ROLA-GS: LINK-CANONICAL 3D GAUSSIAN SPLATTING FOR DIFFERENTIABLE ROBOT SELF-MODELING

Anonymous authors
Paper under double-blind review

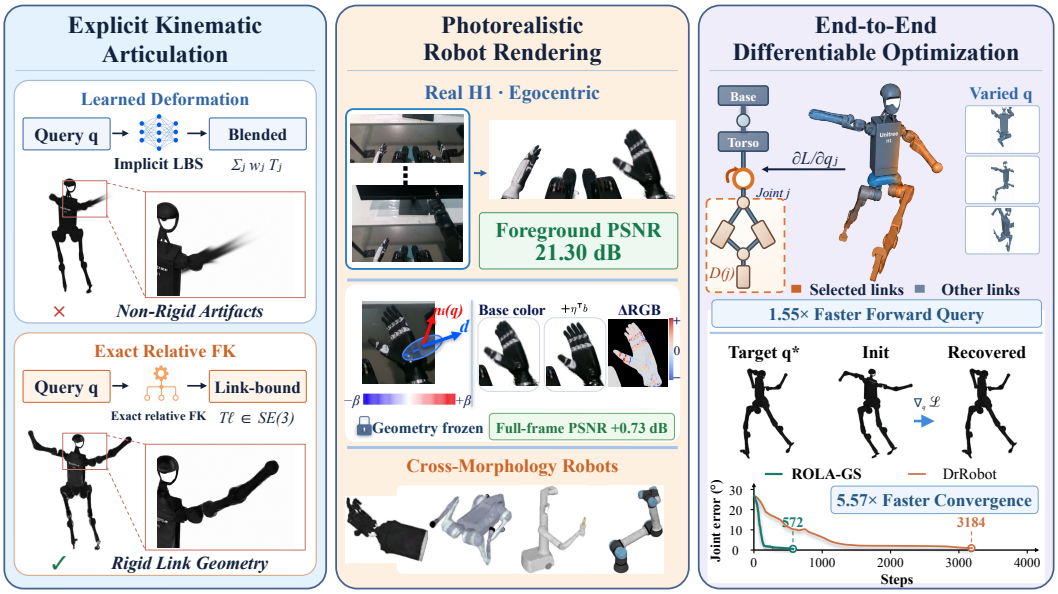


Figure 1: **ROLA-GS** represents articulated robots with link-bound Gaussians, using exact relative forward kinematics for rigid motion and bounded pose-dependent appearance correction for radiance variation. This differentiable representation preserves within-link geometry across joint configurations, supporting photorealistic robot rendering and image-based joint-state optimization.

ABSTRACT

Differentiable robot rendering synthesizes robot images from joint configurations and enables joint-state optimization through image gradients. Although robot kinematics specifies rigid link motion, learned deformation can couple this motion with appearance variation. We introduce ROLA-GS, a Gaussian visual self-model that separates rigid articulation from appearance modeling through fixed link assignments, exact relative forward kinematics, and pose-dependent appearance correction. Specifically, each Gaussian retains a fixed link assignment, allowing its center and covariance to move rigidly with the assigned link. A bounded appearance correction accounts for radiance variation without altering the transported geometry. This construction restricts each joint's spatial influence to its descendant links and provides differentiable rendering for visual inverse kinematics and selective control. Experiments demonstrate improved reconstruction across diverse robots and lower average errors in visual inverse kinematics and selective control. The results further show that rigid transport and appearance correction improve reconstruction, with rigid transport also improving forward-query efficiency.

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.

1 INTRODUCTION

Vision-based robot perception and control benefit from an accurate model of the robot's physical embodiment (Chen et al., 2022, Li et al., 2025). A robot visual self-model provides an interface that maps controllable kinematic states to camera observations (Liu et al., 2024). By connecting joint configurations with image space, such a representation enables sensor simulation (Li et al., 2024, Jiang et al., 2025a), visual state estimation (Labbé et al., 2021), and gradient-based control (Liu et al., 2024, Ruan et al., 2025). To serve these tasks, it must accurately render surface appearance, preserve rigid link articulation across joint states, and provide reliable gradients for optimization (Ruan et al., 2025, Li et al., 2026).

Existing representations struggle to satisfy these criteria simultaneously. Traditional robot descriptions, such as URDF or MJCF models, provide well-defined kinematic trees and exact rigid link transforms, yet their simplified CAD geometries and idealized surface properties fail to reproduce real-world camera captures. Recent differentiable robot renderers bridge this appearance gap by coupling forward kinematics with 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023, Liu et al., 2024, Ruan et al., 2025). However, to capture complex pose-dependent radiance and visual variations, frameworks such as DrRobot (Liu et al., 2024) often introduce pose-conditioned deformation networks. While this flexibility improves visual fidelity, it couples visual appearance with the underlying motion, making it difficult to preserve explicit, link-level kinematic states.

Beyond robot-specific renderers, dynamic radiance fields and articulated reconstruction methods model motion by optimizing continuous deformation fields (Pumarola et al., 2021, Yang et al., 2024), driving pre-defined kinematic templates (Lei et al., 2024), or estimating motion topologies from multi-view captures (Guo et al., 2025, Wang et al., 2026, Yu et al., 2026). These approaches are designed for under-constrained settings where part motion or joint topology must be inferred from visual observations alone. In contrast, calibrated robots already provide an analytical kinematic model that governs link transformations under any joint command. In this setting, relying on unconstrained deformation fields to model motion is not only redundant, but also introduces distinct failure modes: it entangles deterministic link articulation with visual discrepancies, leading to non-rigid distortions and boundary tearing under novel poses.

To resolve this tension, we present ROLA-GS, a structured 3DGS visual self-model designed for known-kinematics robotic systems. Rather than delegating articulation to a black-box deformation network, we factorize the visual state into discrete link ownership and continuous link-canonical Gaussian representations. Specifically, each body Gaussian is anchored to a nominal link frame, parameterizing its geometry and view-dependent appearance in canonical link coordinates. Global joint configurations transport Gaussian positions and covariances strictly through analytical forward kinematics (FK), while a bounded appearance formulation models illumination variations without altering the underlying rigid geometry. To account for visual evidence absent from nominal robot model, we introduce a decoupled residual state that remains invariant to joint articulations. This structural factorization yields descendant-only spatial state Jacobians: each joint changes the Gaussian geometry only on its descendant links. Differentiating the rendered image then supports joint-state optimization for visual IK and selective control.

- We introduce ROLA-GS, a Gaussian visual self-model for robots with known kinematics that separates rigid articulation from appearance modeling, preserving within-link geometry across joint configurations.
- Fixed link assignments and exact relative FK provide rigid Gaussian transport, while a bounded pose-dependent appearance correction models radiance variation. We derive descendant-only spatial state Jacobians for this representation to support visual IK and selective control.
- Experiments on real RGB and MuJoCo benchmarks show reconstruction gains, including a 0.55 dB improvement in mean PSNR over DrRobot across seven MuJoCo robots under 96-state training. Paired task comparisons demonstrate lower average visual IK and selective-control errors than DrRobot, while kinematic transport ablations show faster forward queries than learned deformation.

2 Related Work

Differentiable robot rendering methods. Methods for differentiable robot rendering combine a kinematic representation with a learnable visual representation. DrRobot couples Gaussian splatting with implicit linear blend skinning and pose-conditioned appearance deformation (Liu et al., 2024), while subsequent methods add geometry, collision, or motion components (Ruan et al., 2025, Li et al., 2026, Wang et al., 2025). RoboGSim, Self-Modeling Robots by Photographing, and GSWorld build Gaussian assets for simulation, self-modeling, or manipulation (Li et al., 2024, Hu et al., 2025, Jiang et al., 2025a).

Articulated object reconstruction methods. Dynamic radiance-field methods learn canonical-to-posed deformation from image observations (Pumarola et al., 2021, Yang et al., 2024). Articulated object methods use templates, skeletons, correspondences, or motion constraints (Lei et al., 2024, Hu et al., 2024, Guo et al., 2025, Liu et al., 2025, Lin et al., 2025, Wu et al., 2025, Lewis et al., 2025), including extensions to sparse multi-state input (Wang et al., 2026, Yu et al., 2026, Wu et al., 2026). Recent frameworks also cover sparse few-state observations and uncalibrated views (Lee et al., 2026b), bounding-box structural priors (Lee et al., 2026a), and free-moving captures (Dai et al., 2026).

Geometry-guided Gaussian reconstruction methods. Geometry-guided methods constrain Gaussian primitives with meshes, oriented surface elements, depth, normal, or opacity cues (Guedon & Lepetit, 2024, Huang et al., 2024, Turkulainen et al., 2025, Chen et al., 2024, Yu et al., 2024). Multi-view propagation guides Gaussian densification (Cheng et al., 2024) and surface reconstruction (Wang et al., 2024), while multi-view feature aggregation enables feed-forward generalizable Gaussian prediction (Yang et al., 2025). Other approaches use mesh-guided Gaussian representations (Ye et al., 2025) or surfel-based geometry fields (Jiang et al., 2025b) for surface reconstruction and rendering.

3 Method

3.1 Link-Bound Gaussian Representation

Given a robot kinematic model $\mathcal{K}$ with a discrete link set $\mathcal{L}$, the training observations consist of tuples $\{(I_n, q_n, M_n, \pi_n)\}_{n=1}^{N}$, where I_n denotes an RGB image, q_n is the joint configuration, M_n is the foreground robot mask, and π_n encapsulates the calibrated camera intrinsics and extrinsics. We initialize P reference Gaussians by uniformly sampling the nominal surface meshes of the robot at a chosen reference configuration q_0. Each Gaussian has a fixed link assignment $\ell_i \in \mathcal{L}$ corresponding to the sampled link surface (Figure 2).

At an arbitrary query configuration q, the spatial state of the robot is represented by:

$$\mathcal{G}(q) = \{\mu_i(q),\, \Sigma_i(q),\, \alpha_i,\, c_i,\, \eta_i,\, \ell_i\}_{i=1}^{P}\,, \tag{1}$$

where $\mu_i(q) \in \mathbb{R}^3$ and $\Sigma_i(q) \in \mathbb{S}^3_{++}$ denote the center and positive definite covariance matrix of Gaussian i at configuration q, $\alpha_i \in [0,1]$ is the opacity, c_i represents the learnable spherical harmonics (SH) coefficients, and $\eta_i \in \mathbb{R}^{B\times 3}$ parameterizes a pose-dependent appearance correction. The reference quantities μ_i^0 and Σ_i^0 are established in the robot-base frame at q_0 and remain rigidly tied to link ℓ_i.

The forward kinematics of the robot yields the relative rigid transformation $T_\ell^{q\leftarrow q_0} \in \mathrm{SE}(3)$ for link ℓ from q_0 to q:

$$T_\ell^{q\leftarrow q_0} = T_\ell(q)T_\ell(q_0)^{-1} = \begin{bmatrix} R_\ell^{q\leftarrow q_0} & t_\ell^{q\leftarrow q_0} \\ 0 & 1 \end{bmatrix}. \tag{2}$$

Conditioned on the fixed link assignment ℓ_i, the forward kinematics transforms each Gaussian to configuration q via:

$$\begin{aligned} \mu_i(q) &= R_{\ell_i}^{q\leftarrow q_0}\mu_i^0 + t_{\ell_i}^{q\leftarrow q_0}, \\ \Sigma_i(q) &= R_{\ell_i}^{q\leftarrow q_0}\Sigma_i^0\left(R_{\ell_i}^{q\leftarrow q_0}\right)^{\mathsf{T}}. \end{aligned} \tag{3}$$

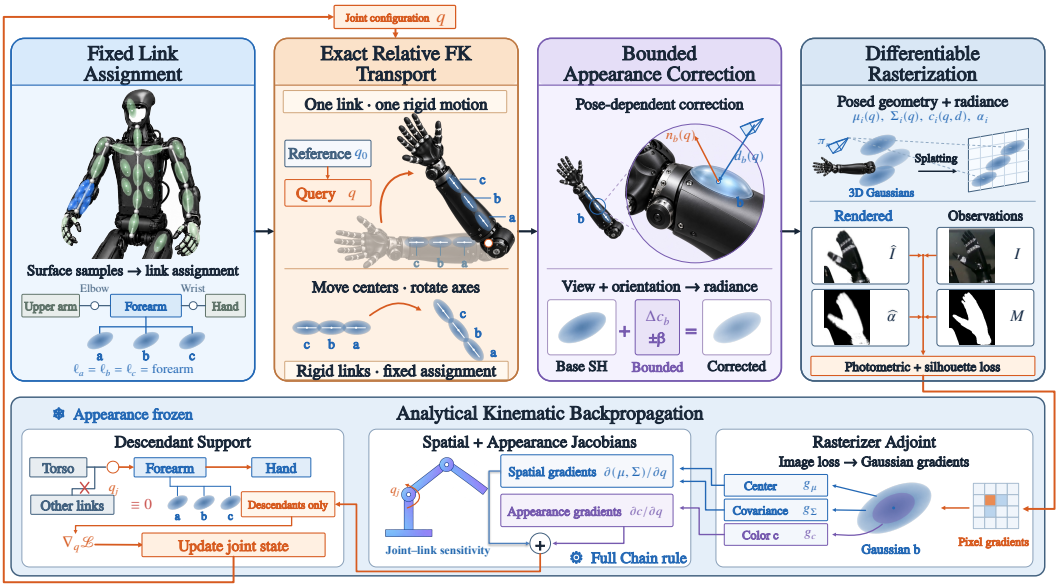


Figure 2: **Framework of ROLA-GS.** Gaussians initialized from link surfaces retain fixed link assignments and follow the query joint configuration through exact relative FK. Bounded appearance correction and differentiable rasterization produce RGB images and silhouettes. During joint-state optimization, photometric and silhouette losses backpropagate through this path to update the joint configuration with Gaussian parameters fixed.

where $R_\ell^{q\leftarrow q_0}$ and $t_\ell^{q\leftarrow q_0}$ are the rotation and translation components. Rigid transport preserves link-local spatial scales while rotating anisotropic covariance with the link frame.

3.2 Appearance Correction and Differentiable Rendering

For a calibrated camera with optical center o_{cam}, the normalized viewing direction $d_i(q)$ originating from the Gaussian center to the camera is given by:

$$d_i(q) = \frac{o_{\text{cam}} - \mu_i(q)}{\|o_{\text{cam}} - \mu_i(q)\|}. \tag{4}$$

Evaluating the learned SH coefficients c_i along $d_i(q)$ yields the base view-dependent radiance $c_i^{\text{SH}}(d_i) \in \mathbb{R}^3$.

While standard SH captures static view dependence, physical joint articulation alters mutual shadows and inter-reflections across links. To account for these variations without corrupting the underlying rigid geometry, we introduce a bounded, pose-dependent appearance correction inspired by classical angular reflectance models (Phong, 1975, Blinn, 1977):

$$z_i(q,d) = \eta_i^{\mathsf{T}} b_i(q,d), \qquad c_i(q,d) = \left[c_i^{\text{SH}}(d) + \beta \tanh(z_i(q,d))\right]_+, \tag{5}$$

where $b_i(q,d) \in \mathbb{R}^B$ encodes the rotated Gaussian normal and its angular response to d, $\eta_i \in \mathbb{R}^{B\times 3}$ contains appearance coefficients, and $z_i(q,d) \in \mathbb{R}^3$ is the response before the bounded mapping. The scalar $\beta > 0$ bounds the correction $\beta\tanh(z_i)$, and $[\cdot]_+ \equiv \max(\cdot, 0)$ ensures non-negative RGB values before splatting.

Given the posed state $\mathcal{G}(q)$, we rasterize the Gaussians using the calibrated camera parameters π_n via differentiable tile-based splatting (Kerbl et al., 2023), rendering an RGB image $\widehat{I}_n$ and an accumulated transmittance-derived silhouette mask $\widehat{\alpha}_n$. This end-to-end differentiable rendering path provides analytical gradients with respect to both the Gaussian parameters and the robot joint configuration q.

3.3 Training and Joint-State Optimization

We optimize the structured Gaussian representation using a multi-objective loss combining photometric fidelity, silhouette consistency, and a geometric anchor regularizer:

$$\mathcal{L}_{\text{total}} = E_{\text{pho}} + \lambda_{\text{sil}} E_{\text{sil}} + \lambda_{\text{anc}} E_{\text{anc}}, \tag{6}$$

where λ_{sil} and λ_{anc} are balancing hyperparameters. Inspired by foreground-guided optimization in articulated rendering (Hu et al., 2024), we formulate the photometric objective E_{pho} as a spatially weighted ℓ_1 loss over observed frames:

$$E_{\text{pho}} = \frac{1}{N|\Omega|} \sum_{n=1}^{N} \sum_{p \in \Omega} w_n(p) \left\| \widehat{I}_n(p) - I_n(p) \right\|_1, \tag{7}$$

where Ω denotes the image lattice and N is the number of training views. $w_n(p)$ selectively emphasizes the robot foreground region by modulating the loss with the ground-truth mask M_n.

The silhouette loss discourages spurious semi-transparent floaters by matching rendered alpha $\widehat{\alpha}_n$ to the robot mask M_n:

$$E_{\text{sil}} = \frac{1}{N|\Omega|} \sum_{n=1}^{N} \|\widehat{\alpha}_n - M_n\|_2^2 . \tag{8}$$

To prevent Gaussian centers from drifting excessively from nominal link surfaces during density control, we introduce a geometric anchor regularization term and regularize appearance correction:

$$E_{\text{anc}} = \frac{1}{P} \left\| \theta - \bar{\theta} \right\|_2^2 + \frac{\lambda_{\text{res}}}{P} \sum_{i=1}^{P} \|\eta_i\|_F^2 , \tag{9}$$

where θ encompasses the canonical centers and log-scales, and $\bar{\theta}$ denotes their reference initialization. This explicit displacement penalty is distinct from mesh-adapter formulations such as GeoSplatting (Ye et al., 2025).

Training first optimizes Gaussians at the reference configuration q_0, then across joint configurations using exact relative FK (Eq. 3). Adaptive density control (Kerbl et al., 2023) prunes and densifies Gaussians, with each new Gaussian inheriting its parent's fixed link assignment ℓ_i.

Representation learning fixes joint configurations q_n and optimizes Gaussian parameters. Visual IK and selective control instead fix these parameters and backpropagate image-loss gradients to update q. The same derivatives characterize joint-state sensitivity for auditing.

For a calibrated camera view n, let $\mu_{i,n}^{\text{I}}$ and $\Sigma_{i,n}^{\text{I}}$ denote the image-plane mean and covariance of Gaussian i. Applying the chain rule, the forward analytical Jacobians with respect to the joint configuration q are given by:

$$\begin{aligned} \frac{\partial \mu_{i,n}^{\text{I}}}{\partial q} &= \frac{\partial \mu_{i,n}^{\text{I}}}{\partial \mu_i(q)} \frac{\partial \mu_i(q)}{\partial q}, \\ \frac{\partial \Sigma_{i,n}^{\text{I}}}{\partial q} &= \frac{\partial \Sigma_{i,n}^{\text{I}}}{\partial J_{i,n}} \frac{\partial J_{i,n}}{\partial \mu_i(q)} \frac{\partial \mu_i(q)}{\partial q} + \frac{\partial \Sigma_{i,n}^{\text{I}}}{\partial \Sigma_i(q)} \frac{\partial \Sigma_i(q)}{\partial q}, \end{aligned} \tag{10}$$

where $J_{i,n}$ represents the affine projection Jacobian at the Gaussian center in camera coordinates. The pose-dependent radiance contributes an appearance derivative through the view direction and the bounded correction in Eq. 5. The full expression is given in Appendix B.5. The underlying 3D spatial derivatives in Eq. 10 are governed strictly by the robot's kinematic hierarchy. For a scalar joint coordinate q_j, let $z_j(q) \in \mathbb{R}^3$ and $o_j(q) \in \mathbb{R}^3$ denote the unit motion axis and origin of joint j in world coordinates. The spatial derivatives correspond to the rigid-body geometric Jacobian columns:

$$\frac{\partial \mu_i(q)}{\partial q_j} = v_{ij}(q), \qquad \frac{\partial \Sigma_i(q)}{\partial q_j} = [\omega_{ij}(q)]_\times \Sigma_i(q) - \Sigma_i(q) [\omega_{ij}(q)]_\times , \tag{11}$$

where $[\cdot]_\times \in \mathfrak{so}(3)$ represents the skew-symmetric cross-product matrix. The Lie bracket formulation in Eq. 11 strictly preserves the symmetry and positive-definiteness of $\Sigma_i(q)$ on the $\mathbb{S}_{++}^3$ manifold.

Crucially, the kinematic influence is topologically localized: if link ℓ_i is a topological descendant of joint j, the induced linear rate $v_{ij}(q)$ and angular rate $\omega_{ij}(q)$ are defined as follows:

$$v_{ij}(q) = \begin{cases} z_j(q) \times (\mu_i(q) - o_j(q)), & \text{revolute joint,} \\ z_j(q), & \text{prismatic joint,} \end{cases} \qquad \omega_{ij}(q) = \begin{cases} z_j(q), & \text{revolute joint,} \\ \mathbf{0}, & \text{prismatic joint.} \end{cases} \tag{12}$$

For links outside joint j's descendant subtree, $v_{ij}(q) = \omega_{ij}(q) = \mathbf{0}$, so their spatial Jacobians vanish. We audit this descendant-only property with single-joint angle sweeps and synthetic link-assignment perturbations.

4 Experiments

4.1 Experimental Setup

Protocol and metrics. The MuJoCo benchmark (Todorov et al., 2012) uses 64 canonical and 96 articulated states, rendered from 12 calibrated 256^2 views, with 500 held-out states. Paired methods share RGB images, cameras, and states. Training schedules and joint-conditioned baseline adaptations are specified in Appendix A.9. We compare DrRobot, GRADRobot (Li et al., 2026), Deformable 3DGS (Yang et al., 2024), 4D-GS (Wu et al., 2024), Part^{2}GS (Yu et al., 2026), and REArtGS++ (Wu et al., 2026). Metrics are PSNR, SSIM, LPIPS (Zhang et al., 2018), official squared center-Chamfer, hinge-angle MAE, target error and protected-link displacement (mm), and synchronized latency. The appendix specifies adapters and separate real-RGB, 10k-state, trajectory, and two-state protocols.

4.2 Reconstruction Quality

Table 1: **Robot reconstruction on real RGB and MuJoCo benchmarks.** PSNR is evaluated on the common foreground for real RGB and on the full image for MuJoCo.

Method	DREAM Panda			Humanoid H1
	Azure	ORB	RealSense	Egocentric
PSNR (dB) ↑				
DrRobot (Liu et al., 2024)	15.873	15.294	12.055	3.902
GRADRobot (Li et al., 2026)	16.679	17.739	18.189	4.849
Deformable 3DGS (Yang et al., 2024)	10.960	9.976	11.138	4.471
4D-GS (Wu et al., 2024)	10.077	10.520	10.868	5.237
Part^{2}GS (Yu et al., 2026)	9.947	13.440	12.731	4.693
REArtGS++ (Wu et al., 2026)	10.187	13.706	12.704	4.695
ROLA-GS	**20.227**	**20.065**	**23.652**	**21.297**

Method	MuJoCo Simulation							
	Panda	Google	ViperX	Shadow	H1	Go1	PND	Avg.
PSNR (dB) ↑								
DrRobot (Liu et al., 2024)	29.02	**36.98**	**31.50**	**32.69**	29.14	**30.62**	29.78	31.39
GRADRobot (Li et al., 2026)	26.91	34.98	28.62	30.78	27.41	29.03	27.96	29.39
Deformable 3DGS (Yang et al., 2024)	23.56	29.92	18.24	23.49	15.55	18.17	19.91	21.26
4D-GS (Wu et al., 2024)	21.97	29.90	14.63	14.95	14.70	17.20	19.92	19.04
Part^{2}GS (Yu et al., 2026)	23.55	29.95	18.36	25.27	15.68	18.74	19.82	21.62
REArtGS++ (Wu et al., 2026)	23.55	29.94	18.68	24.85	15.63	18.77	19.79	21.60
ROLA-GS	**32.40**	36.27	31.45	32.68	**29.53**	30.34	**30.90**	**31.94**
Chamfer Distance (10^{-3} m^2) ↓								
DrRobot (Liu et al., 2024)	7.205	1.143	0.112	0.140	0.684	0.352	7.824	2.494
GRADRobot (Li et al., 2026)	6.835	**0.961**	**0.092**	**0.085**	**0.420**	**0.204**	10.468	2.724
Deformable 3DGS (Yang et al., 2024)	251.344	16.996	30.597	0.651	32.839	3.739	54.066	55.747
4D-GS (Wu et al., 2024)	1.83×10^5	17.860	3.45×10^5	4.95×10^4	52.758	5.776	62.158	8.25×10^4
Part^{2}GS (Yu et al., 2026)	37.272	16.107	24.196	0.594	60.100	5.446	34.235	25.421
REArtGS++ (Wu et al., 2026)	36.928	14.377	35.273	0.715	58.529	5.996	35.593	26.773
ROLA-GS	**0.547**	2.147	0.169	0.244	1.470	0.526	**6.563**	**1.667**

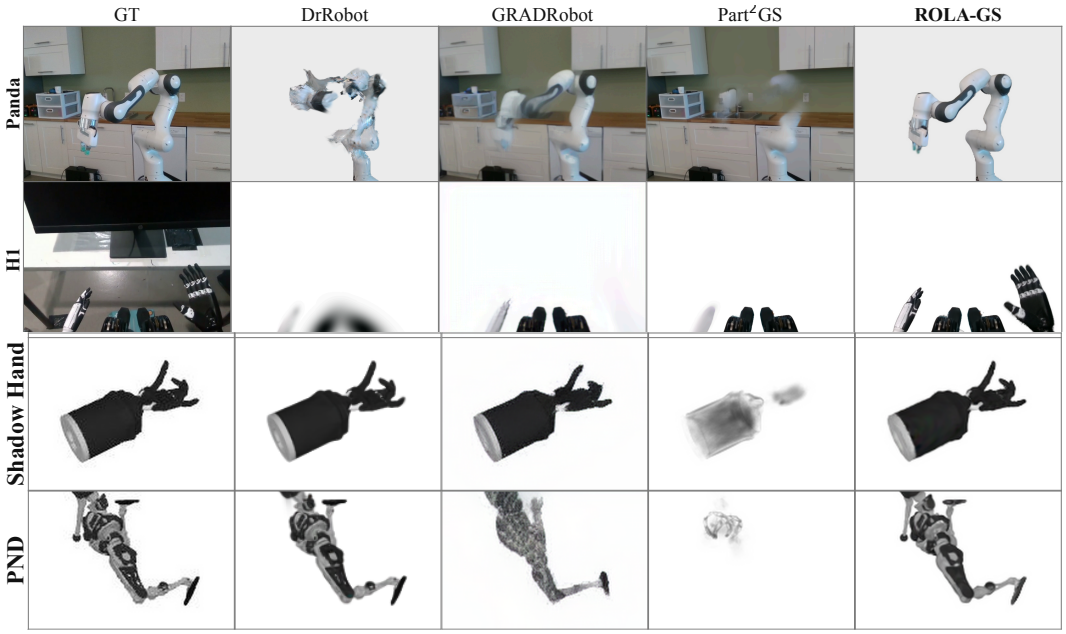


Figure 3: **Qualitative reconstruction on real RGB and MuJoCo benchmarks.** ROLA-GS preserves continuous Panda link surfaces and reconstructs both visible H1 hands, with simulated examples showing finger and arm contours.

We evaluate native 640×480 RGB: held-out frames from three DREAM Panda cameras (Lee et al., 2020) and 630 observed egocentric H1 frames. ROLA-GS exceeds the strongest Panda baselines by $3.55/2.33/5.46$ dB on Azure/ORB/RealSense (Table 1). On H1, common-foreground PSNR rises from 5.24 to 21.30 dB. In Figure 3, Panda links fragment in DrRobot. GRADRobot recovers their shape but blurs surface detail, while ROLA-GS retains continuous link surfaces. On H1, it recovers both visible hands and their separated fingers, which are incomplete in the comparison renders.

Under shared 96-state MuJoCo training, ROLA-GS leads PSNR on Panda, H1, and PND, while DrRobot leads on the other four robots. ROLA-GS raises the strongest baseline average from 31.39 to 31.94 dB, with its largest gains on Panda ($+3.38$ dB) and PND ($+1.12$ dB). In Figure 3, DrRobot preserves much of the simulated robots' overall shape, while finer differences appear in the finger contours and PND arm: GRADRobot has rougher boundaries and Part^{2}GS loses articulated detail. ROLA-GS retains distinct joint contours, with further Go1 and ViperX comparisons in Appendix Figure 6.

Geometry is evaluated by unfiltered bidirectional squared center-Chamfer (CD), with Avg. in Table 1 taken over the seven shown robots. ROLA-GS reduces squared center-Chamfer on Panda from GRADRobot's 6.835 to 0.547, and on PND from DrRobot's 7.824 to 6.563 (all $\times10^{-3}\,\mathrm{m}^2$). The seven-robot mean is 1.667, versus $2.724/2.494$ for GRADRobot/DrRobot. GRADRobot leads on the other five robots: center-Chamfer measures Gaussian-center proximity without visibility or opacity weighting, so its ranking need not follow rendered boundary quality. The advantage persists with 10k states (Appendix Table 14). Further protocol and resolution comparisons appear in Appendix A.

4.3 Articulated State Estimation and Control

To recover joint state from images, we freeze visual parameters and optimize only q from one held-out view over four morphologies and 600 paired cases, sharing targets, query cameras, initial states, losses, and 200-step budgets. Learning rates are method-specific (Appendix Table 4), with per-robot errors reported in Appendix Table 5. ROLA-GS achieves the lowest hinge-angle MAE at all three initialization distances (Table 2). At $d_0 = .100$, its 9.06° error compares with 11.27° for DrRobot, 11.61° for soft link blending, and 12.61° for learned deformation. The fixed–soft assignment gap grows from 0.06° to 2.55° as initialization moves farther from the target. Figure 4(a) localizes the

Table 2: **Visual IK and selective control.** Paired control values report target error and protected-link displacement.

Method	Visual IK: hinge-angle MAE (°) ↓			Selective control: target / protected (mm) ↓		
	$d_0 = .025$	$d_0 = .050$	$d_0 = .100$	5% range	10% range	20% range
w/ Learned Deformation	5.93	6.67	12.61	63.46 / 8.26	79.40 / 8.29	142.17 / 8.20
w/ Soft Link Blending	2.23	3.83	11.61	12.65 / 3.04	18.21 / 3.08	56.83 / 3.17
DrRobot	3.39	4.70	11.27	32.55 / 3.04	43.40 / 3.37	103.66 / 4.10
ROLA-GS	**2.17**	**3.14**	**9.06**	**6.28 / 1.88**	**10.23 / 1.98**	**45.01 / 2.02**

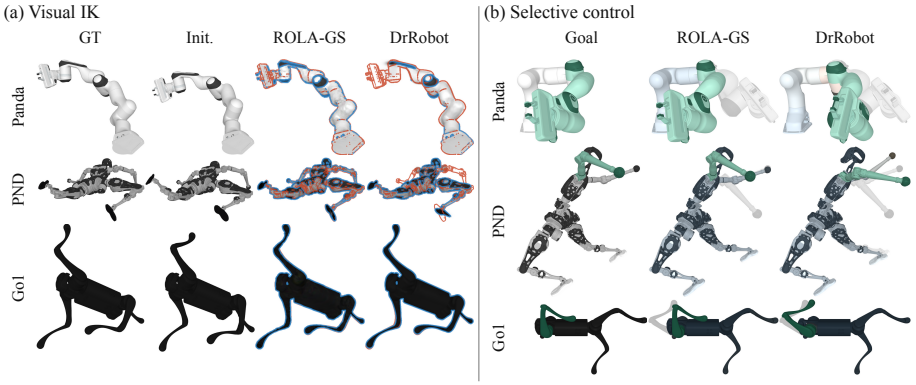


Figure 4: **Visual IK and selective control on Panda, PND, and Go1.** In (a), blue recovered contours and orange unmatched target boundaries show pose discrepancies. In (b), final and initial poses are overlaid to compare target motion and protected-link displacement.

remaining errors: DrRobot leaves larger orange target-boundary mismatches at the Panda gripper and PND limbs, whereas ROLA-GS's blue recovered contours align more closely.

Selective control measures target error and protected-link displacement across 300 paired cases at three target ranges.

ROLA-GS minimizes both cross-robot mean errors at every target range. At 20%, its target error is 45.01 mm versus DrRobot's 103.66 mm, a 56.6% reduction, while protected-link displacement falls from 4.10 to 2.02 mm. Soft link blending reaches 56.83/3.17 mm, isolating the additional benefit of fixed link assignments for both quantities. In Figure 4(b), the Panda gripper, PND arm, and Go1 leg approach the displayed goals more closely under ROLA-GS. The initial-pose overlays distinguish intended motion from changes elsewhere. The lower target error and protected-link displacement together support localized action. This is consistent with descendant-only spatial Jacobians: each joint affects only its subtree, although optimizing multiple joints need not leave every protected link fixed.

4.4 Mechanistic Ablations and Efficiency

Table 3: **Ablations of kinematic transport and appearance correction.** Results are averaged over five robots and three seeds.

(a) Kinematic transport

Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	CD (10^{-3} m^2) ↓	Fwd. (ms) ↓	Fwd. + bwd. (ms) ↓
w/ Learned Deformation	28.43	0.9765	0.0593	19.596	11.56	**19.96**
ROLA-GS	**35.67**	**0.9861**	**0.0366**	**3.905**	**7.15**	28.19

(b) Appearance

Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Fwd. (ms) ↓	Fwd. + bwd. (ms) ↓
w/ World-frame SH	34.31	0.9846	0.0387	**6.24**	**25.93**
w/ Link-normal	35.34	0.9856	0.0373	7.30	28.40
w/o Appearance Bound	**36.19**	**0.9865**	**0.0360**	7.09	28.09
ROLA-GS	35.67	0.9861	0.0366	7.15	28.19

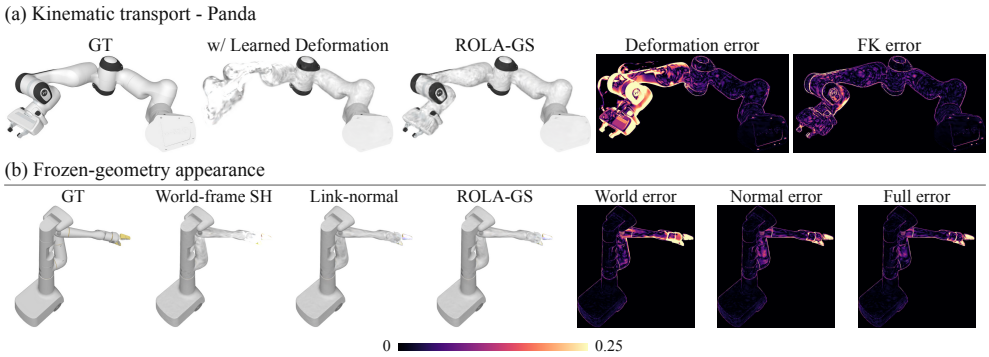


Figure 5: **Qualitative ablations of kinematic transport and appearance correction.** The top row compares learned deformation with exact relative FK, and the bottom row compares appearance variants with geometry frozen. RGB-error maps use a shared color scale.

Table 3 isolates transport and appearance with shared frozen geometry across five robots and three seeds, using native 1536^2 training and evaluation on an A100. Replacing learned deformation with exact relative FK raises mean PSNR from 28.43 to 35.67 dB, with gains in all 15 robot–seed pairs, and lowers squared center-Chamfer from 19.596 to $3.905 \times 10^{-3}\,\mathrm{m}^2$. Figure 5(a) localizes the difference at the wrist and gripper, where exact relative FK better preserves contours and reduces RGB error. Complete forward queries, including FK, transport, appearance, and rasterization, take 7.15 rather than 11.56 ms ($1.62\times$ faster). Adding image loss and joint-state backward takes 28.19 versus 19.96 ms, so the speedup applies to forward queries. Timings exclude controller steps. Appendix Table 13 measures Gaussian-parameter gradients separately.

With transport fixed, appearance conditioning improves fidelity without moving Gaussian centers: all four FK variants retain the same 3.905 center-Chamfer. World-frame SH, link-normal, and the full model achieve 34.31, 35.34, and 35.67 dB, respectively, with the view term improving all 15 pairs (Appendix Table 9). LPIPS decreases from 0.0387 to 0.0373 and 0.0366, a 5.4% reduction relative to world-frame SH alongside a 1.37 dB PSNR gain.

The extended-arm crops in Figure 5(b) compare appearance at fixed geometry and opacity, so RGB-error differences reflect appearance conditioning. Bounding the correction imposes a separate constraint on radiance magnitude: removing it raises held-out PSNR/SSIM from $35.67/0.9861$ to $36.19/0.9865$. The bound therefore limits correction magnitude at a photometric cost.

Finally, the link-assignment audit measures motion locality (Appendix Table 6). Under a commanded 10-mm single-joint articulation, soft assignment produces 6.993 mm target displacement and 0.171 mm off-tree displacement, whereas fixed assignment produces 10.000 mm and numerical-zero off-tree displacement, complementing the downstream assignment comparison in Table 2. The audit measures responses to a fixed command under differing geometry-training conditions. Together with the optimized control results, it distinguishes a structural locality guarantee from the residual error of a multi-joint task. The latter also depends on the target configuration and optimization trajectory.

5 Conclusion

We presented ROLA-GS, a Gaussian visual self-model separating known rigid articulation from learned appearance. Fixed link assignments and exact relative FK connect reconstruction to joint-state optimization while preserving within-link geometry. Real RGB and MuJoCo experiments show reconstruction gains, paired tasks show lower average visual IK and selective-control errors than DrRobot, and controlled ablations show faster forward queries than learned deformation.

Limitations and future work. The method assumes known kinematics and calibrated cameras. Real RGB tests cover reconstruction, including observed-frame H1 fidelity. Future work will address kinematic and calibration errors and evaluate hardware closed-loop control.

AI USE STATEMENT

Generative AI tools assisted with literature search, experimental design and ablation planning, code implementation and debugging, interpretation of results, manuscript drafting and revision, and figure and table preparation. The authors are responsible for the final text, claims, code, and experimental evidence.

ETHICS STATEMENT

ROLA-GS models robot appearance and kinematic state but is not a safety controller. Its outputs should be validated with standard robotic safety procedures before use in planning or human-facing deployment. We use public or locally generated robot datasets and report appearance failures and protocol boundaries alongside favorable results.

REPRODUCIBILITY STATEMENT

Experimental Setup describes the benchmark and evaluation metrics. The *Supplementary Experimental Evidence* appendix details experimental protocols, baseline comparisons, and supplementary evaluations, while *Supplementary Method Details* gives coordinate conventions, appearance and training objectives, optimization stages, and state-gradient derivations. These sections specify the settings and assumptions used in the reported experiments.

REFERENCES

James F. Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 192–198, 1977. doi: 10.1145/563858.563893.

Boyuan Chen, Robert Kwiatkowski, Carl Vondrick, and Hod Lipson. Full-body visual self-modeling of robot morphologies. *Science Robotics*, 7(68):eabn1944, 2022. doi: 10.1126/scirobotics.abn1944.

Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. Vcr-gaus: View consistent depth-normal regularizer for gaussian surface reconstruction. In *Advances in Neural Information Processing Systems*, 2024.

Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Hang Dai, Hongwei Fan, Han Zhang, Duojin Wu, Jiyao Zhang, and Hao Dong. Freeartgs: Articulated gaussian splatting under free-moving scenario. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11777–11787, 2026.

Antoine Guedon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Junfu Guo, Yu Xin, Gaoyi Liu, Kai Xu, Ligang Liu, and Ruizhen Hu. Articulatedgs: Self-supervised digital twin modeling of articulated objects using 3d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Kejun Hu, Peng Yu, and Ning Tan. Learning high-fidelity robot self-model with articulated 3d gaussian splatting, 2025. arXiv preprint arXiv:2503.05398.

Shoukang Hu et al. Gauhuman: Articulated gaussian splatting from monocular human videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation, 2025a. arXiv preprint arXiv:2510.20813.

Kaiwen Jiang, Venkataram Sivaram, Cheng Peng, and Ravi Ramamoorthi. Geometry field splatting with gaussian surfels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5752–5762, 2025b.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023.

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*, pp. 1654–1663, 2021.

Gahye Lee, Gyoonseo Kim, Wonjong Jang, Jooeun Son, and Seungyong Lee. Structuregs: Structure-aware gaussian splatting for articulated object reconstruction, 2026a. arXiv preprint arXiv:2607.26889. To appear in ECCV.

Inseo Lee, Yoonji Kim, Eugene Sohn, Jiwoong Lee, Jungmin You, Joonseok Lee, and Jin-Hwa Kim. Artsplat: Feed-forward articulated 3d gaussian splatting from sparse multi-state uncalibrated views, 2026b. arXiv preprint arXiv:2605.24304.

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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9426–9432, 2020. doi: 10.1109/ICRA40945.2020.9196596.

Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Stanley Lewis, Vishal Chandra, Tom Gao, and Odest Chadwicke Jenkins. Splatart: Articulated gaussian splatting with estimated object structure, 2025. arXiv preprint arXiv:2506.12184.

Sizhe Lester Li, Annan Zhang, Boyuan Chen, Hanna Matusik, Chao Liu, Daniela Rus, and Vincent Sitzmann. Controlling diverse robots by inferring Jacobian fields with deep networks. *Nature*, 643:89–95, 2025. doi: 10.1038/s41586-025-09170-0.

Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator, 2024. arXiv preprint arXiv:2411.11839.

Yunlong Li, Boyuan Chen, Chongjie Ye, Bohan Li, Zhaoxi Chen, Shaocong Xu, Hao Tang, and Hao Zhao. Gradrobot: Geometry-aware rendering with articulation and diffusion for robot modeling. In *International Conference on 3D Vision*, 2026.

Shengjie Lin, Jiading Fang, Muhammad Zubair Irshad, Vitor Campagnolo Guizilini, Rares Andrei Ambrus, Greg Shakhnarovich, and Matthew R. Walter. Splart: Articulation estimation and part-level reconstruction with 3d gaussian splatting. In *IEEE/CVF International Conference on Computer Vision*, pp. 8841–8851, 2025.

Ruoshi Liu, Alper Canberk, Shuran Song, and Carl Vondrick. Differentiable robot rendering, 2024. arXiv preprint arXiv:2410.13851.

Yu Liu, Baoxiong Jia, Ruijie Lu, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Building interactable replicas of complex articulated objects via gaussian splatting. In *International Conference on Learning Representations*, 2025.

Bui-Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18 (6):311–317, 1975. doi: 10.1145/360825.360839.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025.

Hao Wang, Xiaobao Wei, Ying Li, Qingpo Wuwu, Dongli Wu, Jiajun Cao, Ming Lu, Wenzhao Zheng, and Shanghang Zhang. Roboarmgs: High-quality robotic arm splatting via bezier curve refinement, 2025. arXiv preprint arXiv:2511.17961.

Haowen Wang, Xiaoping Yuan, Zhao Jin, Zhen Zhao, Zhengping Che, Yousong Xue, Jin Tian, Yakun Huang, and Jian Tang. Pd^{2}gs: Part-level decoupling and continuous deformation of articulated objects via gaussian splatting. In *International Conference on Learning Representations*, 2026.

Jiepeng Wang, Yuan Liu, Peng Wang, Cheng Lin, Junhui Hou, Xin Li, Taku Komura, and Wenping Wang. Gaussurf: Geometry-guided 3d gaussian splatting for surface reconstruction. *arXiv preprint arXiv:2411.19454*, 2024.

Di Wu, Liu Liu, Linli Zhou, Anran Huang, Liangtu Song, Qiaojun Yu, Qi Wu, and Cewu Lu. Reartgs: Reconstructing and generating articulated objects via 3d gaussian splatting with geometric and motion constraints. In *Advances in Neural Information Processing Systems*, 2025.

Di Wu, Liu Liu, Anran Huang, Yuyan Liu, Qiaojun Yu, Shaofan Liu, Liangtu Song, and Cewu Lu. Reartgs++: Generalizable articulation reconstruction with temporal geometry constraint via planar gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1177–1186, 2026.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20310–20320, 2024.

Hang Yang, Le Hui, Jianjun Qian, Jin Xie, and Jian Yang. Gsrecon: Efficient generalizable gaussian splatting for surface reconstruction from sparse views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25346–25356, 2025.

Ziyi Yang et al. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Kai Ye, Chong Gao, Guanbin Li, Wenzheng Chen, and Baoquan Chen. Geosplatting: Towards geometry guided gaussian splatting for physically-based inverse rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28991–29000, 2025. doi: 10.1109/ICCV51701.2025.02692.

Tianjiao Yu, Vedant Shah, Muntasir Wahed, Ying Shen, Kiet A. Nguyen, and Ismini Lourentzou. Part^{2}gs: Part-aware modeling of articulated objects using 3d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18913–18923, 2026.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

A Supplementary Experimental Evidence

Additional poses, resolutions, and baseline input protocols are evaluated separately from the main reconstruction, state-estimation, control, and component ablation averages (Tables 1–3). Panda, Google, ViperX, Shadow, H1, Go1, and PND abbreviate Franka Panda, Google Robot, ViperX 300, Shadow Hand, Unitree H1, Unitree Go1, and PND Adam Lite. The additional UR5e experiments use the Universal Robots UR5e model.

A.1 Additional Reconstruction Views

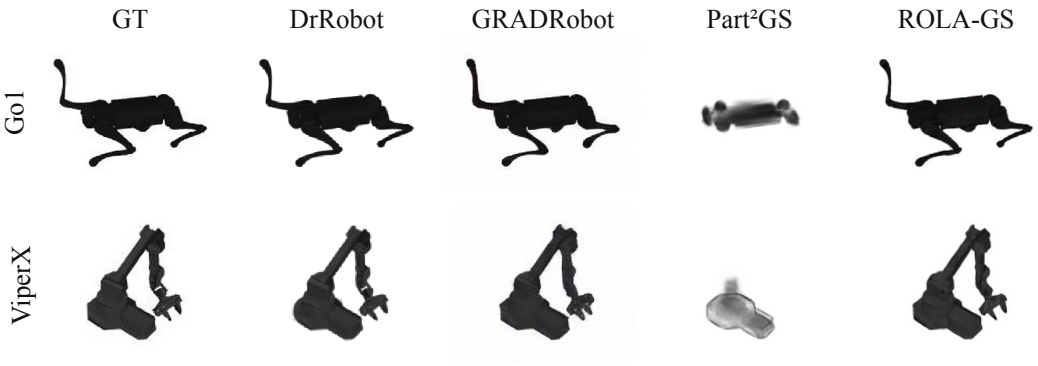


Figure 6: **Additional reconstruction comparisons.** Go1 and ViperX use the same methods, aligned views, and shared per-row crops as Figure 3.

Go1 and ViperX extend the main qualitative comparison to a quadruped and an additional arm, retaining the original views and common crops (Figure 6). Together with the real sensors, dexterous hand, and humanoid in Figure 3, they cover the six original comparison rows. The separate 11-robot figures below use the data-rich protocol, not the 96-state checkpoints used in the main comparison.

A.2 State-Estimation and Control Protocols

Both tasks use Panda, PND, Go1, and Shadow Hand. Visual IK includes 50 cases per robot at each of $d_0 \in \{0.025, 0.050, 0.100\}$: 200 cases per distance and 600 per method. Selective control includes 25 cases per robot at each of 5%, 10%, and 20% target-joint range: 100 cases per range and 300 per method. All four methods use paired case identifiers, targets, cameras, and initializations. All paired cases enter the reported averages. The DrRobot task checkpoints use the exported camera intrinsics during both training and task queries ($f \simeq 309.019$ pixels at 256^2, corresponding to a FoV of 0.785398). These checkpoints use the 10k-state training pool, distinct from the sparse-state reconstruction comparison in Table 1.

Optimization uses 200 Adam steps over normalized joint coordinates, clamped to $[-1, 1]$, with visual parameters frozen. Native 256^2 views are selected using GT foreground coverage with at least four pixels of silhouette clearance from the image border. The shared objective weights foreground Charbonnier, full-image Charbonnier, Sobel L_1, alpha L_1, and soft Dice by 0.45, 0.10, 0.15, 0.15, and 0.15, respectively. Learning rates are method-specific (Table 4).

Table 4: **Joint-state optimization settings.** All methods use 200 steps and the same task objective.

Method	Visual IK learning rate	Control learning rate
w/ Learned Deformation	0.001	0.001
w/ Soft Link Blending	0.001	0.003
DrRobot	0.001	0.001
ROLA-GS	0.003	0.003

Hinge-angle MAE averages absolute errors over revolute joints in degrees. Control targets differ from initialization at one pre-registered joint, while optimization can update all joints. Target error is

the Euclidean target-frame position error, while protected-link displacement averages frame motion over links whose GT displacement is zero for that requested move. Both are reported in millimeters. Protected-link displacement is not required to be zero after multi-joint optimization, even with fixed link assignments.

Soft Link Blending replaces fixed link assignments with fixed convex weights over the two nearest nominal link surfaces. The recorded temperatures are 0.01593, 0.02974, 0.01770, and 0.01435 m for Panda, PND, Go1, and Shadow Hand, respectively. Target cases remain shared across methods. Visual-IK and MuJoCo control displays use 1024^2 and 1536^2 query replays, respectively (Figures 4, 14, and 15), without changing the 256^2 optimization protocol. For the per-robot results (Table 5), stored normalized control distances are converted back to millimeters using the same per-robot scales as the main table: 1.19302, 1.76216, 0.78545, and 0.49020 m for Panda, PND, Go1, and Shadow Hand. Performance varies across morphologies and initialization distances. On

Table 5: **Per-robot state estimation and control.** Each robot contributes 50 IK cases per distance and 25 control cases per target range, with the same 200-step budgets as Table 2.

	Hinge-angle MAE (°)↓			Target / protected (mm)↓		
Method	$d_0 = .025$	.050	.100	5%	10%	20%
Panda						
w/ Learned Deformation	11.53	12.56	24.21	158.15 / 9.17	195.78 / 9.49	297.86 / 9.19
w/ Soft Link Blending	1.48	5.82	23.11	8.72 / 1.27	23.28 / 1.44	81.23 / 1.51
DrRobot	7.17	9.42	21.64	57.62 / 0.91	79.51 / 1.31	173.16 / 2.85
ROLA-GS	0.86	3.51	18.78	6.59 / 0.65	18.72 / 1.06	84.50 / 1.10
PND						
w/ Learned Deformation	5.30	6.42	11.81	72.30 / 13.32	88.26 / 13.03	159.08 / 12.72
w/ Soft Link Blending	3.77	4.79	11.05	26.33 / 4.55	32.07 / 4.57	72.67 / 4.69
DrRobot	3.55	5.32	11.26	62.61 / 8.34	73.37 / 9.21	145.82 / 9.78
ROLA-GS	4.07	5.10	9.54	8.09 / 2.36	10.22 / 2.37	58.79 / 2.45
Go1						
w/ Learned Deformation	2.10	2.68	7.46	13.26 / 4.36	23.87 / 4.44	101.64 / 4.80
w/ Soft Link Blending	1.16	1.87	6.81	8.73 / 2.83	11.19 / 2.86	66.86 / 3.03
DrRobot	0.74	1.35	6.73	6.27 / 1.27	16.68 / 1.34	90.42 / 1.93
ROLA-GS	0.88	0.93	3.36	5.38 / 1.50	6.59 / 1.50	31.47 / 1.55
Shadow Hand						
w/ Learned Deformation	4.78	5.01	6.96	10.14 / 6.17	9.67 / 6.21	10.12 / 6.10
w/ Soft Link Blending	2.51	2.84	5.46	6.81 / 3.53	6.29 / 3.47	6.57 / 3.46
DrRobot	2.10	2.71	5.45	3.70 / 1.63	4.04 / 1.62	5.25 / 1.85
ROLA-GS	2.89	3.02	4.58	5.05 / 3.01	5.38 / 3.01	5.28 / 2.96

Shadow Hand, DrRobot has lower hinge-angle MAE at $d_0 = 0.025$ and 0.050, while ROLA-GS has lower MAE at $d_0 = 0.100$. On PND, DrRobot leads at $d_0 = 0.025$ and ROLA-GS at the two larger distances. Panda favors ROLA-GS at all three distances. On Go1, DrRobot leads at $d_0 = 0.025$ and ROLA-GS at the two larger distances. For Shadow Hand selective control, DrRobot has lower target error and protected-link displacement at all three target ranges. On Go1, DrRobot also has lower protected-link displacement at the two smaller target ranges, while ROLA-GS leads at the largest range. The main task averages summarize these per-robot differences under the method-specific learning rates in Table 4.

A.3 Link-Assignment Audit

The commanded single-joint displacement evaluates link assignment without task optimization. Its off-tree displacement is therefore distinct from protected-link displacement after optimizing all joints in Table 2. The soft and fixed rows use the representation-study variants `soft_geometry_fk` and `hard_fk_fixed_geometry`. They differ in training-time geometry updates as well as link assignment, so this audit is not a geometry-matched replacement of the Table 2 Soft Link Blending experiment.

Table 6: **Link-assignment audit.** Target displacement is compared with a commanded 10 mm, while off-tree displacement measures motion outside the affected subtree.

Method	Target displacement (mm, reference: 10)	Off-tree displacement (mm) ↓
Soft link assignment	6.993	0.171
Fixed link assignment	10.000	**0.000**

A.4 Unified Component Ablations

Component ablations average five robots (Panda, Google Robot, PND, Shadow Hand, and UR5e) and seeds 53, 83, and 113 (Table 3). MuJoCo generates native 1536^2 RGB observations at the original joint states and cameras, with 64 canonical, 96 training, and 24 held-out states and 12 views per state. RGB files use the same JPEG quality setting of 75 for every variant. Each robot–seed pair retrains its shared geometry with 2k canonical and 120k posed updates. Each variant starts from the same robot–seed FK checkpoint and fixes its 120k Gaussian centers, scales, rotations, opacities, and link assignments. Appearance training uses 120k updates with the same image schedule. The learned-deformation variant replaces only transport with frozen learned weights, whereas the unbounded variant removes the correction's $\tanh$ mapping, retaining the same residual scale. Quality averages 24 held-out poses and 12 views per pose, while center-Chamfer averages the 24 posed clouds. Appearance-only variants therefore have identical center-Chamfer, whereas changing transport changes the posed clouds. LPIPS (Zhang et al., 2018) uses the pretrained AlexNet backbone and learned v0.1 weights on the same full 1536^2 RGB images, mapped from $[0, 255]$ to $[-1, 1]$ without masking or cropping. Each replay is checked against its previously admitted PSNR.

The qualitative ablations use native 1536^2 renders of the recorded Table 3 checkpoints at seed 53 (Figure 5), using Panda's `test_sample_5_cam03` and Google Robot's `test_sample_0_cam00`. For each fixed pose, the first camera in index order whose GT silhouette stays five pixels inside the image boundary is used. The GT bounding box plus 20 pixels defines the shared crop. Corresponding methods share the camera, crop, and error scale, using the recorded test RGB as reference. Replaying all five variants under the evaluation precision settings reproduces the recorded per-view PSNR within 10^{-6} dB. Training and evaluation both use 1536^2. The shared arm crop in Figure 7 enlarges the same pixels without changing contrast or selecting another test frame.

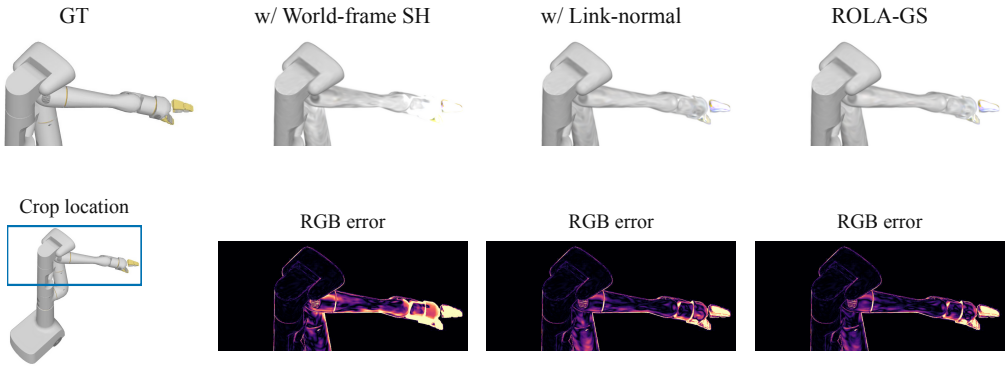


Figure 7: **Appearance correction at the extended arm.** All variants use the marked GT crop, with RGB errors on the shared $[0, 0.25]$ scale.

Timing uses one A100 per evaluator, with all five variants of a robot–seed pair measured on the same GPU at 1536^2 in the same variant order. For each of 24 poses, each query has 10 warmup and 30 synchronized wall-clock measurements, with alternating forward/backward order. Forward recomputes FK from joint state and includes transport, appearance, and rasterization. Forward + backward additionally computes full-image MSE and its joint-state gradient with visual parameters frozen, excluding loading, export, and optimizer steps. This workload differs from the Gaussian-

parameter gradient benchmark in Table 13. Removing the appearance bound increases PSNR in all 15 pairs (mean 0.521 dB), quantifying the photometric cost of the bounded correction in this frozen-geometry setting.

Per-robot results for all five variants first average the evaluated views or poses of a trained model, then the three seeds (Tables 7 and 8). Averaging the five robot entries recovers Table 3. The four appearance variants share frozen geometry and therefore identical CD. Latency averages use the same recorded workloads as the main table and do not measure optimizer steps or hardware control loops.

Table 7: **Per-robot component ablations (1/2).** Results average three seeds under the Table 3 protocol.

Method	PSNR (dB)↑	SSIM↑	LPIPS↓	CD (10^{-3} m^2)↓	Fwd. (ms)↓	Fwd.+bwd. (ms)↓
Panda						
w/ Learned Deformation	26.85	0.9734	0.0803	27.510	10.13	17.71
w/ World-frame SH	33.51	0.9810	0.0541	0.380	5.95	16.48
w/ Link-normal	34.26	0.9822	0.0524	0.380	6.95	18.91
w/o Appearance Bound	34.95	0.9833	0.0503	0.380	6.65	18.53
ROLA-GS	34.78	0.9831	0.0510	0.380	6.72	18.67
Google Robot						
w/ Learned Deformation	36.23	0.9878	0.0334	1.113	11.39	20.65
w/ World-frame SH	34.46	0.9865	0.0351	1.092	6.71	35.92
w/ Link-normal	36.72	0.9879	0.0326	1.092	7.88	38.54
w/o Appearance Bound	38.40	0.9892	0.0309	1.092	7.64	38.12
ROLA-GS	37.23	0.9884	0.0320	1.092	7.59	38.00
PND						
w/ Learned Deformation	21.66	0.9538	0.0938	68.446	12.98	21.37
w/ World-frame SH	33.08	0.9809	0.0275	17.661	6.04	15.61
w/ Link-normal	34.46	0.9819	0.0264	17.661	7.05	18.08
w/o Appearance Bound	35.81	0.9829	0.0256	17.661	6.91	17.83
ROLA-GS	34.65	0.9824	0.0263	17.661	7.01	18.00

Table 8: **Per-robot component ablations (2/2).** Results average three seeds under the Table 3 protocol.

Method	PSNR (dB)↑	SSIM↑	LPIPS↓	CD (10^{-3} m^2)↓	Fwd. (ms)↓	Fwd.+bwd. (ms)↓
Shadow Hand						
w/ Learned Deformation	28.72	0.9891	0.0347	0.199	13.70	22.58
w/ World-frame SH	36.37	0.9927	0.0285	0.140	6.62	36.45
w/ Link-normal	36.46	0.9928	0.0283	0.140	7.70	38.86
w/o Appearance Bound	36.48	0.9928	0.0283	0.140	7.55	38.53
ROLA-GS	36.47	0.9928	0.0284	0.140	7.71	38.84
UR5e						
w/ Learned Deformation	28.70	0.9787	0.0544	0.709	9.57	17.50
w/ World-frame SH	34.12	0.9819	0.0483	0.251	5.88	25.17
w/ Link-normal	34.79	0.9830	0.0467	0.251	6.92	27.61
w/o Appearance Bound	35.31	0.9840	0.0450	0.251	6.68	27.42
ROLA-GS	35.23	0.9839	0.0453	0.251	6.73	27.45

A.5 Appearance across Training Seeds

Adding the view term improves PSNR in all 15 paired robot–seed comparisons, with a mean increase of 0.334 dB. The magnitude varies by robot: the increase is 0.520 dB on Panda and 0.012 dB on Shadow Hand (Table 9). These differences are paired across training seeds, rather than treating views from the same trained model as independent repetitions.

A.6 Reference Ablations at Lower Resolution

Table 10 retains the original 256^2 training and evaluation on the same five robots and three seeds. It supplies the teaser's $1.55\times$ forward-query comparison and the checkpoints for the bound visual-IK study below. The native 1536^2 replication retrains shared geometry and appearance, and evaluates images at its own resolution. Differences between these tables therefore combine training and evaluation resolution rather than isolating training resolution on a common test grid. Center-Chamfer also changes with the retrained geometry.

A.7 Appearance Bound in Joint-State Optimization

The bounded and unbounded 256^2 checkpoints from Table 10 are compared on all 24 held-out poses per robot and seed. Each pose uses the camera with the largest GT foreground, selected independently of either prediction. Both variants start at identical normalized joint-range distances 0.025, 0.050, and 0.100 and use the same image objective and 200 Adam steps at learning rate 0.003, with visual parameters fixed. This gives 2,160 optimizations across five robots and three seeds. The bound reduces mean hinge-angle MAE by 0.06°, with a lower robot–seed mean in 8 of 15 pairs (Table 11). Both variants complete all cases with finite losses and gradients. The small, mixed task effect complements the photometric trade-off rather than establishing a general stability advantage.

A.8 Transport and Density Control

Exact relative FK improves mean PSNR over learned deformation by 3.97 dB with fixed density and 3.72 dB with adaptive density, while reducing mean squared center-Chamfer in both settings. Adaptive density improves mean center-Chamfer but lowers PSNR for both transport mechanisms, so the FK advantage does not depend on a photometric benefit from density control.

A.9 Baseline Adaptation and Training Schedules

The main MuJoCo comparison supplies the same RGB images, calibrated cameras, and joint configurations to the compared methods. The Part^{2}GS and REArtGS++ entries are joint-conditioned adaptations, not their native two-state pipelines. A learned mapping takes the full normalized joint configuration to rigid dual-quaternion transforms of the learned segments, retaining a static root segment and the corresponding renderer. These adapters use RGB L_1 and DSSIM supervision without supplied link assignments, FK transforms, depth, or Chamfer supervision. Their native two-state evaluations below are separate experiments. For 4D-GS, the official no-grid deformation input is expanded from spatial coordinates and time to spatial coordinates and the full normalized joint configuration. The one-dimensional time grid is omitted. The reported variant freezes base opacity after coarse training while retaining deformation, RGB supervision, and density control during posed training.

The paired Gaussian training runs use 2k canonical and 120k posed updates. GRADRobot instead retains its staged schedule, with 30k geometry updates, 30k appearance updates, and a 105-epoch LRS stage at batch size eight. Thus, shared training observations do not imply identical update counts or optimization schedules across all baselines.

The admitted 4D-GS adaptation exhibits severe rendering failures on Panda, ViperX, and Shadow Hand. The Panda diagnostic finds finite but displaced Gaussian centers on both training and held-out states, with approximately 11.7 m mean per-state 95th-percentile displacement. All states remain in the reported metrics. This diagnostic establishes a failed fitted representation, not a causal attribution to missing kinematic priors or a test-only failure.

Table 9: **Effect of the view term.** Entries are mean and sample standard deviation of paired PSNR differences across three training seeds, with geometry frozen.

Robot	ΔPSNR (dB)
Panda	0.520 ± 0.009
Google Robot	0.509 ± 0.008
PND	0.188 ± 0.004
Shadow Hand	0.012 ± 0.002
UR5e	0.439 ± 0.019

Table 10: **Component ablations at 256^2.** Results are averaged over five robots and three seeds.

(a) Kinematic transport

Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	CD (10^{-3} m^2) ↓	Fwd. (ms) ↓	Fwd. + bwd. (ms) ↓
w/ Learned Deformation	28.23	0.9592	0.0763	17.748	11.38	**20.01**
ROLA-GS	**33.02**	**0.9767**	**0.0275**	**2.184**	**7.33**	30.29

(b) Appearance

Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Fwd. (ms) ↓	Fwd. + bwd. (ms) ↓
w/ World-frame SH	32.29	0.9745	0.0307	**6.56**	**28.22**
w/ Link-normal	32.92	0.9762	0.0279	7.40	30.44
w/o Appearance Bound	**33.25**	**0.9773**	**0.0265**	7.33	30.26
ROLA-GS	33.02	0.9767	0.0275	7.33	30.29

Table 11: **Appearance bound in visual IK.** Hinge-angle MAE (°)↓ averages three seeds and all initialization distances.

Robot	ROLA-GS	w/o Appearance Bound
Panda	12.73	12.97
Google Robot	13.76	13.57
PND	6.81	6.84
Shadow Hand	3.93	3.92
UR5e	35.37	35.60
Avg.	14.52	14.58

A.10 Metric and Protocol Details

Let A_n be rendered alpha and M_n the target robot mask. We report foreground PSNR only over pixels in M_n and silhouette IoU as

$$\mathrm{IoU}(A_n, M_n) = \frac{|\mathbf{1}[A_n > \tau] \cap M_n|}{|\mathbf{1}[A_n > \tau] \cup M_n|}. \tag{13}$$

For the MuJoCo benchmark, the 500 held-out states each have 12 frames from 8 unique cameras. Per-pose bootstrap samples average frames before resampling states. Foreground-only diagnostics exclude empty GT masks, whereas the official full-frame PSNR and center-Chamfer results retain every held-out state and frame.

The strict data generator stores 256×256 images, matching the released DrRobot generator default, and the loader reads that native size without resizing. The data-rich protocol uses the same ordered 2k canonical + 120k posed RGB updates for both methods, with 10k training states and 500 held-out states. The 96-state main benchmark is a separate training pool.

A.11 Data-Rich Breadth and High-Resolution Controls

The 10k-state breadth test compares against the released full-appearance DrRobot configuration (Table 14), using the paper's full-frame PSNR and unfiltered bidirectional squared center-Chamfer,

Table 12: **Transport and density control.** All four configurations use the same five robots, three seeds, training schedule, and held-out views.

Method	Density	PSNR (dB) ↑	Chamfer Distance (10^{-3} m^2) ↓
Learned deformation	Fixed	28.83	12.907
Exact relative FK	Fixed	32.80	3.629
Learned deformation	Adaptive	28.26	9.968
Exact relative FK	Adaptive	31.99	2.184

Table 13: **Isolated transport ablation.** Quality uses five robots and three seeds with fixed density, while timing uses four matched 120k-Gaussian assets at 256^2.

Method	PSNR (dB) ↑	Chamfer Distance (10^{-3} m^2) ↓	Transport (ms) ↓	Forward (ms) ↓	Fwd. + bwd. (ms) ↓
Learned deformation	28.83	12.907	13.74	15.24	29.93
Exact relative FK	**32.80**	**3.629**	**3.06**	**6.78**	**15.69**

which does not account for opacity or visibility. The mean Chamfer is lower for ROLA-GS because of large Panda and PND improvements, while DrRobot remains lower on 9/11 individual robots, including H1. Thus the aggregate and per-robot rankings capture different aspects of geometric error.

The 512^2 stress test uses Chamfer-L1 rather than squared center-Chamfer. The released full-appearance runs on Shadow Hand and Go1 have zero foreground support, but their measured scores are retained.

A.12 Comparison with 3D-Supervised and Diffusion-Assisted Rendering

We evaluate GRADRobot (Li et al., 2026) with its released three-stage implementation. The real-RGB row in Table 1 uses its final image output. Its 2-D refinement neither changes the queryable Stage-2 Gaussians nor supplies the state gradients used by our estimation and control studies. We evaluate refined RGB separately from queryable 3-D geometry.

For Table 1, the real-data adapters preserve f_x, f_y, c_x, c_y through training and rendering. Image resizing applies the same scale to the corresponding intrinsic coordinates. Evaluation uses the original 640×480 RGB and common foreground masks. GRADRobot's fixed 256^2 Stage-3 output is bilinearly resized to this grid before scoring. Table 16 separates it from Stage 2. The adapters use the same train/test frames as the reconstruction comparison. DrRobot uses 15k posed steps on each real dataset (2k canonical steps on Azure), while its seven MuJoCo runs use 2k canonical and 120k posed steps with the original image schedules. The Azure result replaces an earlier 3k-posed-step run, so its change includes a training-budget difference. Deformable 3DGS uses 40k steps, while 4D-GS uses 3k coarse and 30k fine steps. Part^{2}GS and REArtGS++ use 2k canonical and 30k posed steps. GRADRobot uses 30k steps in each of its first two stages and 20 Stage-3 epochs.

A.13 Full 11-Robot Qualitative Comparison

Under the 10k-state protocol (Table 14), each robot's qualitative example is the held-out frame nearest its median paired FG-PSNR difference, with a common crop across methods (Figures 8–11).

A.14 Qualitative Selection and Provenance

The teaser's visual-IK curve is one H1 pose-recovery illustration, distinct from Table 2. Both methods use the same target pose and initialization, with cameras 3, 5, and 7 for the first 1,000 updates and all 12 cameras thereafter. Their learning-rate schedules are method-specific. The markers at 572 and 3,184 updates indicate the first of 20 consecutive states with mean joint-angle error at most 1° and maximum error at most 3°. Their ratio, $3184/572 \simeq 5.57$, gives the teaser's reduction in IK



Figure 8: **Robot reconstruction with 10k training states (1/4).** GT, ROLA-GS, and DrRobot are compared across 11 robots over four figures.



Figure 9: **Robot reconstruction with 10k training states (2/4).** Paired renders compare link surfaces and arm contours on Panda, Google Robot, and ViperX 300.



Figure 10: **Robot reconstruction with 10k training states (3/4).** Paired renders compare arm and leg contours on xArm7, Go1, and Go2.



Figure 11: **Robot reconstruction with 10k training states (4/4).** Paired renders compare link surfaces and arm contours on UR5e and PND Adam Lite.

Table 14: **Data-rich reconstruction across 11 robots.** Each robot uses 10k posed training states.

Method	Panda	Google	ViperX	xArm7	UR5e	Shadow
Full-frame PSNR (dB) ↑						
DrRobot	28.148	36.607	**31.955**	**35.482**	**32.274**	14.949
ROLA-GS	**32.552**	**36.899**	31.931	34.022	32.156	**32.889**
Chamfer Distance ($10^{-3}\,\mathrm{m}^2$) ↓						
DrRobot	13.880	**1.243**	**0.096**	**0.371**	**0.328**	**0.206**
ROLA-GS	**0.504**	2.113	0.165	0.373	0.456	0.247

Method	G1	H1	Go1	Go2	PND	Avg.
Full-frame PSNR (dB) ↑						
DrRobot	29.938	28.760	12.742	**32.535**	28.537	28.357
ROLA-GS	**30.737**	**29.595**	**30.603**	32.400	**31.104**	**32.263**
Chamfer Distance ($10^{-3}\,\mathrm{m}^2$) ↓						
DrRobot	**0.339**	**0.589**	**0.417**	**0.221**	9.737	2.493
ROLA-GS	0.673	1.450	0.511	0.502	**6.412**	**1.219**

Table 15: Reconstruction at 512×512 with a shared ordered RGB schedule.

Method	Panda	Shadow	Go1	Avg.
FG-PSNR (dB) ↑				
DrRobot	17.434	2.321	1.354	7.037
ROLA-GS	**24.455**	**23.518**	**23.741**	**23.905**
Silhouette IoU ↑				
DrRobot	0.7572	0.0000	0.0000	0.2524
ROLA-GS	**0.9612**	**0.9561**	**0.9478**	**0.9550**
Symmetric Chamfer-L1 (mm) ↓				
DrRobot	40.262	8.318	12.143	20.241
ROLA-GS	**10.609**	**7.810**	**11.080**	**9.833**

Table 16: **Real-RGB image refinement with calibrated cameras.** Both stages use common-foreground PSNR (dB) on the original evaluation grid.

Output	Azure	ORB	RealSense	H1
GRADRobot Stage 2	13.402	15.662	13.089	1.520
GRADRobot Stage 3	16.679	17.739	18.189	4.849

steps for this H1 example, not a wall-clock speedup. The separate $1.55\times$ forward-query speedup is the ratio of mean learned deformation and exact relative FK latencies in Table 10 ($11.3769/7.3342$), measured on five robots and three seeds at 256^2 on an A100.

The robot poses and ellipsoids in Figure 2 are schematic, while its image comparisons use observed-frame H1 outputs and a proxy mask. The main five-method montage uses the paired views and shared row crops preserved in its figure manifest. The four-row layout and Figure 6 only relocate those existing rows. The appearance panel uses the GT-based visibility rule described above, while Table 3 reports the aggregate over all 15 robot–seed pairs. Figures are embedded as PDFs with vector labels and raster evidence. Benchmark images retain native 256^2 resolution outside the native 1536^2 component ablations, and real-camera images retain 640×480 resolution. High-resolution query replays are identified separately from these native evaluation images.

A.15 Additional Qualitative Evidence

The main paper keeps compact representative views, while the panels below retain the complete synchronized comparisons, causal error maps, and real-camera frames. All crops and overlays are deterministic exports of the evaluated renders. The earlier appearance-design comparisons use their

original checkpoints and aligned 160^2 crops (Figures 17 and 19), with the Panda crops taken from a native 1024^2 query. These are separate from the Table 3 checkpoint replays in Figures 5 and 7. The selective-control views use 1536^2 MuJoCo query renders.

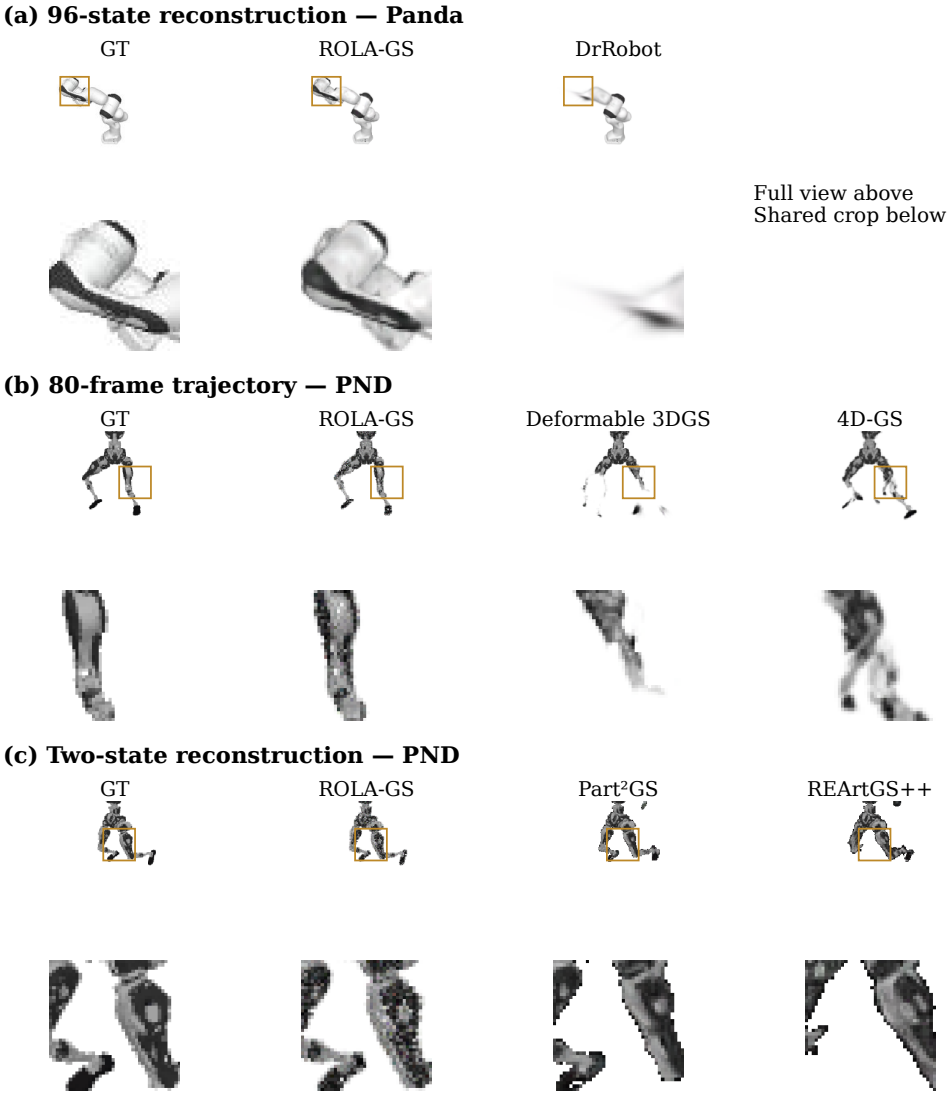


Figure 12: **Reconstruction under state, trajectory, and two-state supervision.** Rows compare matched target views under the 96-state, 80-frame, and two-endpoint protocols.

A.16 Released DrRobot Appearance-Branch Audit

The DrRobot paper reports full-frame PSNR. Under its released full-appearance 256^2 pipeline, 7/10 local values meet or exceed the published number. Shadow Hand and Go1 lose silhouette support in this branch, while Panda remains non-collapsed. This audit is separate from the stable appearance-disabled DrRobot representation used in the sparse MuJoCo block of Table 1, where Shadow Hand renders normally.

Switching off pose-conditioned appearance restores nonzero support on all 11 robots. The full branch fails only on Shadow Hand and Go1 in this stress test. This identifies a branch-specific instability rather than a general failure of DrRobot transport.

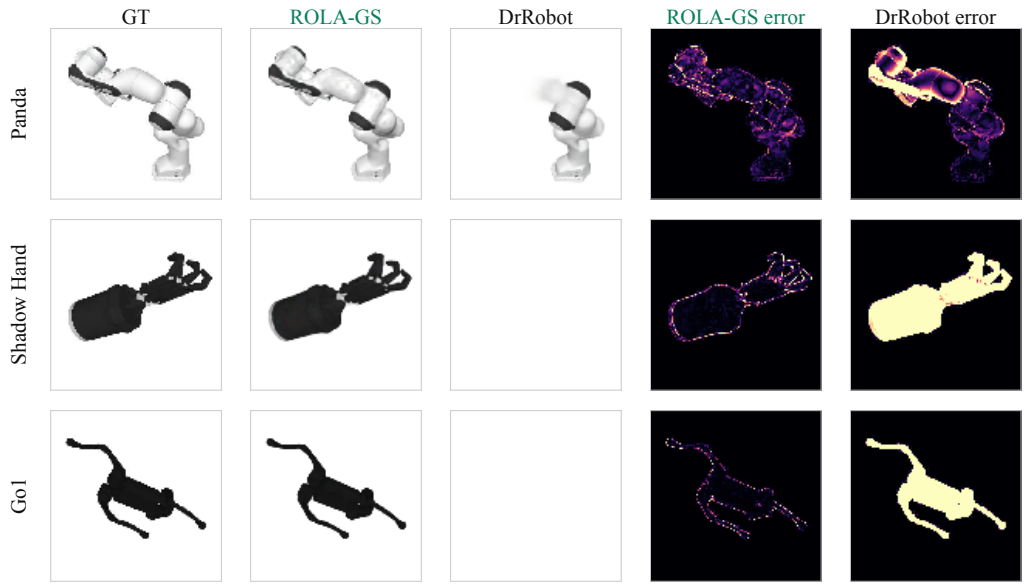


Figure 13: **Reconstruction at 512^2 on Panda, Shadow Hand, and Go1.** RGB-error maps use a shared $[0, 0.25]$ color scale.

Table 17: DrRobot released full-appearance branch on ten robots, using full-frame PSNR.

Robot	Published PSNR (dB) ↑	Reproduced PSNR (dB) ↑
Shadow Hand	31.440	14.949
Unitree G1	28.310	29.938
Unitree H1	27.770	28.760
Franka Panda	32.840	28.148
Google Robot	32.600	36.607
ViperX 300	30.590	31.955
xArm7	33.250	35.482
Unitree Go1	29.740	12.742
Unitree Go2	29.600	32.535
UR5e	32.010	32.274

A.17 Symmetric Input-Count Controls

Unequal observation counts confound reconstruction comparisons. We therefore evaluate Google Robot, Shadow Hand, UR5e, and ViperX with 1,920 or 34 shared images per method (Table 18). ROLA-GS leads FG-PSNR on all four robots at both counts. Three DrRobot runs lose all silhouette support with 34 images, identifying a low-data failure regime, while the 1,920-image setting provides the corresponding higher-coverage comparison.

A.18 Pose-Pool Scaling

We additionally vary the nested posed-state pool over $\{96, 500, 2000, 10000\}$ for Panda, Shadow Hand, and UR5e while keeping 64 canonical states, 500 held-out states, and 120k RGB updates fixed. All 12 paired settings completed. Mean ROLA-GS FG-PSNR is $21.549, 21.780, 21.691, 21.658$ dB, respectively, showing a nonmonotonic trend. Across these pool sizes, mean FG-PSNR varies by 0.231 dB under the fixed update budget. The corresponding mean silhouette IoU stays between 0.9370 and 0.9406, indicating a small aggregate response to pool size.

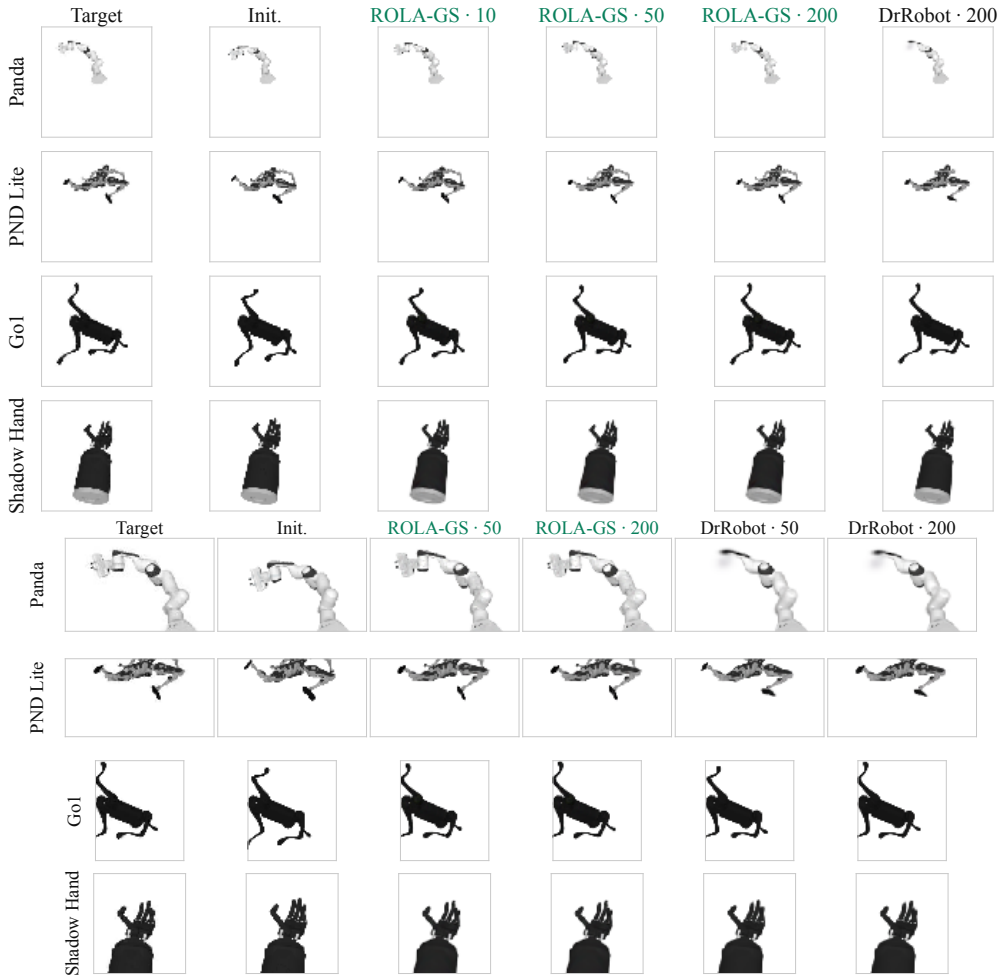


Figure 14: **Single-view visual IK across four robots.** Matched optimization snapshots and detail crops show how the rendered pose changes toward the target.



Figure 15: **Selective control across four robots.** Green marks the target link, and blue marks protected links.

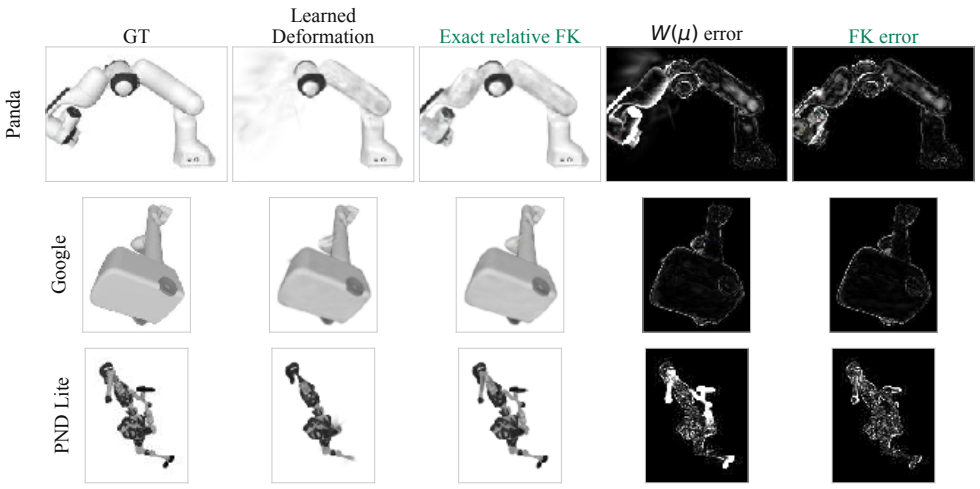


Figure 16: **Qualitative comparison of learned deformation and exact relative FK.** Paired RGB renders and error maps compare the transport mechanisms with all other components fixed.

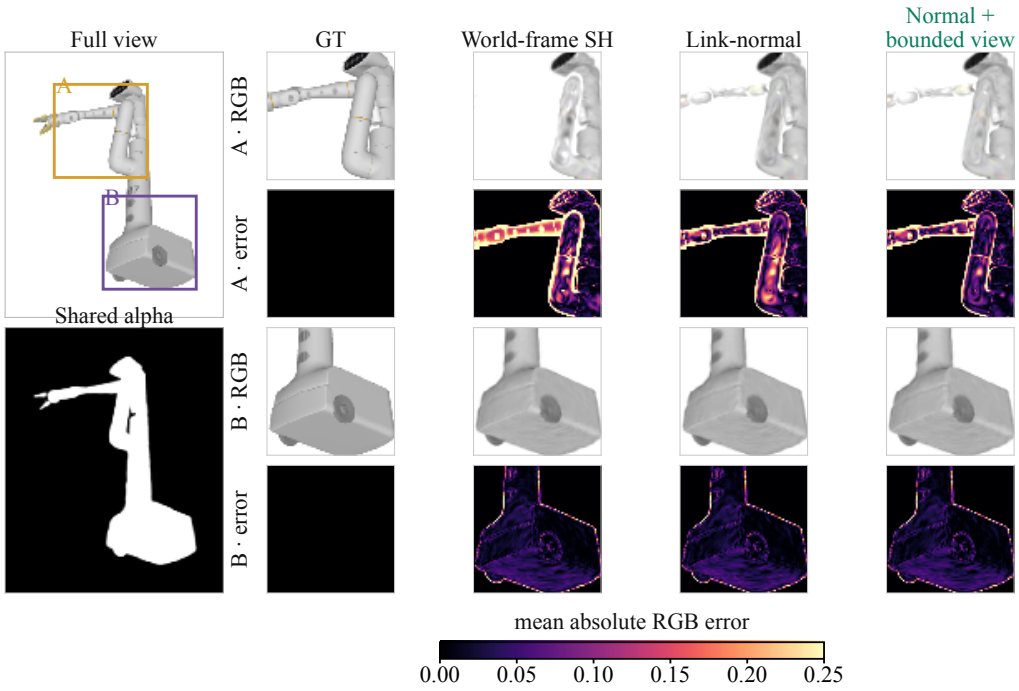


Figure 17: **Appearance correction with frozen geometry on Google Robot.** Paired RGB crops and error maps compare appearance variants with shared alpha.

Table 18: Foreground PSNR (dB ↑) under matched input counts.

1,920 shared RGB images				
Method	Google Robot	Shadow Hand	UR5e	ViperX 300
DrRobot	25.043	16.208	17.950	18.410
ROLA-GS	**25.053**	**22.322**	**21.465**	**18.970**
34 shared RGB images				
Method	Google Robot	Shadow Hand	UR5e	ViperX 300
DrRobot	15.178	4.103	5.466	3.207
ROLA-GS	**17.434**	**13.304**	**17.833**	**15.483**

A.19 Injected Link-Assignment Faults

We perturb a fraction of link-assignment weight and sweep one joint at a time. The clean discrete state has numerical-zero non-descendant motion, while wrong assignment produces monotonic off-tree displacement (Table 19). This turns the structural invariant into a measurable response to link-assignment errors.

A.20 Runtime Scope and Gradient Stability

The single-step gradient audit in Table 20 and Figure 18 retains its original checkpoint cohort, including DrRobot checkpoints from before the camera-intrinsics correction. It is separate from the corrected 200-step task results in Table 2.

The updated matched timing audit uses four robots, 120k Gaussians, three held-out poses, three order-varied blocks, and synchronized wall-clock measurements after warmup. The companion Visual IK replay covers 120 paired cases per method with 200 joint-state Adam updates. These task times are not combined with the mechanism timings below.

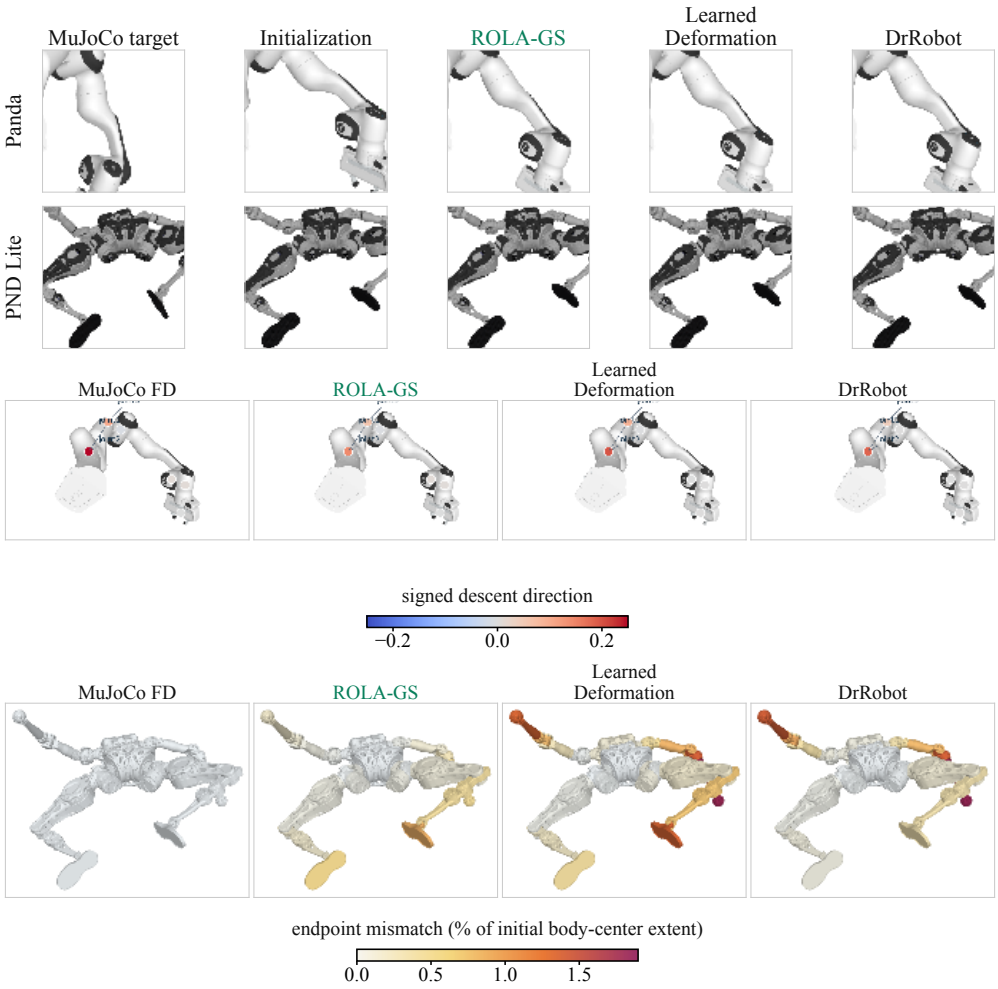


Figure 18: **Spatial audit of image-loss gradients.** The panels show MuJoCo update replays, signed joint influence, and link endpoint discrepancy.

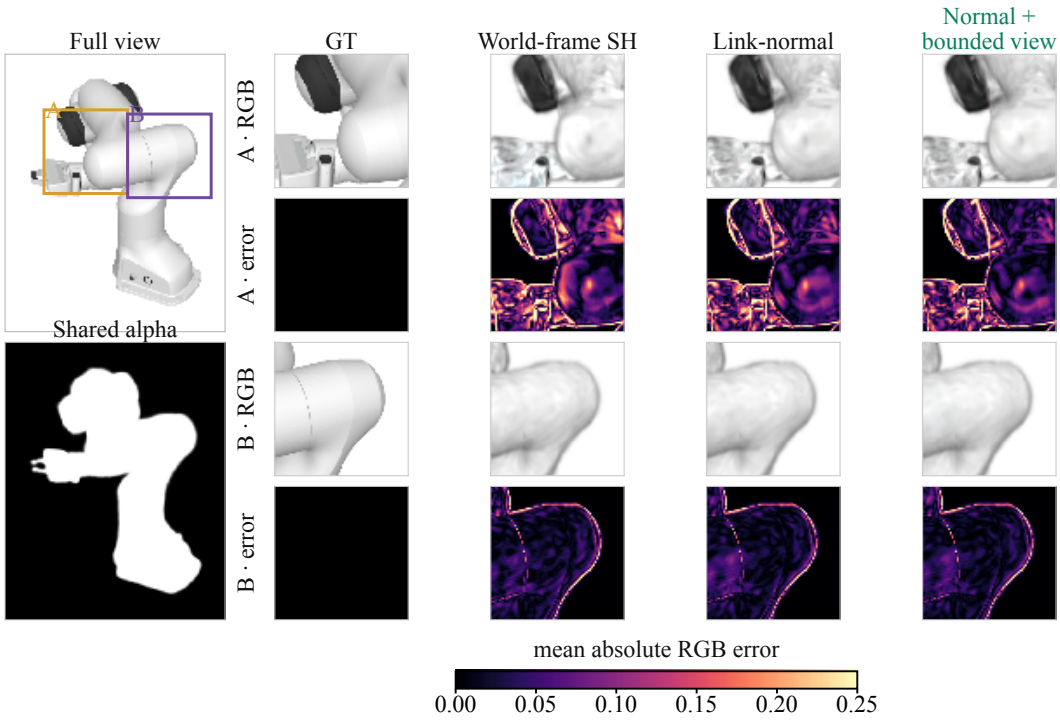


Figure 19: **Appearance correction with frozen geometry on Panda.** Paired RGB crops and error maps compare the appearance variants on a shared color scale.

Table 19: Off-tree displacement under injected wrong-link assignments.

Wrong-link mass	PND Adam Lite (mm)	UR5e (mm)
0%	0.0000	0.0000
1%	0.5409	2.6605
5%	2.8262	14.1621
20%	10.9014	54.7329

Table 20: Gradient cosine similarity to central finite-difference references in the MuJoCo renderer. Results average 100 cases at each initial distance across four morphologies.

Method	$d_0 = 0.025$	$d_0 = 0.050$
w/ Learned Deformation	0.769	0.834
DrRobot	<u>0.929</u>	<u>0.905</u>
ROLA-GS	**0.956**	**0.974**

We measure three workloads separately (Table 21). A forward query starts from joint state and includes FK construction and splatting. Equal-cardinality transport isolates the motion mechanism, whereas forward + backward adds RGB/alpha loss and Gaussian-parameter differentiation. These measurements use 120k Gaussians on Panda, PND, Shadow Hand, and Go1, and every tested gradient is finite. The resolution audit verifies output dimensions, scaled intrinsics, stable alpha coverage, and order-varied blocks. The measured latency trend persists across resolutions. The lower panel separately reports warmed two-state query times.

Table 21: **Runtime comparisons.** The upper block reports speedups over learned deformation, and the lower block reports two-state query latency.

Workload	Robots	256×256	512×512	1024×1024
Complete forward query	4	$2.23\times$	$2.41\times$	$2.47\times$
Forward + backward (120k)	4	$1.90\times$	$2.03\times$	$2.07\times$
Transport only (120k)	4	$4.47\times$ (raster-independent)		

Two-state query time (ms) ↓					
Method	Panda	PND	Go2	UR5e	Avg.
Part^{2}GS	<u>6.18</u>	**4.72**	<u>5.14</u>	<u>5.54</u>	<u>5.39</u>
REArtGS++	6.62	7.75	8.09	7.59	7.51
ROLA-GS	**5.00**	<u>5.89</u>	**4.99**	**4.60**	**5.12**

A.21 Broader Dynamic and Articulated Baselines

Dynamic methods use 80-frame 192^2 trajectories, while two-state methods use 40 192^2 views per endpoint. Per-robot reconstruction metrics are compared only within each protocol block (Table 22). These runs have valid foreground on every registered evaluation frame. Under 80-frame trajectory supervision, ROLA-GS leads the best dynamic method on mean FG-PSNR and silhouette IoU, with the best silhouette IoU on all four robots. Under two-state supervision, ROLA-GS has the best FG-PSNR on all four robots and improves mean silhouette IoU by 0.2059 over Part^{2}GS.

Table 22: Native-protocol comparisons with dynamic and two-state articulated Gaussian baselines. Rows are grouped by protocol.

Input	Method	Panda	PND	Go2	UR5e	Avg.
FG-PSNR (dB) ↑						
80 frames	Deformable 3DGS	14.080	11.312	<u>17.661</u>	<u>12.714</u>	<u>13.941</u>
	4D-GS	13.570	<u>11.689</u>	**18.649**	10.770	13.670
	ROLA-GS	**19.466**	**17.870**	17.295	**16.193**	**17.706**
2 states	Part^{2}GS	<u>11.499</u>	<u>10.318</u>	<u>12.899</u>	<u>7.805</u>	<u>10.630</u>
	REArtGS++	10.280	8.795	10.499	7.414	9.247
	ROLA-GS	**15.452**	**17.282**	**16.534**	**14.339**	**15.902**
Silhouette IoU ↑						
80 frames	Deformable 3DGS	0.5927	0.6696	0.8317	<u>0.6889</u>	<u>0.6957</u>
	4D-GS	<u>0.6165</u>	<u>0.6733</u>	<u>0.8528</u>	0.5500	0.6731
	ROLA-GS	**0.7971**	**0.9338**	**0.8582**	**0.9025**	**0.8729**
2 states	Part^{2}GS	**0.7648**	<u>0.7407</u>	<u>0.7407</u>	<u>0.3243</u>	<u>0.6426</u>
	REArtGS++	<u>0.7589</u>	0.6592	0.7226	0.3137	0.6136
	ROLA-GS	0.7291	**0.9417**	**0.8252**	**0.8978**	**0.8485**

A.22 Selective-Control Motion in Robot Context

These physical-mesh replays use the recorded final states, a shared camera, and native 1536^2 target and protected-link crops. Protected colors share one per-robot displacement scale, while Table 2 reports the aggregate target error and protected-link displacement.

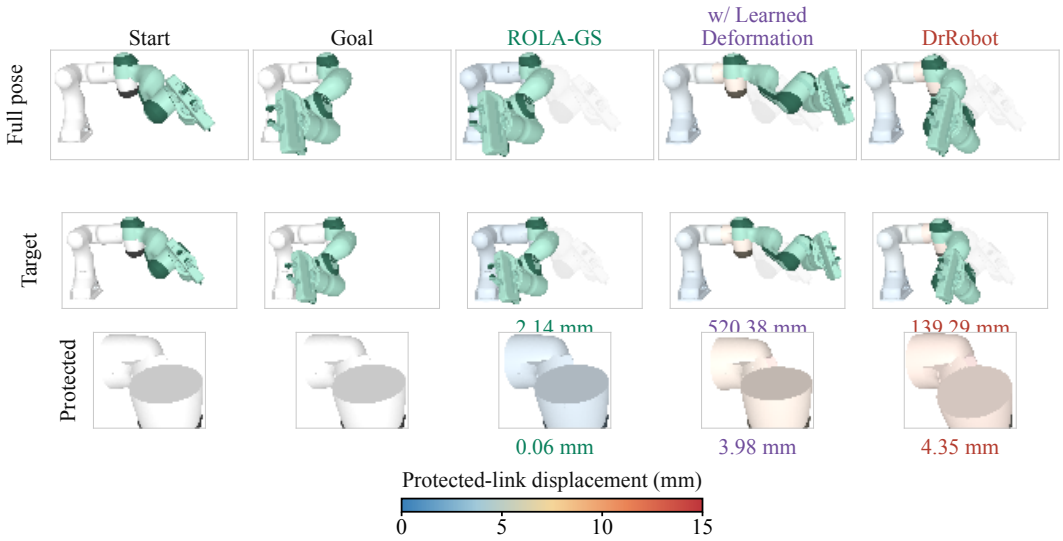


Figure 20: **Panda gripper control.** Rows show the full pose, target crop, and protected-link crop with target error and protected-link displacement.

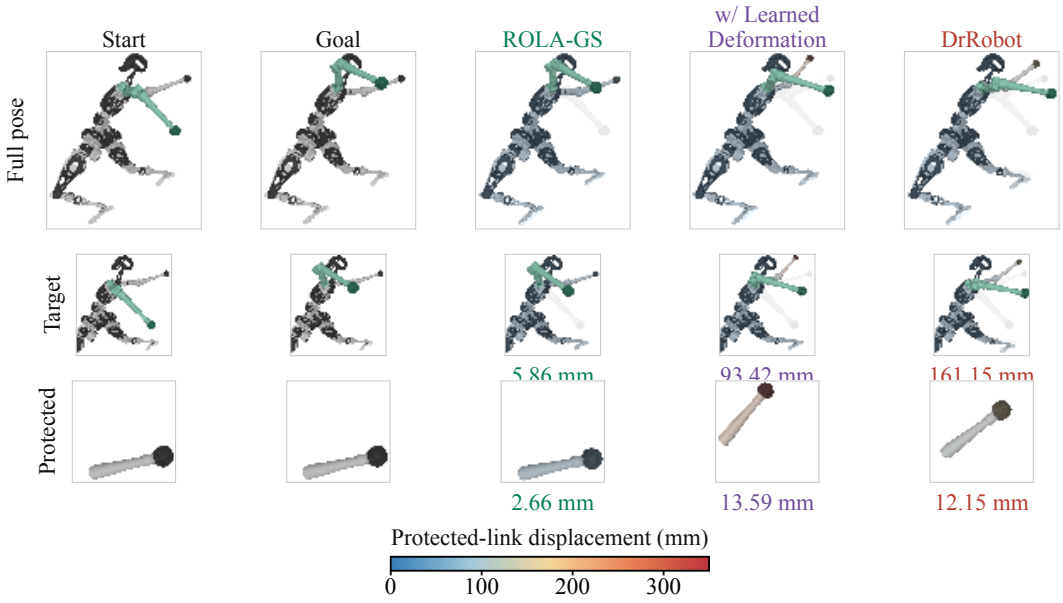


Figure 21: **PND left-wrist control.** Full poses and detail crops compare target motion and protected-link displacement.

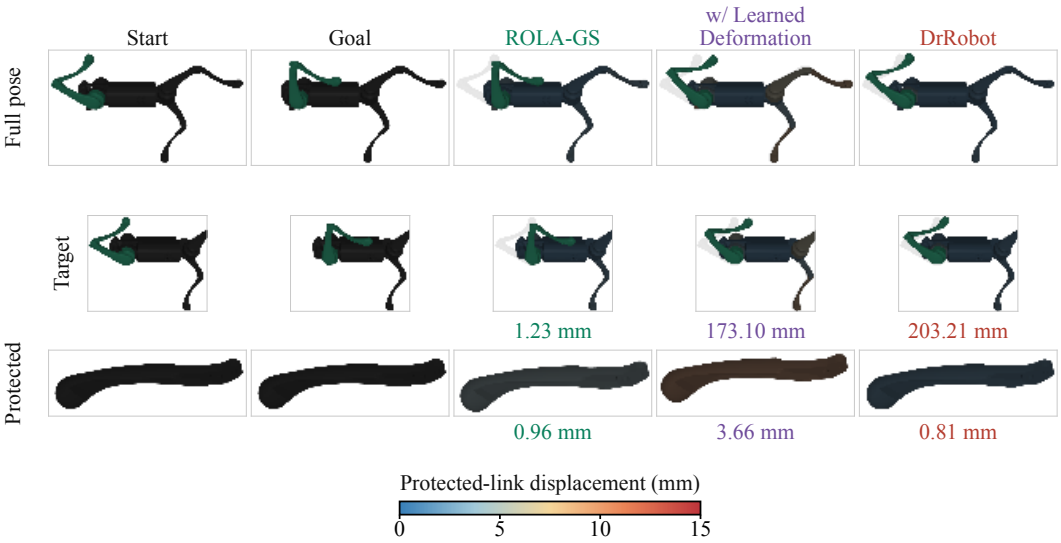


Figure 22: **Go1 front-left foot control.** Target and protected-link crops use a shared displacement scale.

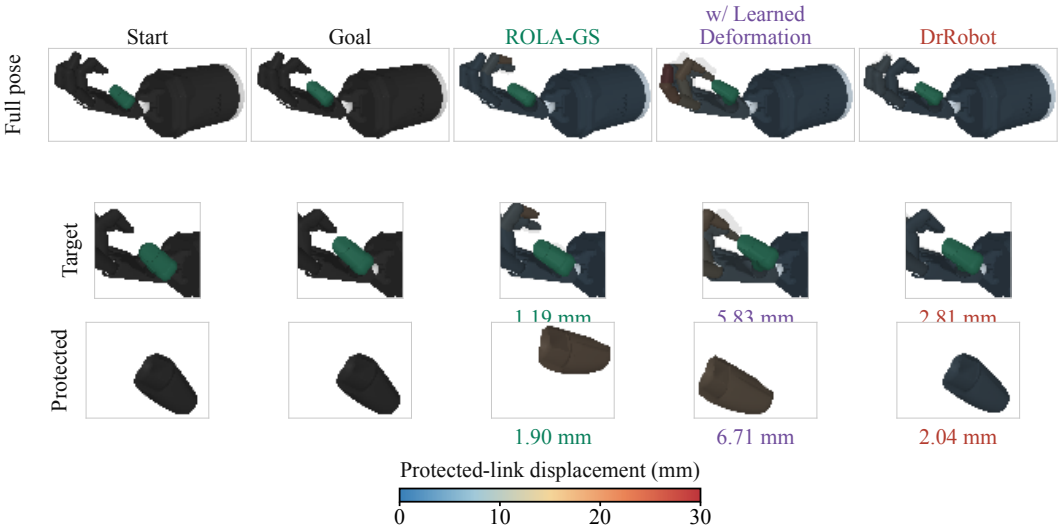


Figure 23: **Shadow Hand thumb control.** Rows show the full pose, target crop, and protected-link crop with target error and protected-link displacement.

B SUPPLEMENTARY METHOD DETAILS

This supplement gives the implementation conventions, expanded loss definitions, gradient derivation, complexity analysis, and fixed training constants for the final ROLA-GS model.

B.1 COORDINATE AND CAMERA CONVENTIONS

The robot model supplies $T_\ell(q)$, mapping link-local coordinates to the robot-base frame, through forward kinematics. In our experiments, it is instantiated from a MuJoCo XML scene. q_0 is the fixed reference joint configuration for each robot. The fixed calibrated camera model π_n remains fixed during training and supplies the rendering and projection parameters for observation n. For simulator scenes, q_0 uses zero for joints whose limits contain zero, the nearest valid limit for one-sided joints, and the midpoint of each finite Panda joint range.

The same q_0 is used to generate the initial Gaussian samples and to construct the reference inverse $T_\ell(q_0)^{-1}$, so the relative transform is the identity at $q = q_0$.

The Gaussian state is initialized from the geometric representation specified by the robot model at q_0. For Gaussian i, μ_i^0, Σ_i^0, α_i, and c_i denote its center and anisotropic covariance at q_0, opacity, and SH coefficients. Each Gaussian is assigned a link ℓ_i corresponding to its sampled surface point. This assignment selects the link transform applied to the Gaussian and remains fixed during optimization. The covariance uses the standard anisotropic 3DGS parameterization. The implementation represents it with standard log-scales s_i and a normalized quaternion r_i when evaluating the anchor regularizer. Each Gaussian also stores η_i, the learnable coefficients of the pose-dependent appearance correction defined below. The centers μ_i^0 and covariances Σ_i^0 are expressed in the robot-base frame at q_0. The initial Gaussian orientation is constructed from a surface frame whose third axis is aligned with the sampled surface normal, and the covariance uses larger tangential than normal scales. The corresponding initialization values are denoted by $\bar{\mu}_i, \bar{S}_i, \bar{r}_i, \bar{\alpha}_i, \bar{c}_i$. For a query joint configuration q, the link transform from the reference configuration q_0 to the query configuration q is given by:

$$T_\ell^{q \leftarrow q_0} = T_\ell(q) T_\ell(q_0)^{-1} = \begin{bmatrix} R_\ell^{q \leftarrow q_0} & t_\ell^{q \leftarrow q_0} \\ 0 & 1 \end{bmatrix}. \tag{14}$$

The query Gaussian representation and its geometric transform are given by:

$$\begin{aligned} \mu_i(q) &= R_{\ell_i}^{q \leftarrow q_0} \mu_i^0 + t_{\ell_i}^{q \leftarrow q_0}, \\ \Sigma_i(q) &= R_{\ell_i}^{q \leftarrow q_0} \Sigma_i^0 \left(R_{\ell_i}^{q \leftarrow q_0}\right)^\mathsf{T}. \end{aligned} \tag{15}$$

where $R_\ell^{q \leftarrow q_0} \in \mathrm{SO}(3)$ and $t_\ell^{q \leftarrow q_0} \in \mathbb{R}^3$ are the rotation and translation blocks of the link transform, and the superscript records its target and source configurations. Equation 15 changes only the geometric state (μ_i, Σ_i). The fixed link assignment ℓ_i selects the link transform applied to Gaussian i and is propagated whenever the Gaussian set is densified or pruned. The opacity and appearance parameters remain shared across poses. The SH coefficients are evaluated at the query viewing direction, while the pose-dependent appearance correction uses the Gaussian orientation induced by the query configuration q.

B.2 POSE-DEPENDENT APPEARANCE CORRECTION

The base view-dependent radiance $c_i^{\mathrm{SH}}(d)$ is evaluated from the learnable coefficients c_i at the current viewing direction d. The renderer direction d points from the camera to the Gaussian center, so it equals $-d_i(q)$ under the main-text point-to-camera convention. The two directions describe the same ray with opposite orientations. In the pose-dependent appearance correction, let $n_i(q)$ denote the unit third axis of the Gaussian orientation at q, and let $v_i(q, d) = -d$ denote the unit direction from the Gaussian toward the camera. We parameterize the residual response linearly in η_i using $b_i(q, d)$, whose terms are determined by $n_i(q)$ and d:

$$b_i(q, d) = \left[1, n_x, n_y, n_z, n_x n_y, n_y n_z, n_z n_x, n_x^2 - n_y^2, 3n_z^2 - 1, [n^\mathsf{T} v]_+^2, [n^\mathsf{T} v]_+^8\right]^\mathsf{T}. \tag{16}$$

Here n_x, n_y, n_z are the components of $n_i(q)$, $v = v_i(q, d)$, and $[x]_+ = \max(x, 0)$. The unit-vector dot product is clamped to $[0, 1]$. Base SH colors include the standard 3DGS activation, $c_i^{\mathrm{SH}}(d) =$

$[\mathrm{SH}(c_i, d) + 0.5]_+$. The first nine terms form the unnormalized real spherical-harmonic expansion through degree two on the Gaussian orientation induced by the query configuration q. The final two terms provide broad and concentrated normal-view appearance responses, motivated by the powered angular responses used in classical shading models (Phong, 1975, Blinn, 1977). The per-Gaussian appearance coefficient matrix $\eta_i \in \mathbb{R}^{11\times 3}$ produces the residual response $z_i(q, d)$ and final Gaussian color $c_i(q, d)$:

$$z_i(q,d) = \eta_i^{\mathsf{T}} b_i(q,d), \qquad c_i(q,d) = \left[c_i^{\mathrm{SH}}(d) + \beta \tanh(z_i(q,d))\right]_+ . \tag{17}$$

The appearance regularizer is

$$E_{\mathrm{res}} = \frac{1}{3BP} \sum_{i=1}^{P} \|\eta_i\|_F^2 . \tag{18}$$

B.3 Training Objective

Mapping to the main-text objective. Equations 7–9 describe the foreground photometric, silhouette, and anchor components. The implementation expressions below expand these components rather than repeat identical objectives: RGB error additionally averages the three color channels, silhouette supervision includes region and support terms, and the anchor includes rotation, opacity, SH, and scale-bound penalties. Here the component weights absorb the outer balancing coefficients in Eq. 6, so each coefficient is applied once. Table 23 lists the implementation weights.

Let N be the number of training observations, Ω the image domain, and B_n the binary mask of the tight bounding box enclosing foreground mask M_n. Let S_n be the supplied support mask. When the target foreground also supplies support, $S_n = M_n$. The weighted RGB loss is

$$E_{\mathrm{pho}} = \frac{1}{3N|\Omega|} \sum_{n=1}^{N} \sum_{p\in\Omega} w_n(p) \|\widehat{I}_n(p) - I_n(p)\|_1, \tag{19}$$

with

$$w_n(p) = 0.08 + 0.70 B_n(p) + 3.6 M_n(p) + 0.8\,\mathrm{sg}(\widehat{\alpha}_n(p)). \tag{20}$$

where sg stops the gradient through its argument. For a pixel subset $S \subseteq \Omega$, MSE_S denotes the mean squared error evaluated over S. The silhouette loss is

$$\begin{aligned} E_{\mathrm{sil}} = \frac{1}{N} \sum_{n=1}^{N} \big[& \lambda_{\mathrm{out}} \,\mathrm{MSE}_{B_n=0}(\widehat{\alpha}_n, 0) + \lambda_{\mathrm{mask}} \,\mathrm{MSE}(\widehat{\alpha}_n, M_n) \\ & + \lambda_{\mathrm{in}} \,\mathrm{MSE}_{S_n=1}(\widehat{\alpha}_n, 1) + \lambda_{\mathrm{sup,out}} \,\mathrm{MSE}_{B_n=1, S_n=0}(\widehat{\alpha}_n, 0) + \lambda_{\mathrm{Dice}} E_{\mathrm{Dice},n} \big], \end{aligned} \tag{21}$$

with

$$E_{\mathrm{Dice},n} = 1 - \frac{2\sum_p \widehat{\alpha}_n(p) S_n(p) + 1}{\sum_p \widehat{\alpha}_n(p) + \sum_p S_n(p) + 1}. \tag{22}$$

The three region-restricted MSE terms supplement the whole-image mask loss. Empty regions contribute zero. Without a support mask, the inside-support, outside-support, and Dice terms are zero. Foreground-normalized RGB losses and mask-boundary weighting are separate real-RGB fitting options, not the weighted RGB objective above.

The scale bound penalty used by the anchor regularizer is

$$E_{\mathrm{bound}} = \frac{1}{3P} \sum_{i=1}^{P} \left[\|\mathrm{ReLU}(\log S_{\min} - s_i)\|_2^2 + \|\mathrm{ReLU}(s_i - \log S_{\max})\|_2^2 \right], \tag{23}$$

where $S_{\min}$ and $S_{\max}$ are the lower and upper bounds on the activated Gaussian scales, and s_i is the corresponding log-scale.

The anchor regularizer is

$$\begin{aligned} E_{\mathrm{anc}} = & \frac{\lambda_\mu}{P} \sum_i \|\mu_i^0 - \bar{\mu}_i\|_2^2 + \lambda_s \,\mathrm{MSE}(s_i, \log \bar{S}_i) \\ & + \frac{\lambda_R}{P} \sum_i (1 - |\langle \mathrm{norm}(r_i), \mathrm{norm}(\bar{r}_i)\rangle|) + \lambda_\alpha \,\mathrm{MSE}(\alpha_i, \bar{\alpha}_i) \\ & + \lambda_{c_0} \,\mathrm{MSE}(c_{i,0}, \bar{c}_{i,0}) + \lambda_{c_{>0}} \,\mathrm{MSE}(c_{i,>0}, 0) + \lambda_{\mathrm{bound}} E_{\mathrm{bound}} + \lambda_{\mathrm{res}} E_{\mathrm{res}}. \end{aligned} \tag{24}$$

MSE averages all scalar entries. The degree-zero SH coefficients $c_{i,0}$ are anchored to initialization, while higher-order coefficients $c_{i,>0}$ are regularized toward zero. Geometry, opacity, SH, and scale-bound penalties are implementation terms beyond the center and log-scale anchor shown in Eq. 9. The appearance penalty averages all $3BP$ coefficient entries, with its weight applied once.

B.4 Training Stages and Density Control

We first optimize the 3D Gaussians for 2,000 iterations using observations at q_0. We then continue optimization for 120,000 iterations using observations at measured robot configurations, applying the corresponding relative link transforms before rendering each observation. Both stages optimize Gaussian centers, scales, rotations, opacity, SH coefficients, and appearance correction coefficients. During adaptive density control, Gaussians with large gradients of their projected centers are duplicated or split, and Gaussians with low opacity are pruned. Each new Gaussian inherits the link assignment of its parent, and pruning removes the corresponding link assignments.

Table 23: Fixed constants for the final full-Gaussian training protocol.

Parameter	Value	Parameter	Value
Canonical iterations	2,000	Posed iterations	120,000
Initial Gaussians	52,000	Maximum Gaussians	120,000
SH degree	3	Appearance bound β	0.28
RGB bbox weight	0.70	RGB foreground weight	3.60
Alpha outside weight	0.35	Alpha mask weight	1.05
Support inside weight	0.15	Support outside weight	4.00
Dice weight	0.35	Appearance regularization weight	10^{-4}
Mean prior weight	0.12	Scale prior weight	0.035
Quaternion prior weight	0.025	Opacity prior weight	0.004
SH prior weight	5×10^{-4}	Scale bound weight	0.25

B.5 Differentiable State Audit

For observation n, let $R_{c,n}$ and $t_{c,n}$ denote the fixed world-to-camera rotation and translation, and let π_n be the calibrated perspective projection. The image plane Gaussian state is

$$\begin{aligned} \mu^{\mathrm{I}}_{i,n} &= \pi_n(R_{c,n}\mu_i(q_n) + t_{c,n}), \\ \Sigma^{\mathrm{I}}_{i,n} &= J_{i,n}R_{c,n}\Sigma_i(q_n)R_{c,n}^{\mathsf{T}}J_{i,n}^{\mathsf{T}}, \end{aligned} \tag{25}$$

where $J_{i,n}$ is the Jacobian of π_n at the camera space Gaussian center. The rendering objective receives the derivatives of this image plane state through the differentiable rasterizer. The anchor regularizer supplies the corresponding derivatives in the canonical parameterization.

For each scalar joint coordinate q_j, let $v_{ij}(q)$ and $\omega_{ij}(q)$ denote the linear and angular velocities of Gaussian i induced by a unit change in q_j. The state derivatives are

$$\frac{\partial \mu_i(q)}{\partial q_j} = v_{ij}(q), \qquad \frac{\partial \Sigma_i(q)}{\partial q_j} = [\omega_{ij}(q)]_\times \Sigma_i(q) - \Sigma_i(q)\,[\omega_{ij}(q)]_\times . \tag{26}$$

For a revolute joint on the kinematic path to ℓ_i, $v_{ij}(q) = \omega_j(q) \times (\mu_i(q) - o_j(q))$ and $\omega_{ij}(q) = \omega_j(q)$. For a prismatic joint on that path, $v_{ij}(q) = \nu_j(q)$ and $\omega_{ij}(q) = 0$. Both quantities are zero when the joint does not affect ℓ_i. For the appearance correction, let d_{in} denote the viewing direction for observation n, $z_{in} = z_i(q_n, d_{in})$, and $b_{in} = b_i(q_n, d_{in})$. Let m_{in} denote the elementwise indicator of $c_i^{\mathrm{SH}}(d_{in}) + \beta \tanh(z_{in}) > 0$. The color derivative is

$$\frac{\partial c_i(q_n, d_{in})}{\partial q_j} = m_{in} \odot \left[\frac{\partial c_i^{\mathrm{SH}}}{\partial d_{in}} \frac{\partial d_{in}}{\partial q_j} + \beta \left(1 - \tanh^2(z_{in})\right) \odot \eta_i^{\mathsf{T}} \frac{\partial b_{in}}{\partial q_j} \right]. \tag{27}$$

Because η_i is shared across observations, its photometric gradient accumulates over the observed configurations:

$$\nabla_{\eta_i}(E_{\mathrm{pho}} + E_{\mathrm{anc}}) = \sum_{n=1}^{N} b_i(q_n, d_{in}) \left[\nabla_{z_i(q_n, d_{in})} E_{\mathrm{pho}}\right]^{\mathsf{T}} + \nabla_{\eta_i} E_{\mathrm{anc}}. \tag{28}$$

The gradient with respect to z_i includes the appearance activation and the differentiable rasterization path. The measured configurations q_n remain fixed during training, so the derivatives with respect to q characterize query sensitivity and state auditing rather than an additional optimization variable. If $\ell_i \notin \mathrm{Desc}(j)$, the mean, covariance, and color derivatives are all zero. For gradient queries, the implementation evaluates these dependencies by automatic differentiation through exact relative FK, appearance evaluation, and Gaussian rasterization.

B.6 Computational Complexity

For a query joint configuration, the operation count is

$$\mathcal{C}_{\mathrm{query}} = O(|\mathcal{L}| + P + \Gamma)\,, \qquad \Gamma = \sum_{i=1}^{P} t_i, \tag{29}$$

where t_i is the projected tile footprint of Gaussian i. The three terms correspond to link-level FK, one rigid transform per Gaussian, and projected Gaussian rasterization. For T training steps with state size P_t,

$$\mathcal{C}_{\mathrm{train}} = O\left(\sum_{t=1}^{T} (|\mathcal{L}| + P_t + \Gamma_t)\right). \tag{30}$$

Pose-conditioned deformation-network evaluation does not appear in either count.