda_{\text{anc}}\mathcal{L}_{\text{anc}}. \tag{5}$$

The two blocks use different evidence for their respective corrections; these objectives are therefore state-conditioned subproblems rather than coordinate restrictions of one unchanged joint loss. At cycle k, we seek

$$\widetilde{\Phi}^{(k+1)} \approx \underset{\Phi\in\mathcal{Q}^{(k)}}{\arg\min}\ \mathcal{L}_{\text{art}}\left(\Phi \mid T_R^{C,(k)}, \Phi^{(k)}\right), \quad \widetilde{T}_R^{C,(k+1)} \approx \underset{T_R^C\in\mathcal{T}^{(k)}}{\arg\min}\ \mathcal{L}_{\text{pose}}\left(T_R^C \mid \Phi^{(k+1)}, T_R^{C,(k)}\right), \tag{6}$$

where $\mathcal{Q}^{(k)} \subseteq \mathcal{Q}$ and $\mathcal{T}^{(k)}$ are bounded joint and pose neighborhoods. Tildes denote numerical proposals; untilded iterates are the states retained after candidate checks. Fixed evidence and model parameters are omitted from the notation.

An articulation correction supplies the shape used for pose alignment. The resulting pose changes which link surfaces explain the observed points, so we re-render the accepted state and refresh associations before the next cycle. After L cycles, the output is $(\Phi^{(L)}, T_R^{C,(L)})$. The following sections define the constraints and the bounded local solver.

3.2 Consistency Constraints

Point-to-link consistency. The kinematic model makes geometric evidence specific to an articulated part. Let $F_\ell(\Phi)$ map link ℓ into the robot frame and $\bar{\mathbf{x}}_{\ell a}^{\{\ell\}}$ be a sampled point on its surface, expressed in homogeneous coordinates. Its camera-frame position is

$$\bar{\mathbf{x}}_{\ell a}^{\{C\}}(\Phi, T_R^C) = T_R^C F_\ell(\Phi)\bar{\mathbf{x}}_{\ell a}^{\{\ell\}}. \tag{7}$$

A joint update moves its downstream links through F_ℓ, whereas a pose update moves the entire articulated shape. We preserve link identity in the sampled surface sets $\mathcal{X}_\ell^C(\Phi, T_R^C)$ so that articulation is guided by this kinematic structure.

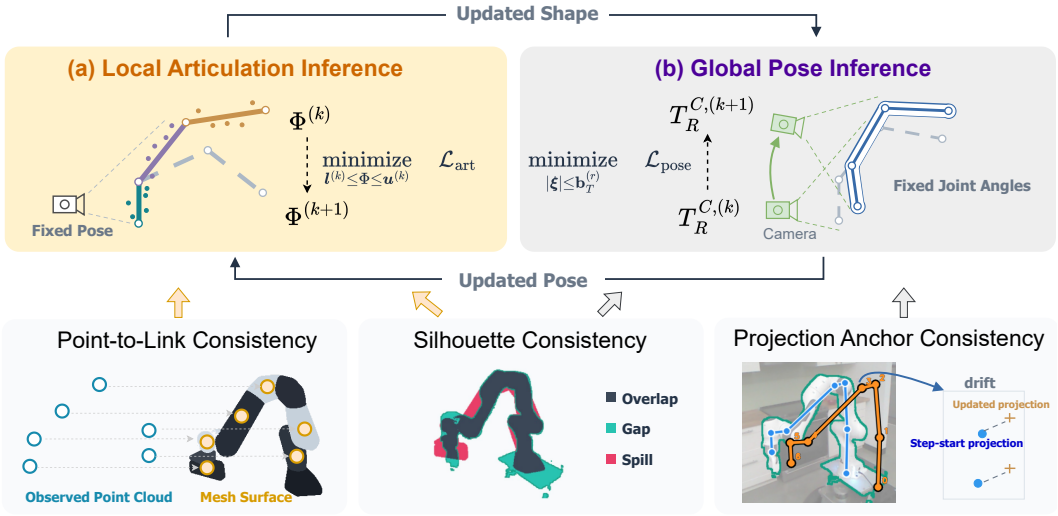


Figure 3: **Solving robot state refinement by structure-guided alternating inference.** SGAI seeks a state consistent with visual evidence and the robot model through two bounded optimization subproblems: articulation at fixed pose, followed by global pose at fixed articulation. Structure guides the evidence used in each objective, while refreshed geometry connects successive solutions and prevents simultaneous articulation–pose compensation within an update.

At the start of cycle k, each accepted observed point is assigned to the link of its nearest model sample. Distance and nearest-neighbor separation tests filter unreliable assignments. With pose fixed at $T_R^{C,(k)}$, the assigned link $\ell_a^{(k)}$ remains fixed within the articulation block, but the nearest sample on that link can change:

$$d_a(\Phi) = \min_{\mathbf{x} \in \mathcal{X}^C_{\ell_a^{(k)}}(\Phi, T_R^{C,(k)})} \|\mathbf{y}_a - \mathbf{x}\|_2. \tag{8}$$

Let $\mathcal{I}_\ell(\Phi)$ retain the smallest fraction η_{trim} of residuals assigned to link ℓ. The robust point-to-link loss is

$$\mathcal{L}_{\text{link}} = \frac{\sum_\ell \sum_{a \in \mathcal{I}_\ell(\Phi)} \rho(d_a(\Phi)/\sigma_x)}{\max(1, \sum_\ell |\mathcal{I}_\ell(\Phi)|)}, \qquad \rho(z) = \begin{cases} \frac{1}{2}z^2, & 0 \le z < 1, \\ z - \frac{1}{2}, & z \ge 1. \end{cases} \tag{9}$$

Here σ_x normalizes distance. Trimming and the Huber penalty reduce the influence of inconsistent depth samples, while link-restricted matching directs each residual toward the relevant articulated surface.

Silhouette consistency. Point-cloud agreement alone does not describe the visible extent of the robot. We therefore compare a differentiably rendered soft mask $A(\Phi, T_R^C)$ with the fixed foreground support M (Ravi et al., 2020). For pixel domain Ω and stabilizer ε, an overlap term measures soft intersection-over-union:

$$\mathcal{L}_{\text{sil}} = 1 - \frac{\sum_{u \in \Omega} A_u M_u + \varepsilon}{\sum_{u \in \Omega}(A_u + M_u - A_u M_u) + \varepsilon}. \tag{10}$$

Overlap is complemented by the spatial extent of the disagreement. Let $\delta_{\text{out}}(u)$ be the normalized distance to target foreground and $\delta_{\text{in}}(u)$ the normalized distance to target background. The boundary-and-coverage loss is

$$\mathcal{L}_{\text{bd}} = \frac{\sum_u A_u(1 - M_u)\delta_{\text{out}}(u)}{\max(\sum_u A_u, \varepsilon)} + \frac{\sum_u (1 - A_u) M_u \delta_{\text{in}}(u)}{\max(\sum_u M_u, \varepsilon)}. \tag{11}$$

Together, overlap and boundary coverage keep articulation corrections consistent with the visible robot. The pose objective uses boundary coverage to align the corrected shape as a rigid whole (Figure 3). It omits direct depth fitting, limiting the influence of depth bias on global placement.

Projection anchor consistency. Silhouette alignment can admit multiple rigid corrections when boundaries are incomplete or uncertain. We regularize this freedom with a projection anchor tied to the start of each pose block. Let m be the number of kinematic keypoints and $\mathbf{p}_i^{\{R\}}(\Phi)$ the ith keypoint from forward kinematics. Its projection is

$$P^i_{\text{proj}}(\Phi, T_R^C) = \pi_K\left(T_R^C \bar{\mathbf{p}}_i^{\{R\}}(\Phi)\right), \qquad i = 1, \dots, m, \tag{12}$$

where π_K is perspective projection. After articulation has been updated, we define the reference using the new shape and the current pose:

$$P^{i,(k)}_{\text{ref}} = P^i_{\text{proj}}\left(\Phi^{(k+1)}, T_R^{C,(k)}\right). \tag{13}$$

The reference remains fixed during that pose solve. Projection anchor consistency is then

$$\mathcal{L}_{\text{anc}}(T_R^C) = \frac{1}{m\sigma_p^2} \sum_{i=1}^{m} \left\| P^i_{\text{proj}}(\Phi^{(k+1)}, T_R^C) - P^{i,(k)}_{\text{ref}} \right\|_2^2. \tag{14}$$

The scale σ_p normalizes displacement in pixel units. The anchor is computed from the current state, so it requires neither another keypoint detector nor ground-truth projections. Resetting it after the articulation update lets the new shape guide pose inference without treating the articulation-induced projection change as pose drift.

3.3 Solving the Conditional Subproblems

We approximate the minimizers in Equation 6 using a finite number of gradient-based updates. Articulation changes are regularized by $\mathcal{L}_{\text{prior}} = \frac{1}{n}\sum_{j=1}^{n}\left((\Phi_j - \Phi_j^{(k)})/\sigma_{\Phi,j}\right)^2$, where $\sigma_{\Phi,j}$ sets the regularization scale for joint j. The reference $\Phi^{(k)}$ is reset at the start of each articulation block, while a separate bound around the initialization $\Phi^{(0)}$ limits cumulative displacement across rounds.

Bounded updates. For articulation, $\mathcal{Q}^{(k)} = [\mathbf{l}^{(k)}, \mathbf{u}^{(k)}]$ intersects URDF limits, the allowed neighborhood of the initialization, and joint-specific ranges derived from rendered-depth sensitivity. Coarse-to-fine optimization produces a proposal, which is passed to the retention and candidate checks. For pose, $\mathcal{T}^{(k)}$ is parameterized by a bounded left-multiplicative correction:

$$T_R^C(\boldsymbol{\xi}) = \exp(\widehat{\boldsymbol{\xi}}) T_R^{C,(k)}, \qquad \boldsymbol{\xi} = \mathbf{b}_T^{(k)} \odot \tanh \mathbf{z}. \tag{15}$$

Here $\mathbf{z} \in \mathbb{R}^6$, $\mathbf{b}_T^{(k)}$ contains per-component twist bounds, and the hat operator maps a twist to $\mathfrak{se}(3)$.

Candidate selection. The pose proposal is accepted only if boundary-and-coverage loss improves sufficiently. Subsequent articulation proposals are checked at damped step sizes using both the articulation objective and a silhouette score. The current state remains a candidate, and later cycles use smaller correction ranges. These rules limit weakly supported changes within the fixed cycle budget; ground-truth error is never used for selection. Appendix A gives the numerical settings and acceptance rules.

4 Experiments

We evaluate 3D localization across initializers and robot models, then examine surface geometry through rendered-depth comparisons. Ablations test the consistency constraints and update strategy, the accuracy–runtime trade-off across rounds, initialization sensitivity, and returned state components.

4.1 Evaluation Setup

Datasets. We use the nine DREAM settings (Lee et al., 2020) evaluated by RoboPEPP: six for Franka Emika Panda, two for KUKA iiwa, and one for Baxter. Synthetic images cover photorealistic (Photo) and domain-randomized (DR) rendering. Panda additionally provides Azure Kinect (AK), Xbox Kinect (XK), RealSense (RS), and varying-viewpoint Orbital (ORB) real-image sequences. The nine settings cover changes in appearance, camera viewpoint, and robot geometry. The rendered-depth analysis uses 61,613 Panda images.

Table 1: **3D robot localization on DREAM: ADD-AUC (%, ↑).** SGAI uses fixed hyperparameters across both baselines and all three robots, with no additional training or baseline- or robot-specific retuning. Bold marks the higher value within each pair; Δ is refined minus initial ADD-AUC.

Method	Known joint angles	Known bounding box	Panda						KUKA		Baxter
			Synthetic		Real				Synthetic		Synthetic
			Photo	DR	AK	XK	RS	ORB	Photo	DR	DR
Published results: robot-specific learned estimators											
DREAM-F	Yes	No	79.5	81.3	68.9	24.4	76.1	61.9	–	–	–
DREAM-Q	Yes	No	74.3	77.8	52.4	37.5	78.0	57.1	–	–	75.5
DREAM-H	Yes	No	81.1	82.9	60.5	64.0	78.8	69.1	72.1	73.3	–
HPE	No	Yes	82.0	82.7	82.2	76.0	75.2	75.2	73.9	75.1	58.8
HPE	No	No	40.7	41.4	66.7	–	49.1	51.6	56.7	56.2	9.8
RoboPose	No	No	79.7	82.9	70.4	77.6	74.3	70.4	73.2	80.2	32.7
RoboTAG	No	No	84.3	82.5	83.1	75.7	78.3	77.5	76.6	75.0	58.8
RoboPEPP	No	No	84.1	83.0	75.3	78.5	80.5	77.5	76.1	76.2	34.4
Paired results: shared refinement (fixed hyperparameters across baselines and robots)											
RoboPose (reprod.)	No	No	79.74	**82.91**	70.35	77.79	74.25	**70.42**	73.41	**80.15**	32.58
+ Ours	No	No	**80.11**	82.72	**72.71**	**78.83**	**75.91**	70.13	**73.54**	80.13	**32.77**
Δ (pp)			+0.37	-0.19	+2.36	+1.04	+1.66	-0.29	+0.13	-0.02	+0.19
RoboPEPP (reprod.)	No	No	84.03	**82.31**	75.18	78.49	80.56	77.52	76.20	73.89	**34.44**
+ Ours	No	No	**84.31**	82.10	**76.25**	**81.29**	**80.60**	**77.62**	**76.49**	**74.04**	34.40
Δ (pp)			+0.28	-0.21	+1.07	+2.80	+0.04	+0.10	+0.29	+0.15	-0.04

Initializers and comparison protocol. We refine reproduced RoboPEPP (Goswami et al., 2025) and RoboPose (Labbé et al., 2021) estimates on the same evaluated images. Both supply joint angles and robot-to-camera pose through a common state interface. Published methods in Table 1 provide context under their stated input conditions.

Evaluation metrics. Following RoboPEPP, we measure robot localization using the average distance (ADD) between predicted and ground-truth three-dimensional robot keypoints in the camera frame. For image s, let $\mathbf{p}_{s,i}^{\{C\}}$ and $\hat{\mathbf{p}}_{s,i}^{\{C\}}$ denote the ground-truth and estimated positions of its m evaluated keypoints, including the robot base. Then

$$e_s = \frac{1}{m}\sum_{i=1}^{m}\left\|\hat{\mathbf{p}}_{s,i}^{\{C\}} - \mathbf{p}_{s,i}^{\{C\}}\right\|_2, \qquad \text{ADD-AUC} = \frac{100}{\tau_{\max}}\int_0^{\tau_{\max}} \frac{1}{N}\sum_{s=1}^{N}\mathbf{1}[e_s \le \tau]\,d\tau, \tag{16}$$

with $\tau_{\max} = 100\,\text{mm}$. We use the released evaluation convention, approximating the integral on a 0.01 mm threshold grid. Higher ADD-AUC and lower mean ADD indicate better localization. AUC changes are reported in percentage points. Since ADD depends on the composed articulated state, an ADD improvement need not imply that every joint angle becomes more accurate.

Implementation. We extract auxiliary cues using Depth Anything 3 (DA3) (Lin et al., 2025) for depth and camera-frame geometry, and RobotSeg (Mei et al., 2026) for foreground segmentation. Their predictions are computed once and reused throughout refinement. All runs use known camera intrinsics and robot-specific URDF geometry and joint limits. For articulation, we set $(\lambda_{\text{link}}, \lambda_{\text{sil}}, \lambda_{\text{bd}}, \lambda_{\text{prior}}) = (0.45, 0.70, 0.25, 0.10)$. For pose, we use $(\lambda_{\text{bd}}^{T}, \lambda_{\text{anc}}) = (1.0, 0.08)$, with a projection-anchor scale of $\sigma_p = 4$ pixels. Both subproblems are optimized with Adam. Articulation uses learning rates of 0.020 and 0.008 at the coarse and fine rendering scales, respectively; pose uses a learning rate of 0.08. We perform two alternating rounds: articulation uses 24 and 6 steps per scale in the first and second rounds, respectively, while pose uses 18 and 9 steps. In the second round, the sensitivity-based articulation ranges are multiplied by $1/4$ and the pose bounds by $1/2$ to restrict subsequent corrections. SGAI requires no additional training, and all refinement hyperparameters are fixed across both initializers and all three robot models, without baseline- or robot-specific retuning. We evaluate the final accepted state without ground-truth-based iteration selection. Appendix A provides the remaining settings and solver details.

Table 2: **Rendered-depth MAE before and after refining RoboPEPP** (mm, ↓). Depth is rendered from joint angles and robot-to-camera pose using the URDF and compared with ground-truth depth. Count and % report frames with lower MAE.

Dataset	Frames	MAE (mm)		Depth improved	
		Initial	Final	Count	%
Panda AK	6,394	35.21	33.82	5,191	81.19
Panda XK	4,966	32.31	28.80	4,736	95.37
Panda RS	5,944	29.44	29.07	3,596	60.50
Panda ORB	32,315	34.79	34.09	20,829	64.46
Panda DR	5,997	24.08	24.25	3,382	56.39
Panda Photo	5,997	22.27	21.61	4,176	69.63
All Panda	61,613	31.86	30.98	41,910	68.02

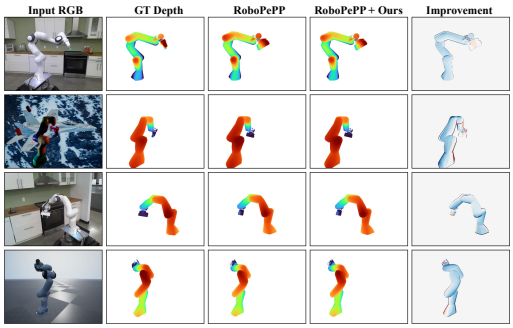


Figure 4: **Rendered-depth comparisons on DREAM.** Columns show RGB, ground-truth depth, initial and refined renderings, and the error change. Blue denotes reduced absolute depth error; red denotes increased error.

Table 3: **Constraints and update strategy on Panda XK.** ADD-AUC is in % (↑). Time is the extra cost per sample on an NVIDIA A6000 GPU, excluding the 0.31 s initialization.

Variant	Updates	ADD-AUC	Time (s)
Initial	–	78.49	0.00
w/o Point-to-Link	Decoupled	79.95	0.80
w/o Silhouette	Decoupled	80.12	1.05
w/o Projection Anchor	Decoupled	80.54	0.72
All constraints	Joint	80.21	2.45
Ours	**Decoupled**	81.29	2.21

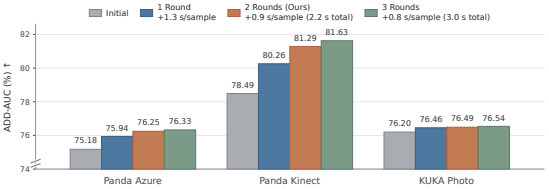


Figure 5: **Alternating-round ablation with RoboPEPP initialization.** Two rounds improve all three subsets over one round; a third adds 0.05–0.34 percentage points for another 0.8 s per sample. We use two rounds by default. Times exclude initialization; the vertical axis starts at 74%.

4.2 Robot State Refinement

Refinement across initializers and robots. The paired rows in Table 1 show higher ADD-AUC in six of nine settings for RoboPose and seven of nine for RoboPEPP. Both initializers improve on Panda AK and XK, with the largest gains occurring on AK for RoboPose and XK for RoboPEPP. SGAI obtains these gains through the estimated state and robot model, without accessing either initializer's learned features or changing its refinement hyperparameters between baselines and robots. The comparison supports a common refinement interface for the two tested estimators. Additional mean-ADD comparisons with published methods are provided in Appendix B.1.

Geometric effect of refinement. Keypoint localization does not fully describe the estimated robot surface. We therefore render depth from the initial and refined RoboPEPP states using the URDF and compare each rendering with ground-truth depth. This evaluates the geometry produced by the state estimates; the pretrained depth predictor remains unchanged. Table 2 shows lower mean absolute error on five of six Panda subsets, with the largest reduction on XK. Across all 61,613 frames, mean error decreases from 31.86 to 30.98 mm and 68.02% of frames improve. Figure 4 likewise shows both reduced-error and increased-error regions within individual images. Refinement can thus improve aggregate surface agreement without improving every frame or every rendered region. Appendix B.3 provides the depth metric definition.

4.3 Ablation Studies

All ablations use RoboPEPP estimates as initialization. We first assess the consistency constraints and update strategy on Panda XK, then examine the number of alternating rounds on three subsets, and finally evaluate sensitivity to initialization perturbations.

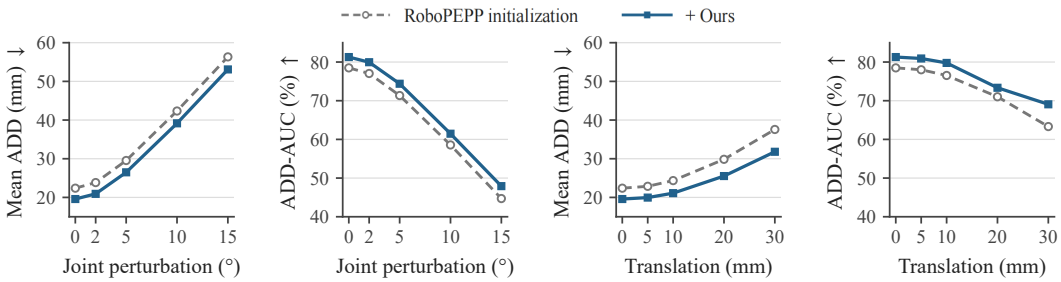


Figure 6: **Initialization-perturbation ablation on Panda XK.** From left to right: mean ADD and ADD-AUC under joint-angle perturbations, then the same metrics under translation perturbations. Gray curves show perturbed RoboPEPP states; blue curves show their outputs after complete refinement. Zero denotes the original initialization. Refinement improves both metrics at every tested magnitude, but does not eliminate the loss of accuracy as perturbations increase.

Effect of the three constraints. Table 3 removes one consistency constraint at a time while retaining decoupled updates and the same Panda XK initial states. Every reduced objective still improves over initialization, but none matches the full configuration. Removing point-to-link consistency causes the largest drop, showing that the 3D surface term contributes beyond the image-space constraints. Removing silhouette or projection anchor consistency also lowers accuracy, supporting their complementary roles in this setting. The reduced objectives run faster; retaining all three constraints trades additional computation for higher measured accuracy.

Joint versus decoupled updates. With all constraints included, decoupled updates achieve 81.29 ADD-AUC versus 80.21 for joint updates, while taking 2.21 rather than 2.45 s of extra time per sample. Thus, separating the state blocks improves final 3D localization in this comparison without increasing the measured runtime. This result complements the local projection diagnostic in Section 3.1: the diagnostic motivates the separation, and the ablation tests its effect under the complete refinement procedure on Panda XK.

Number of alternating rounds. Figure 5 compares zero to three rounds on Panda Azure (AK), Panda Kinect (XK), and KUKA Photo. The first round provides the largest individual gain on each subset. A second round improves all three, particularly XK, indicating that another pass through articulation and pose optimization remains useful after the initial correction. A third round gives the highest tested ADD-AUC, but it requires another 0.8 s per sample, increasing cumulative refinement time from 2.2 to 3.0 s. Therefore, we use two rounds as a practical choice: they retain most of the improvement observed after three rounds at a lower extra cost.

Initialization perturbations. We perturb RoboPEPP states on Panda XK while keeping the image, visual predictions, robot model, and refinement procedure fixed. Joint-angle perturbations are applied across the joint vector at magnitudes of 2°, 5°, 10°, and 15°. Translation is displaced in a random three-dimensional direction by 5, 10, 20, or 30 mm. The two perturbation families are tested separately, with zero perturbation as the control, and every perturbed state is passed through the complete refiner. Figure 6 shows lower mean ADD and higher ADD-AUC after refinement at every tested magnitude. The joint-perturbation curves retain a similar mean-ADD reduction as the initial error grows; the translation curves show a larger absolute reduction at larger displacements.

5 Conclusion and Limitations

SGAI refines robot states by solving bounded articulation and pose subproblems, guided by three consistency constraints and connected through refreshed geometry. A shared configuration improves ADD-AUC in most DREAM settings for both tested initializers, with small decreases elsewhere.

Limitations. SGAI remains sensitive to initialization quality and errors in the predicted depth and foreground cues. Its bounded local updates may not recover from large state errors, and the shared refinement configuration does not improve every evaluated setting. Iterative optimization also introduces additional computation per image.

AI USE STATEMENT

In this work, we have not used generative AI tools for any tasks with required disclosure, and all such tasks are not applicable to this work. We used generative AI tools only to polish the writing of parts of the manuscript. The authors reviewed all AI-assisted edits to ensure that they preserve the intended technical meaning and contain no unsupported claims. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 3 defines the optimization objectives, consistency constraints, and update procedure, while Section 4.1 describes the datasets, initializers, evaluation metrics, and key hyperparameters. Appendix A provides the numerical settings, solver details, candidate-selection rules, and inference procedure. Appendix B reports additional localization results and evaluation details. The pretrained visual components and benchmark sources are identified and cited in the main text. We will release the full source code.

REFERENCES

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

Ivan Bilić, Filip Marić, Ivan Marković, and Ivan Petrović. A distance-geometric method for recovering robot joint angles from an RGB image. *IFAC-PapersOnLine*, 56(2):1003–1008, 2023.

Ivan Bilić, Filip Marić, Fabio Bonsignorio, and Ivan Petrović. GISR: Geometric initialization and silhouette-based refinement for single-view robot pose and configuration estimation. *IEEE Robotics and Automation Letters*, 9(11):9883–9890, 2024. doi: 10.1109/LRA.2024.3466075.

Jeannette Bohg, Javier Romero, Alexander Herzog, and Stefan Schaal. Robot arm pose estimation through pixel-wise part classification. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3143–3150, 2014. doi: 10.1109/ICRA.2014.6907311.

Linghao Chen, Yuzhe Qin, Xiaowei Zhou, and Hao Su. EasyHeC: Accurate and automatic hand-eye calibration via differentiable rendering and space exploration. *IEEE Robotics and Automation Letters*, 2023.

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)*, pp. 6930–6939, 2025.

Xujun Han, Shaochen Wang, Xiucai Huang, and Zhen Kan. PoseFusion: Multi-scale keypoint correspondence for monocular camera-to-robot pose estimation in robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 795–801, 2024.

Yanghe Hao, Martin Huber, Christos Bergeles, and Tom Vercauteren. Streamlining stereo-differentiable rendering for marker-free real-time tracking of surgical robots. *International Journal of Computer Assisted Radiology and Surgery*, 2026. doi: 10.1007/s11548-026-03730-z.

Zhengdong Hong, Kangfu Zheng, and Linghao Chen. Fully automatic hand-eye calibration with pretrained image models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 816–823, 2024.

Martin Huber, Huanyu Tian, Christopher E. Mower, Lucas-Raphael Müller, Sébastien Ourselin, Christos Bergeles, and Tom Vercauteren. Hydra: Marker-free RGB-D hand-eye calibration. *arXiv preprint arXiv:2504.20584*, 2025.

Leonard Klüpfel, Lukas Burkhard, Anne Elisabeth Reichert, Maximilian Durner, and Rudolph Triebel. Seeing through uncertainty: Robot pose estimation based on imperfect prior kinematic knowledge. *IEEE Transactions on Robotics*, 41:4459–4478, 2025. doi: 10.1109/TRO.2025.3577030.

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)*, pp. 1654–1663, 2021.

Jens Lambrecht. Robust few-shot pose estimation of articulated robots using monocular cameras and deep-learning-based keypoint detection. In *International Conference on Robot Intelligence Technology and Applications (RiTA)*, pp. 136–141, 2019. doi: 10.1109/RITAPP.2019.8932886.

Jens Lambrecht and Linh Kästner. Towards the usage of synthetic data for marker-less pose estimation of articulated robots in RGB images. In *International Conference on Advanced Robotics (ICAR)*, pp. 240–247, 2019. doi: 10.1109/ICAR46387.2019.8981600.

Jens Lambrecht, Philipp Grosenick, and Marvin Meusel. Optimizing keypoint-based single-shot camera-to-robot pose estimation through shape segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13843–13849, 2021. doi: 10.1109/ICRA48506.2021.9561670.

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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6D pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 683–698, 2018.

Zekai Liang, Zih-Yun Chiu, Florian Richter, and Michael C. Yip. Differentiable rendering-based pose estimation for surgical robotic instruments. *arXiv preprint arXiv:2503.05953*, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Ruoshi Liu, Alper Canberk, Shuran Song, and Carl Vondrick. Differentiable robot rendering. In *Conference on Robot Learning (CoRL)*, 2024.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7708–7717, 2019.

Yifan Liu, Fangneng Zhan, Wanhua Li, Haowen Sun, Katerina Fragkiadaki, and Hanspeter Pfister. RoboTAG: End-to-end robot pose estimation via topological alignment graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35146–35155, 2026.

Jingpei Lu, Florian Richter, and Michael C. Yip. Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer. *IEEE Robotics and Automation Letters*, 7(2):4622–4629, 2022.

Jingpei Lu, Fei Liu, Cédric Girerd, and Michael C. Yip. Image-based pose estimation and shape reconstruction for robot manipulators and soft, continuum robots via differentiable rendering. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 560–566, 2023a. doi: 10.1109/ICRA48891.2023.10161066.

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)*, pp. 21296–21306, 2023b.

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 *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1914–1920, 2025.

Haiyang Mei, Qiming Huang, Hai Ci, and Mike Zheng Shou. RobotSeg: A model and dataset for segmenting robots in image and video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14346–14356, 2026.

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D deep learning with PyTorch3D. *arXiv preprint arXiv:2007.08501*, 2020.

Quanyuan Ruan, Jiabao Lei, Wenhao Yuan, Yanglin Zhang, Dekun Lu, Guiliang Liu, and Kui Jia. Prof. Robot: Differentiable robot rendering without static and self-collisions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22562–22572, 2025.

Alessandro Simoni, Stefano Pini, Guido Borghi, and Roberto Vezzani. Semi-perspective decoupled heatmaps for 3D robot pose estimation from depth maps. *IEEE Robotics and Automation Letters*, 7(4):11569–11576, 2022.

Alessandro Simoni, Francesco Marchetti, Guido Borghi, Federico Becattini, Lorenzo Seidenari, Roberto Vezzani, and Alberto Del Bimbo. 3D pose nowcasting: Forecast the future to improve the present. *Computer Vision and Image Understanding*, 251:104233, 2025. doi: 10.1016/j.cviu.2024.104233.

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)*, pp. 8917–8926, 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879, 2024.

Xiaopin Zhong, Wenxuan Zhu, Weixiang Liu, Jianye Yi, Chengxiang Liu, and Zongze Wu. G-SAM: A robust one-shot keypoint detection framework for PnP based robot pose estimation. *Journal of Intelligent & Robotic Systems*, 109(2):28, 2023. doi: 10.1007/s10846-023-01957-5.

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*, 10(11):11832–11839, 2025.

Appendix

A Implementation Details

This appendix gives the Panda reference implementation of the three main-text constraints and their alternating solver. The state interface and conditional objectives are independent of the initializer, and the refinement hyperparameters are shared across both initializers and Panda, KUKA, and Baxter without baseline- or robot-specific retuning. The numerical vectors below describe the six reported Panda joint angles; each robot supplies its own geometry, kinematic structure, state dimensions, and joint limits.

A.1 Observation and Geometry Preparation

The refiner reads cached camera-frame point coordinates, their pixel locations, camera intrinsics, and the image dimensions associated with the predicted geometry. It reconstructs M by marking all cached pixel locations as foreground *before* uniformly subsampling the points used in $\mathcal{L}_{\text{link}}$. Thus, M denotes the available, filtered foreground support; it is not assumed to equal an unfiltered segmentation output. Geometry and segmentation are supplied by pretrained visual components (Lin et al., 2025; Mei et al., 2026). Point coordinates, link meshes, and translations are interpreted in metres by the refiner. No depth scale or offset is estimated within the refinement loop.

The reference mesh model uses Panda link0–link7 and the hand, with at most approximately 800 faces per link. It draws 3,072 fixed surface samples using triangle-area weighting and barycentric sampling; forward kinematics moves these same samples during optimization. Up to 1,536 observed points are retained. The two mask-rendering resolutions are 189×252 and 378×504 (height $\times$ width), with camera intrinsics scaled accordingly. Soft silhouette rendering uses PyTorch3D (Ravi et al., 2020), eight faces per pixel, and softness settings 5×10^{-5} and 10^{-5} at the two scales. The model-depth query used for sensitivity follows the same camera convention.

A.2 Constraint Implementation

The three consistency losses and both conditional objectives are defined in the main text. This section records the sampling and normalization details used to evaluate them.

Point-to-link assignment. At the start of an articulation update, find the two nearest surface samples to each observed point across the entire articulated model. Denote their Euclidean distances by $d_{a,1}$ and $d_{a,2}$, with $d_{a,1} \le d_{a,2}$. The point is assigned the link label of the nearest sample only if

$$d_{a,1} \le \tau_{\text{dist}}, \qquad d_{a,2} - d_{a,1} \ge \tau_{\text{gap}}. \tag{17}$$

The two samples can belong to the same link. The separation test therefore concerns sampled-surface proximity, not a margin between two different link classes. Unassigned points do not contribute to the link loss.

Robust link aggregation. Equation 9 uses links with at least six assigned observed points and eight surface samples. Within each eligible link, the smallest $\lceil \eta_{\text{trim}} N_\ell \rceil$ residuals are retained and their indices are recomputed as the state changes. Retained points receive equal weight; links are not averaged equally. An empty retained set contributes zero link loss.

Silhouette distances and scales. For Equation 11, both target distance transforms are clipped at c_s and divided by c_s. Given source dimensions $H \times W$ and render dimensions $H_s \times W_s$, the reference implementation uses $c_s = \max(1, 25(H_s/H + W_s/W)/2)$ pixels and nearest-neighbor resizing for M. Articulation uses both render scales, while pose optimization and candidate acceptance use the finest scale.

Local references. The articulation prior defined in Section 3.3 is reset at the start of an articulation block and held fixed across its rendering scales. The separate cumulative bounds retain the original initialization as their reference. The anchor in Equation 14 is reset after articulation and held fixed throughout the pose block. The implementation computes it through a stabilized RMS; squaring introduces only a state-independent numerical constant.

A.3 Sensitivity and Bounded State Updates

Rendered-depth sensitivity. For joint j, perturb the current joint vector by $\pm h\mathbf{e}_j$ while holding pose fixed. At the finest render scale, the sensitivity is

$$s_j^{(k)} = \left[\frac{1}{|\mathcal{V}_j|} \sum_{u \in \mathcal{V}_j} \left(\frac{D_u(\Phi^{(k)} + h\mathbf{e}_j, T_R^{C,(k)}) - D_u(\Phi^{(k)} - h\mathbf{e}_j, T_R^{C,(k)})}{2h} \right)^2 \right]^{1/2}. \tag{18}$$

The set $\mathcal{V}_j$ includes probe pixels whose rendered depths exceed $0.05\,\text{m}$ for both perturbations. If fewer than eight valid probes remain, set $s_j^{(k)} = 0$. The shared observation loader samples up to 384

pixels using nonnegative cached confidence, or uniform confidence when unavailable. It reserves 30% in a separate partition; sensitivity uses the remaining partition. The reserved partition is not used for acceptance or held-out validation in this implementation, and observed depth values do not enter the sensitivity derivative.

Normalize $s_j^{(k)}$ by the 75th percentile $s_{\mathrm{ref}}^{(k)}$ of the positive sensitivities, lower-bounded by 10^{-8}. If none are positive, use $s_{\mathrm{ref}}^{(k)} = 1$. The joint range is

$$\begin{aligned} o_j^{(k)} &= \mathrm{clip}\Big(s_j^{(k)}/s_{\mathrm{ref}}^{(k)}, 0, 1\Big)\,, \\ b_j^{(k)} &= \gamma_\Phi^{(k)}\left[b_{\min} + (b_{\mathrm{base},j} - b_{\min})\sqrt{o_j^{(k)}}\right]. \end{aligned} \tag{19}$$

Here $\gamma_\Phi^{(k)}$ contracts later updates. These values measure per-joint depth response conditional on the current pose; they are not an observability decomposition of the joint-and-pose system.

Articulation bounds and solver. For URDF bounds $\Phi_j^{\min}, \Phi_j^{\max}$, define

$$\begin{aligned} l_j^{(k)} &= \max\Big(\Phi_j^{\min}, \Phi_j^{(k)} - b_j^{(k)}, \Phi_j^{(0)} - b_{\mathrm{base},j}\Big)\,, \\ u_j^{(k)} &= \min\Big(\Phi_j^{\max}, \Phi_j^{(k)} + b_j^{(k)}, \Phi_j^{(0)} + b_{\mathrm{base},j}\Big)\,. \end{aligned} \tag{20}$$

At initialization, the cumulative term is redundant because $b_j^{(0)} \le b_{\mathrm{base},j}$. We parameterize the bounded joint vector by

$$\Phi_j = \frac{l_j^{(k)} + u_j^{(k)}}{2} + \frac{u_j^{(k)} - l_j^{(k)}}{2}\tanh z_j. \tag{21}$$

Adam optimizes the unconstrained variables, with gradient norm clipped to 10. At each scale, the current state initializes the best-state record, and only finite, lower-loss states replace it. A non-finite loss stops that scale's inner optimization. Each new scale establishes its own best-state record, so this rule does not imply monotonicity of a single objective across scales.

The Panda solver internally augments the six reported angles with a seventh mesh angle ψ. Each articulation block starts at $\psi = 0$ and allows $|\psi| \le 0.05°$, subject to the URDF limit, with an additional prior $(\psi/30°)^2$. Geometry inside that solver is evaluated on (Φ, ψ). The block returns only the first six angles; candidate scoring, subsequent pose rendering, and evaluation append a zero seventh angle. This small auxiliary degree of freedom is not part of the output state and is not strictly frozen during the inner articulation optimization.

Pose bounds and solver. We order the twist as $\boldsymbol{\xi} = (\boldsymbol{\omega}, \mathbf{v})$. Writing the step-start pose as $(R_0, \mathbf{t}_0)$, Equation 15 gives

$$R = \exp([\boldsymbol{\omega}]_\times)R_0, \qquad \mathbf{t} = \exp([\boldsymbol{\omega}]_\times)\mathbf{t}_0 + J(\boldsymbol{\omega})\mathbf{v}, \tag{22}$$

where J is the left Jacobian of $SO(3)$. Adam starts from a zero twist and retains the evaluated state with the lowest pose objective. Rotation bounds apply to the components of $\boldsymbol{\omega}$, and translation bounds apply to $\mathbf{v}$; they are not bounds on the rotation-angle norm or on $\mathbf{t} - \mathbf{t}_0$. The projection anchor is a soft objective term, not a hard acceptance threshold.

A.4 Candidate Selection

Articulation selection. The initial articulation block returns its coarse-to-fine proposal using the within-scale retention rule above. Subsequent conservative refinement additionally checks damped candidates. For proposal $\widetilde{\Phi}$ from the current state $\Phi^{(k)}$, define

$$\Phi_\alpha = \Phi^{(k)} + \alpha(\widetilde{\Phi} - \Phi^{(k)}), \quad \alpha \in \mathcal{A}, \qquad G = \mathcal{L}_{\mathrm{sil}} + \beta\mathcal{L}_{\mathrm{bd}}. \tag{23}$$

All candidates are scored at the finest scale with fixed pose, link assignments, and prior reference. For a scalar score f, define relative improvement by

$$\Delta_f(x'; x) = \frac{f(x) - f(x')}{\max(|f(x)|, \varepsilon_f)}. \tag{24}$$

The zero candidate is always valid. A nonzero candidate is valid only when

$$\Delta_{\mathcal{L}_{\mathrm{art}}}(\Phi_\alpha; \Phi^{(k)}) \ge \tau_{\mathrm{art}}, \qquad \Delta_G(\Phi_\alpha; \Phi^{(k)}) \ge \tau_{\mathrm{mask}}. \tag{25}$$

The valid candidate with the lowest $\mathcal{L}_{\mathrm{art}}$ is selected. This is a joint check of a mixed objective and a mask score, not a guarantee based on independent observation sources.

Pose selection. For the best evaluated proposal $\widetilde{T}_R^C$, keep the updated articulation fixed and accept the pose only if

$$\Delta_{\mathcal{L}_{\mathrm{bd}}}(\widetilde{T}_R^C; T_R^{C,(k)}) \geq \tau_T. \tag{26}$$

Otherwise retain $T_R^{C,(k)}$. RGB-edge and anchor diagnostics in the shared solver are not acceptance votes for this contour-based configuration. Direct point-cloud fitting, ray-depth residuals, and affine depth calibration are not used in the pose block. Ground-truth states and evaluation metrics are never used to select the deployed output.

A.5 Reference Settings and Inference Procedure

The reference run instantiates $L = 2$ with $\gamma_\Phi^{(0)} = 1$, $\gamma_\Phi^{(1)} = 1/4$, and pose-bound multipliers 1 and $1/2$. The initial articulation block uses 24 Adam steps per scale; the subsequent block uses six. Pose refinement uses 18 and nine steps, respectively, at the finest scale. The output is the final state after these local selection rules, without selecting among outer iterations using ground-truth error. Table 4 lists the remaining settings. The schedule is an implementation of the conditional loop, not a convergence-based stopping criterion.

Table 4: **Numerical settings of the reference refiner.** Joint-specific vectors are shown for the six reported Panda angles. Angles are converted to radians during computation; pose bounds apply to individual twist components. The two-round step budgets are specified in the accompanying text.

Parameter	Value
Link assignment $(\tau_{\mathrm{dist}}, \tau_{\mathrm{gap}})$	$(55, 1.5)$ mm
Retained fraction η_{trim}; distance scale σ_x	0.8; 20 mm
$(\lambda_{\mathrm{link}}, \lambda_{\mathrm{sil}}, \lambda_{\mathrm{bd}}, \lambda_{\mathrm{prior}})$	$(0.45, 0.70, 0.25, 0.10)$
Joint prior scales $\boldsymbol{\sigma}_\Phi$	$(15, 25, 25, 25, 8, 8)^\circ$
Base joint ranges $\mathbf{b}_{\mathrm{base}}$	$(1.5, 3, 3, 3, 1.5, 1.5)^\circ$
Sensitivity perturbation h; range floor $b_{\min}$	0.25°; 0.15° before contraction
$(\lambda_{\mathrm{bd}}^T, \lambda_{\mathrm{anc}})$; anchor scale σ_p	$(1, 0.08)$; 4 pixels
Initial rotation-vector component bounds	$(0.5, 0.5, 0.5)^\circ$
Initial translation-twist component bounds	$(2, 2, 3)$ mm
Articulation candidate set $\mathcal{A}$; mask weight β	$\{0, 0.25, 0.5, 1\}$; 0.25
$(\tau_{\mathrm{art}}, \tau_{\mathrm{mask}}, \tau_T)$	$(0.002, 0.001, 0.003)$
Numerical stabilizers $(\varepsilon, \varepsilon_f)$	$(10^{-6}, 10^{-9})$
Articulation Adam rates, coarse / fine	0.020 / 0.008
Pose Adam rate	0.08

Inference procedure. The following pseudocode summarizes the reference control flow; numerical budgets follow the settings above.

Input: $I, K, \mathcal{S}, \mathcal{P}, M$ and $(\Phi^{(0)}, T_R^{C,(0)})$.
For $k = 0, \ldots, L-1$:
1. Fix $T_R^{C,(k)}$; reset the joint prior to $\Phi^{(k)}$.
2. Refresh point-to-link labels and rendered-depth sensitivities.
3. Set joint bounds and obtain a coarse-to-fine articulation proposal.
4. If $k = 0$, retain that proposal; otherwise apply the damped candidate checks.
5. Fix $\Phi^{(k+1)}$ and construct the step-start projection anchor.
6. Optimize the bounded pose increment and apply the contour acceptance test.

Return: $(\Phi^{(L)}, T_R^{C,(L)})$.

A.6 Projection-Jacobian Diagnostic

This diagnostic uses 100 frames and seven URDF nodes. Each frame contributes a 14-dimensional vector of image coordinates and 12 local parameters: six joint angles, three rotational pose coordinates, and three translational pose coordinates. The Jacobian columns are scaled by that sample's actual allowed update ranges as in Equation 2. This convention measures image sensitivity to a comparable fraction of the permitted update, rather than to one radian versus one metre. We summarize local numerical conditioning by the spectral condition number $\kappa(J) = \sigma_{\max}(J)/\sigma_{\min}(J)$; a larger value indicates greater disparity in sensitivity across parameter directions.

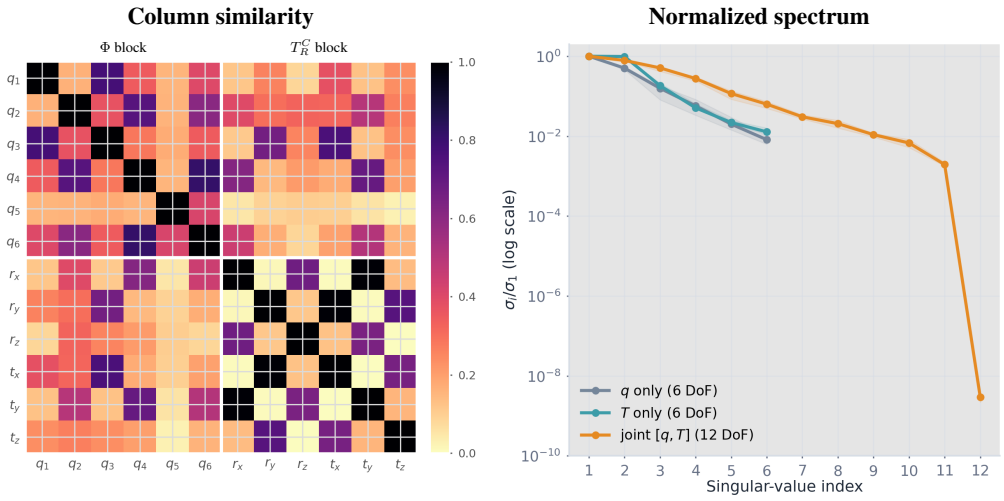


Figure 7: **Projection-Jacobian diagnostic over 100 frames.** Left: median absolute cosine similarity between columns; white lines separate articulation and pose, and diagonal entries equal one. Right: singular spectra normalized by each Jacobian's largest singular value, for the articulation-only, pose-only, and joint parameterizations. The joint spectrum has a much smaller minimum. Here q denotes Φ, and $r_x, r_y, r_z, t_x, t_y, t_z$ denote local pose coordinates. Figure 2b reports the median condition numbers.

The off-diagonal cross-block similarity reaches approximately 0.76 in the median matrix. Its absolute value captures both parallel and antiparallel image motions. The much smaller joint singular value diagnoses a linear combination of columns, rather than requiring any individual pair to be perfectly correlated. The normalized joint minimum is on the order of 10^{-9}; this number denotes σ_{12}/σ_1, not an unnormalized singular value. The conditional blocks have typical normalized minima near 10^{-2}. Figure 2b redraws only the three supplied condition-number medians; it does not reconstruct uncertainty intervals from the source image.

Fixing one block removes access to the mixed compensating direction within that local solve. It does not remove all possible weak directions or imply that the complete multiresidual problem has the same spectrum. In particular, the implemented refiner additionally uses link-associated geometry, masks, local priors, and bounded update rules. It is also distinct from the rendered-depth sensitivity used to allocate joint update ranges. The completed update-strategy ablation in Table 3 measures localization accuracy and runtime separately.

B Additional Experimental Results

B.1 Mean ADD on Real Panda Subsets

Table 5: **Mean ADD on real Panda subsets** (mm, ↓). Published baseline results are listed for context; + Ours refines reproduced RoboPEPP states. "–" denotes an unavailable result.

Method	AK	XK	RS	ORB	Avg.
DREAM-F	11,413	491,911	2,077	95,319	150,180
DREAM-Q	78,089	54,178	27	64,248	49,136
DREAM-H	57	7,382	24	25,685	8,287
RoboPose	34	22	26	30	28
HPE	46	–	61	52	53
RoboPEPP	29	22	23	**27**	26
+ Ours	**28**	**19**	**22**	**27**	**24**

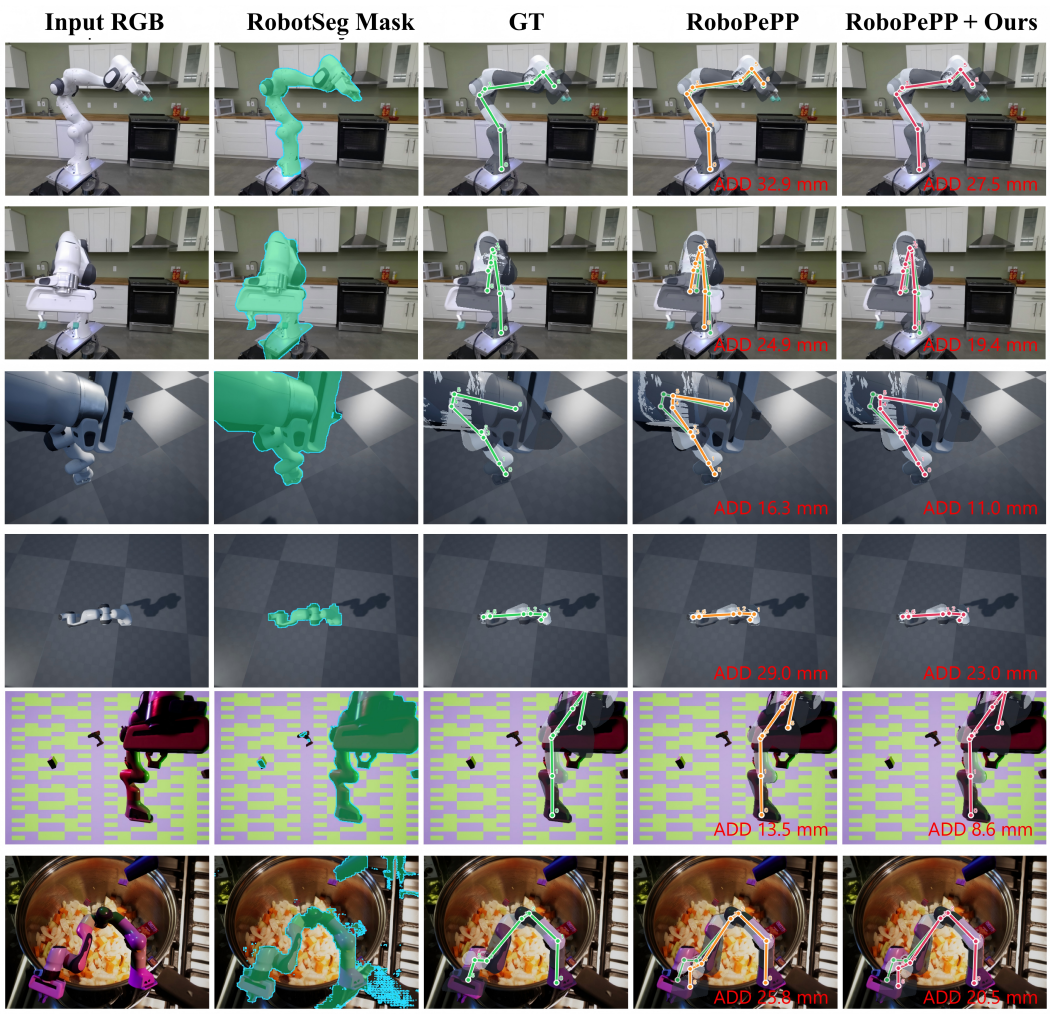


Figure 8: **Qualitative refinement results with RoboPEPP initialization.** Columns show the input RGB image, RobotSeg foreground mask, ground-truth state, RoboPEPP estimate, and SGAI-refined estimate. Green, orange, and pink skeletons denote ground truth, initialization, and refinement, respectively. The examples cover different viewpoints, apparent robot sizes, and backgrounds. Per-image ADD is reported in millimeters (↓).

Table 5 additionally reports mean ADD alongside published methods on the four real Panda subsets. This complements the main-text ADD-AUC comparison with absolute 3D keypoint localization error in millimeters. Whereas ADD-AUC summarizes the error distribution up to 100 mm, mean ADD remains sensitive to the magnitude of errors above that threshold.

At the reported precision, the refined estimates have lower mean ADD than the published RoboPEPP row on AK, XK, and RS, and the same value on ORB. These rows provide benchmark context: the published RoboPEPP values are not the reproduced initial states used by SGAI. Consequently, their differences should not be interpreted as paired refinement gains; Table 1 provides the paired comparison under a common evaluation protocol.

B.2 State-Component Ablation Details

This output-state ablation keeps the stored final articulation or final pose and substitutes the initialization for the other component. Because both final components were obtained by alternating inference, each was inferred in the context of an updated counterpart. The hybrid states show how these stored components combine geometrically; they do not measure what would happen if an op-

Table 6: **State-component recombination ablation with RoboPEPP initialization.** ADD-AUC (%, ↑) is evaluated after combining initial and final articulation and pose from the same completed refinement run. T abbreviates T_R^C. The middle columns substitute one initial component after optimization; they are not runs with a solver block disabled. The full refined state is not the best-scoring combination in every setting.

Dataset	$(\Phi^{(0)}, T^{(0)})$	$(\Phi^{(L)}, T^{(0)})$	$(\Phi^{(0)}, T^{(L)})$	$(\Phi^{(L)}, T^{(L)})$
Panda Photo	84.0288	84.0787	84.3043	84.3065
Panda DR	82.3053	82.3538	82.1046	82.1011
Panda AK	75.1755	75.1360	76.2976	76.2486
Panda XK	78.4922	78.6297	81.1726	81.2899
Panda RS	80.5576	80.6558	80.5185	80.6006
Panda ORB	77.5205	77.5430	77.6316	77.6194
KUKA Photo	76.1988	76.2576	76.4562	76.4888
KUKA DR	73.8877	73.9577	74.0669	74.0358
Baxter DR	34.4400	34.4720	34.3780	34.3960

timizer block were disabled during inference. Every setting is retained, including those where a hybrid scores above the complete output. Ground-truth metrics are not used to select the returned state.

B.3 Frame-Level and Depth Results

Table 2 reports sequence frame counts, rendered-depth MAE, and the number of frames whose error decreases after refinement. The percentages use each sequence's frame count as the denominator. Improvement frequency and mean error summarize different properties: DR has more improved than worsened frames, but a higher final mean error.

Rendered-depth error. For each image, the estimated joint angles and robot-to-camera pose are applied to the URDF to render depth D_s^{est}. The reported error is the mean absolute difference from ground-truth depth D_s^{gt} over evaluated pixels:

$$e_s^{\mathrm{depth}} = \frac{1}{|\Omega_s^{\mathrm{eval}}|} \sum_{u \in \Omega_s^{\mathrm{eval}}} \left| D_s^{\mathrm{est}}(u) - D_s^{\mathrm{gt}}(u) \right|. \tag{27}$$

Depth errors are expressed in millimeters. This metric evaluates rendered surfaces, whereas Equation 16 evaluates 3D keypoint distances. Refinement changes the state used for rendering, not the pretrained depth predictor. In Figure 4, blue and red mark decreases and increases in absolute depth error, respectively.

B.4 Qualitative State Refinement

Figure 8 complements the quantitative evaluation with six selected examples. The initial and refined projections can look similar, particularly in the distant top-down view in the fourth row, despite a reduction in 3D localization error. This illustrates why image-plane agreement alone is insufficient to assess the estimated state. ADD decreases in all six displayed cases, with reductions of 3.3–6.0 mm.

The foreground predictions also vary in quality. In the last row, the RobotSeg mask includes substantial background regions, yet refinement reduces ADD from 23.8 to 20.5 mm. This example illustrates a useful state correction with imperfect foreground evidence. The selected images show individual outcomes; dataset-wide performance, including settings with reduced accuracy, is reported in Table 1.