

1 A Preliminaries

2 The representation of 3D scenes with Gaussian splatting employs a collection of colored 3D Gaussian
 3 primitives. Each Gaussian primitive G is characterized by its center μ in 3D space and a corresponding
 4 3D covariance matrix Σ , conforming to:

$$G(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^\top \Sigma^{-1}(\mathbf{x} - \mu)\right). \quad (1)$$

5 To facilitate optimization, we decompose the covariance matrix Σ into $\mathbf{R}\mathbf{S}\mathbf{S}^\top\mathbf{R}^\top$, where \mathbf{R} rep-
 6 represents a rotation matrix encoded by a quaternion $\mathbf{q} \in SO(3)$, and \mathbf{S} denotes a scaling matrix
 7 parameterized by a 3D vector \mathbf{s} . The complete parameterization of each Gaussian includes an opacity
 8 value σ governing its rendering influence, alongside spherical harmonic (SH) coefficients \mathbf{sh} that
 9 capture view-dependent appearance variations.

10 The scene representation therefore consists of a set $\mathcal{G} = \{G_j : \mu_j, \mathbf{q}_j, \mathbf{s}_j, \sigma_j, \mathbf{sh}_j\}$, where μ_j is the
 11 position, \mathbf{q}_j is the orientation, \mathbf{s}_j is the scale, σ_j is the standard deviation, and \mathbf{sh}_j is the spherical
 12 harmonics coefficients. The rendering pipeline projects these 3D Gaussians onto the image plane,
 13 where they undergo efficient α -blending. During projection, the 2D covariance matrix Σ' and center
 14 μ' are computed as:

$$\Sigma' = \mathbf{J}\mathbf{W}\Sigma\mathbf{W}^\top\mathbf{J}^\top, \quad \mu' = \mathbf{J}\mathbf{W}\mu, \quad (2)$$

15 where \mathbf{J} is the Jacobian matrix of the linear approximation of the projective transformation and \mathbf{W} is
 16 the rotation matrix of the viewpoint. The final color $C(\mathbf{u})$ at pixel \mathbf{u} emerges from neural point-based
 17 α -blending:

$$C(\mathbf{u}) = \sum_{i \in \mathcal{N}} T_i \alpha_i \text{SH}(\mathbf{sh}_i, \mathbf{v}_i), \quad \text{where } T_i = \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (3)$$

18 Here, \mathcal{N} shows the number of Gaussians that overlap with the pixel \mathbf{u} . In this formulation, $\text{SH}(\cdot, \cdot)$
 19 represents the spherical harmonic function evaluated with respect to the view-direction \mathbf{v}_i . The
 20 α -value for each Gaussian is determined by:

$$\alpha_i = \sigma_i \exp\left(-\frac{1}{2}(\mathbf{p} - \mu'_i)^\top \Sigma'^{-1}_i (\mathbf{p} - \mu'_i)\right), \quad (4)$$

21 where μ'_i and Σ'_i correspond to the projected center and covariance matrix of Gaussian G_i . Real-
 22 time and high-fidelity image synthesis is achieved through the optimization of Gaussian parameters
 23 $\{G_j : \mu_j, \mathbf{q}_j, \mathbf{s}_j, \sigma_j, \mathbf{sh}_j\}$ coupled with adaptive density adjustment. We propose a framework that
 24 builds on the foundation of Gaussian Splatting for dynamic scenes by adding sparse control, without
 25 compromising computational efficiency and rendering quality.

26 B Differentiable Dense Bundle Adjustment (DBA) layer

27 We further refine the camera poses through the Differentiable Dense Bundle Adjustment (DBA)
 28 layer [5], which incorporates optical flow information to improve geometry estimation. This approach
 29 is particularly effective in dynamic scenes as it allows us to focus optimization on static regions while
 30 accounting for motion consistency in dynamic areas:

$$E_{DBA}(C'_t, d'_t) = \sum_{(i,j) \in \mathcal{E}} (1 - M_t(i)) \|p_{ij}^* - \Pi_e(C'_{ij} \circ \Pi_e^{-1}(p_i, d'_i))\|_{\Sigma_{ij}}^2, \quad (5)$$

31 where C'_t is the camera pose and d'_t is the depth values. $\|\cdot\|_{\Sigma_{ij}}$ is the Mahalanobis distance weighted
 32 by the confidence scores, $(i, j) \in \mathcal{E}$ denotes an overlapping field-of-view with shared points between
 33 image I_i and I_j , p_{ij}^* is the sum of optical flow r_{ij} and p_{ij} , and the term $(1 - M_t(i))$ ensures that only
 34 static regions contribute to the optimization.

35 The system optimizes for updated camera pose C'_t and depth values d'_t through a sparse matrix
 36 formulation $\Delta\xi_t$ and Δd_t , which is the normal equation derived from the cost function in Eqn.(5):

$$\begin{bmatrix} B & E \\ E^\top & C \end{bmatrix} \begin{bmatrix} \Delta\xi_t \\ \Delta d_t \end{bmatrix} = \begin{bmatrix} v \\ w \end{bmatrix} \quad (6)$$

Two-Stage Optimization Process

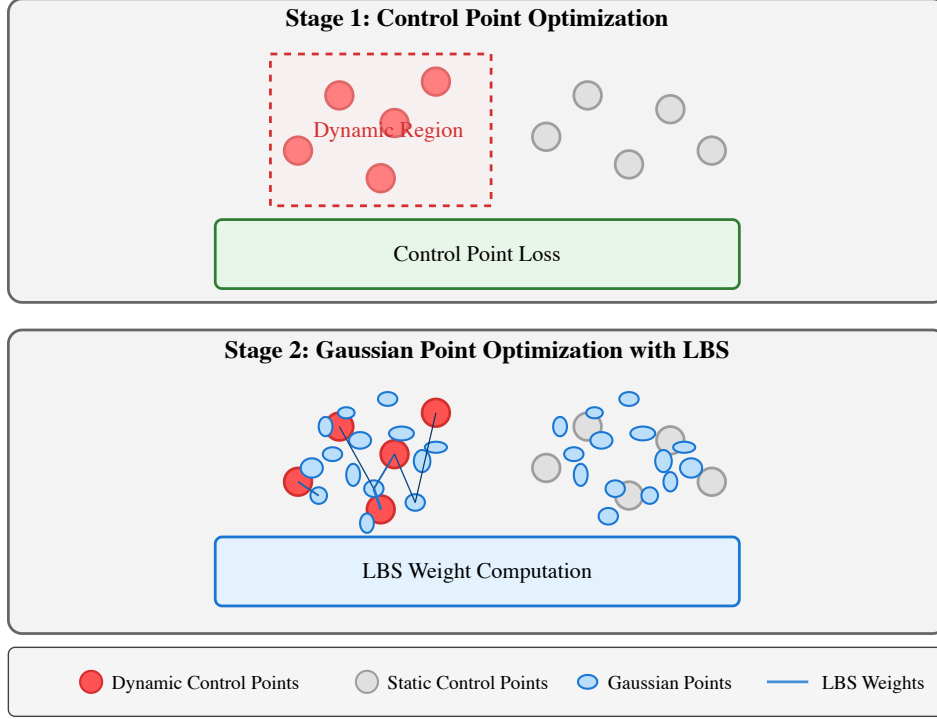


Figure 1: Two-Stage Optimization Process for Motion-Aware Gaussian Splatting. Stage 1 optimizes control points in dynamic regions (red) using control point loss, while static regions (gray) remain fixed. Stage 2 performs Gaussian optimization (blue ellipses) through Linear Blend Skinning, with connection lines showing influence weights between control points and Gaussians.

where E models the coupling between pose and depth parameters, C captures the depth-depth relationships, B represents the pose-pose interactions, v and w are the gradient terms corresponding to pose updates and depth updates, respectively.

C Two-stage Optimization Strategy in MAGS

Our two-stage optimization strategy reduces computational cost by focusing on dynamic regions only, as described in Sec. 3.5 of the main paper. We illustrate the strategy in Fig 1, which demonstrates our approach to efficient motion-aware scene reconstruction. In the first stage, we selectively optimize control points only within regions identified as dynamic by our motion segmentation module, significantly reducing the computational burden compared to methods that optimize all scene parameters simultaneously. The dynamic region boundary (indicated by the dashed red outline) separates areas requiring deformation modeling from static background elements. During the second stage, Gaussian primitives are optimized using Linear Blend Skinning weights computed through normalized exponential kernels based on spatial proximity to control points. The varying opacity of connection lines visualizes the influence magnitude of each control point on nearby Gaussian primitives, with stronger connections (darker lines) indicating higher LBS weights. This strategic separation of optimization stages enables our method to achieve both computational efficiency and high-quality reconstruction by concentrating computational resources where motion occurs while maintaining stable anchoring in static regions.

Table 1: Quantitative evaluation of camera poses estimation on the MPI Sintel dataset. The **best** and the **second best** results are denoted by pink and yellow. The methods of the top block discard the dynamic components and do not reconstruct the dynamic scenes; thus they cannot render novel views. We exclude the COLMAP results since it fails to produce poses in 5 out of 14 sequences.

Method	ATE↓	RPE trans↓	RPE rot↓
DROID-SLAM* [5]	0.175	0.084	1.912
DPVO* [6]	0.115	0.072	1.975
ParticleSFM [11]	0.129	0.031	0.535
LEAP-VO* [1]	0.089	0.066	1.250
Robust-CVD [2]	0.360	0.154	3.443
CasualSAM [10]	0.141	0.035	0.615
DUSt3R [7] w/ mask	0.417	0.250	5.796
MonST3R [9]	0.108	0.042	0.732
NeRF- [8]	0.433	0.220	3.088
BARF [3]	0.447	0.203	6.353
RoDynRF [4]	0.089	0.073	1.313
Ours	0.086	0.035	0.639

* requires ground truth camera intrinsics as input

55 D LBS parameter learning

56 As shown in Eqn. (8) of the main paper, the LBS weights are computed with a normalized exponential
57 kernel. Our motion-aware framework significantly improves the efficiency and stability of LBS
58 parameter learning through three key mechanisms:

- 59 1) Focusing control point optimization exclusively on dynamic regions identified by our motion mask,
60 which concentrates computational resources where they’re most needed.
- 61 2) Preserving static points’ positions during the GS optimization process, which provides stable
62 anchors for the scene representation.
- 63 3) Substantially reducing the number of points requiring LBS parameter learning, which improves
64 both computational efficiency and optimization stability.

65 E Results on Pose Estimation

66 Tab 1 presents our pose estimation results on the MPI Sintel dataset, where our method demonstrates
67 exceptional performance across all metrics. The methods in the upper portion of the table discard
68 dynamic components and cannot render novel views. COLMAP results are excluded because it fails
69 to produce poses in 5 out of 14 sequences, which further illustrates the challenges faced by COLMAP-
70 dependent methods in handling general scenes with complex camera motions. We achieved an ATE
71 of 0.086, showing comparable accuracy to the SOTA method LEAP-VO (0.089) and significantly
72 outperforming methods like BARF (0.447) and NeRF- (0.433). For relative pose errors, our method
73 achieves 0.035 for translation and 0.639 for rotation, matching or exceeding the performance of
74 specialized pose estimation methods.

75 The strong pose estimation performance can be attributed to several factors. First, our MA-BA
76 effectively leverages the dynamic masks refined by our mask refinement pipeline, reducing the
77 noise introduced by dynamic objects during pose optimization. Second, integrating SAM2 for mask
78 refinement significantly improves the accuracy of dynamic object segmentation, leading to more
79 reliable static point selection for RANSAC-based pose estimation.

80 F Technical Discussion and Analysis

81 F.1 Technical Contributions

82 Our motion-aware framework delivers three key innovations: (1) an integrated MA-BA module
83 that combines transformer-based motion priors with SAM2’s segmentation capabilities in a unified

84 pipeline; (2) a gradient-detached dynamic control point mechanism that strategically allocates
85 computational resources to motion-significant regions; and (3) an end-to-end pose-free approach
86 that eliminates dependency on pre-computed camera poses. Empirical validation confirms these
87 innovations deliver substantial improvements, with a 1.3dB PSNR gain over combined baseline
88 components.

89 **F.2 Scene Coordinate Regression Advantages**

90 Our approach leverages Scene Coordinate Regression (SCR) over Structure from Motion (SfM) for its
91 dual benefits: SCR not only delivers faster and more accurate results but also generates high-quality
92 dynamic masks through our motion-aware bundle adjustment module. This creates a synergistic
93 pipeline where dynamic region identification directly informs reconstruction, enabling more efficient
94 processing while maintaining or exceeding the accuracy of traditional SfM approaches.

95 **F.3 Two-Stage Optimization Strategy**

96 Our framework employs a two-stage strategy that separates motion estimation from reconstruction.
97 The first stage establishes coherent motion estimates (16-19 PSNR) without optimizing Gaussian
98 points directly. Control points define the deformation field but don't serve as Gaussian centers.
99 Gaussians are initialized independently during the second stage, guided by the established motion
100 field, enabling more stable convergence and higher-quality results.

101 **F.4 Control Point Efficiency**

102 Our control point approach delivers two significant advantages: (1) a 5 \times improvement in training
103 time by focusing optimization efforts only on dynamic regions; and (2) a remarkably compact
104 representation requiring only 80MB of storage compared to 153-200MB for competing methods.
105 These efficiency gains enable handling of challenging scenes with large motion magnitudes and
106 longer sequences, as demonstrated by superior performance on the DyNeRF dataset.

107 **F.5 Deformation Modeling**

108 Our approach employs As-Rigid-As-Possible (ARAP) regularization as a flexible prior for both rigid
109 and non-rigid deformations. For soft-body objects, this acts as a smoothness constraint rather than
110 enforcing strict rigidity. The system adaptively allocates more control points to highly deformable
111 regions, creating a finer deformation grid where needed. This approach successfully handles soft
112 objects, as demonstrated in the "peel-banana" sequence.

113 **F.6 Implementation Parameters**

114 Our implementation uses 512 control points by default, initialized uniformly with higher density in
115 dynamic regions. Performance remains robust across a range of control point quantities (100-1000),
116 reflecting the effectiveness of our adaptive allocation strategy that focuses computational resources
117 based on motion significance rather than a fixed distribution.

118 **References**

- 119 [1] Weirong Chen, Le Chen, Rui Wang, and Marc Pollefeys. Leap-vo: Long-term effective any point tracking
120 for visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
121 Recognition*, pages 19844–19853, 2024.
- 122 [2] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Computer
123 Vision and Pattern Recognition*, 2021.
- 124 [3] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural
125 radiance fields. In *ICCV*, 2021.
- 126 [4] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang,
127 Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF
128 Conference on Computer Vision and Pattern Recognition*, pages 13–23, 2023.

- 129 [5] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras.
130 *Advances in neural information processing systems*, 34:16558–16569, 2021.
- 131 [6] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information*
132 *Processing Systems*, 36, 2024.
- 133 [7] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric
134 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
135 *Recognition*, pages 20697–20709, 2024.
- 136 [8] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin.
137 inerf: Inverting neural radiance fields for pose estimation. In *IROS*, 2021.
- 138 [9] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun,
139 and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion.
140 *arXiv preprint arXiv:2410.03825*, 2024.
- 141 [10] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman.
142 Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37.
143 Springer, 2022.
- 144 [11] Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense
145 point trajectories for localizing moving cameras in the wild. In *European Conference on Computer Vision*,
146 pages 523–542. Springer, 2022.