SpeeDe3DGS: Speedy Deformable 3D Gaussian Splatting with Temporal Pruning and Motion Grouping

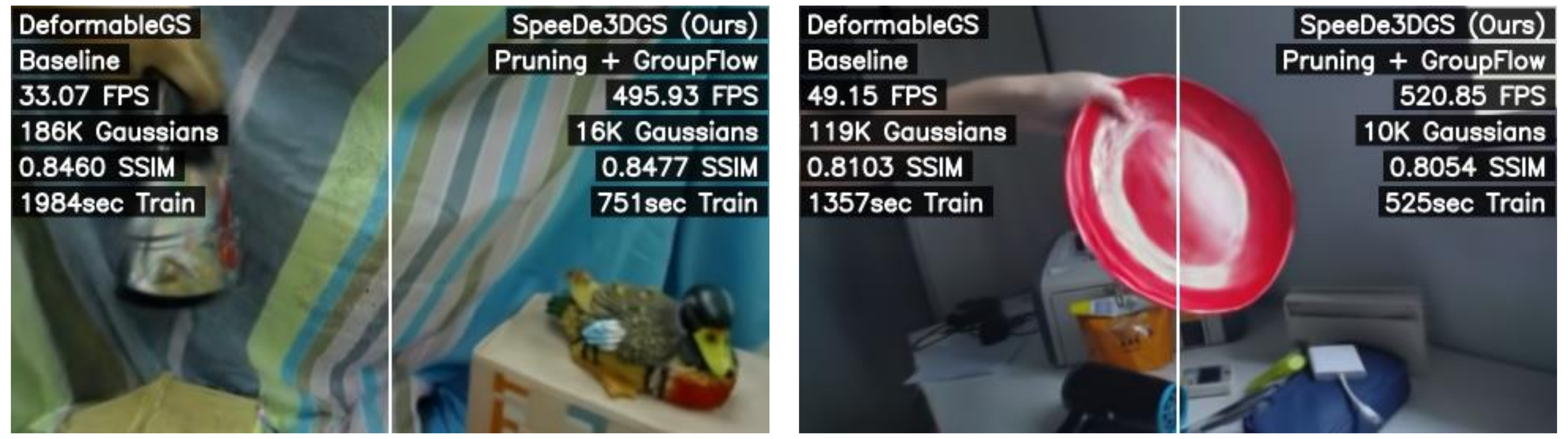
Allen Tu*, Haiyang Ying*, Alex Hanson, Yonghan Lee, Tom Goldstein, Matthias Zwicker

speede3dgs.github.io

CVPR JUNE 3-7, 2026 DENVER COLORADO

Background and Motivation

How can dynamic Gaussian Splatting deliver the high fidelity of neural motion while rendering 100+ FPS faster than non-neural motion models?



Dynamic Gaussian Splatting extends the set of 3D Gaussians $\mathcal{G}$ with a deformation network $\mathcal{D}$ that predicts time-varying offsets for the means μ, rotations r, and scales s at each timestep t:

$$(\mu + \Delta\mu_t,\ r + \Delta r_t,\ s + \Delta s_t) = \mathcal{D}(\mu, r, s, t).$$

$\mathcal{G}$ and $\mathcal{D}$ are jointly optimized over training frames $\mathcal{P}_{gt}$, where $I_{\mathcal{G}_t}(\phi)$ is the rendered image at pose ϕ and timestep t. g_i is the projected contribution of Gaussian $\mathcal{G}_i$ in $I_{\mathcal{G}_t}(\phi)$. Motion models for dynamic Gaussian Splatting methods fall in two categories:

- **Neural motion** is slow but produces high visual fidelity.
- **Non-neural motion** is fast but produces lower visual fidelity.

Acknowledgements

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 140D0423C0076. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Commercial support was provided by the Amazon Research Awards program and Open Philanthropy. Further support was provided by DARPA TIAMAT and the NSF TRAILS Institute (2229885).

Method

Temporal Sensitivity Pruning

We compute a **Temporal Sensitivity Pruning (TSP)** score $\tilde{U}_{\mathcal{G}_i}$ for each Gaussian $\mathcal{G}_i$ as the second-order sensitivity of the L_2 reconstruction error to its projected value:

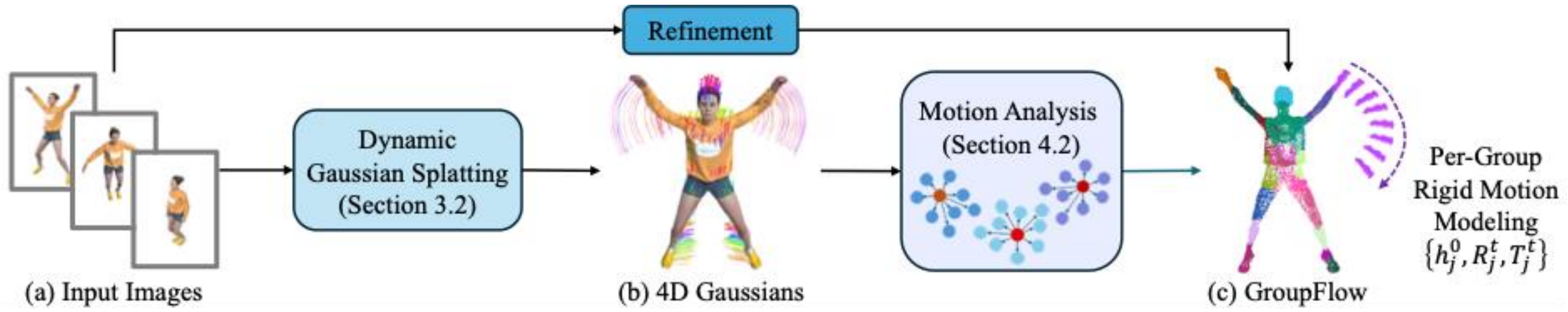
$$\tilde{U}_{\mathcal{G}_i} \approx \nabla^2_{g_i} L_2 \approx \sum_{\phi, t \in \mathcal{P}_{gt}} \left(\nabla_{g_i} I_{\mathcal{G}_t}(\phi)\right)^2.$$

Temporal Sensitivity Sampling (TSS) adds an annealing temporal perturbation to the timestamp input of the deformation network during TSP score computation, **revealing unstable primitives and suppressing floaters during pruning:**

$$(\mu + \Delta\mu, r + \Delta r, s + \Delta s) = \mathcal{D}(\mu, r, s, t + \mathcal{N}(0,1)\beta\Delta t(1 - i/\tau)).$$

We use our TSP + TSS score to prune low-sensitivity Gaussians during training, **reducing the total number of Gaussians by over 90% while preserving fidelity.**

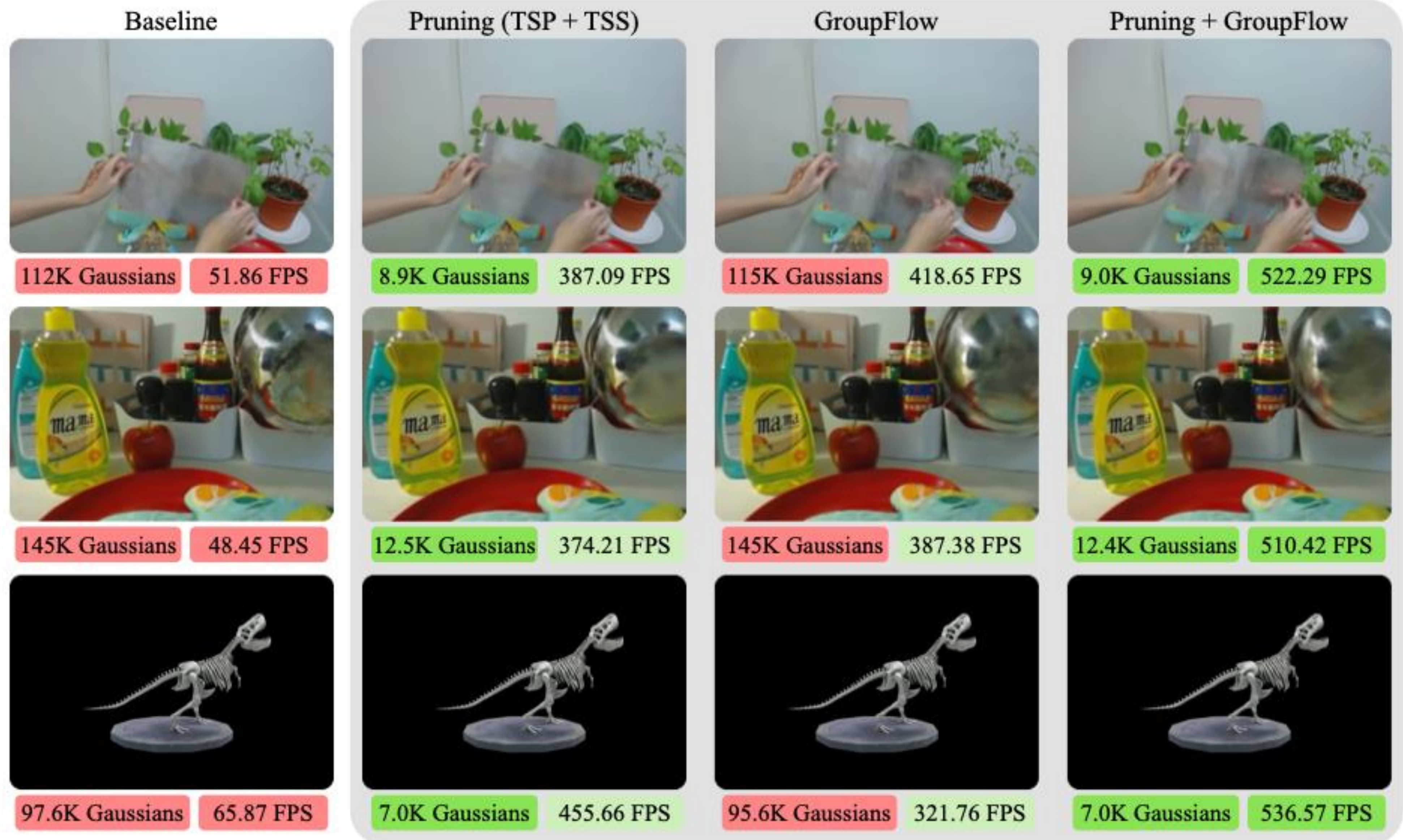
GroupFlow



Our GroupFlow method distills the high-fidelity per-Gaussian neural deformations into a smaller set of efficient grouped SE(3) transforms.

1. We start with a dense dynamic Gaussian Splatting model, where each Gaussian $\mathcal{G}_i$ is represented as a sequence of mean positions $\mathcal{M}_i = \{\mu_i^t\}_{t=0}^{F-1}$.
2. Then, we initialize J control trajectories $\{h_j^t\}$ and assign each mean μ_i to the most similar control point h_j using trajectory similarity score:
$$S_{i,j} = \lambda_r \operatorname{std}_t(\|\mu_i^t - h_j^t\|) + (1 - \lambda_r) \operatorname{mean}_t(\|\mu_i^t - h_j^t\|).$$
3. We fit a group-wise SE(3) flow to each group $\mathcal{M}^j$ using Umeyama alignment. The rotation R_j^t and translation T_j^t applied to $\mu_i \in \mathcal{M}^j$ at timestep t are:
$$\mu_i^t = R_j^t(\mu_i^0 - h_j^0) + h_j^0 + T_j^t.$$
4. The shared flow $\{h_j^0, R_j^t, T_j^t\}$ is optimized jointly with the scene, **reducing the number of transforms per frame from N (per-Gaussian) to J (per-group).**

Results



When compared to DeformableGS, **SpeeDe3DGS** achieves **11× FPS and 59% faster training on NeRF-DS** and **29× FPS and 73% faster training on HyperNeRF.**

Method	PSNR ↑	SSIM ↑	MS-SSIM ↑	LPIPS ↓	FPS ↑	Train Time (s) ↓
EffGS [20]	21.84	0.672	0.725	0.347	177.21	3757.81
STG-decoder [24]	21.81	0.678	0.742	0.352	109.42	5980.64
STG [24]	19.51	0.583	0.643	0.475	181.70	5359.56
RTGS [52]	21.61	0.663	0.720	0.350	143.37	7352.52
4DGS [47]	23.55	0.708	0.765	0.277	62.99 (1.00×)	8628.89 (1.00×)
+ Pruning (Ours)	22.44	0.689	0.737	0.334	**179.64 (2.85×)**	**4358.17* (1.47×)**
+ GroupFlow (Ours)	21.00	0.667	0.705	0.380	**290.21 (4.61×)**	**4176.49* (2.07×)**
DeformableGS [51]	24.07	0.694	0.755	0.283	20.20 (1.00×)	6227.43 (1.00×)
+ Pruning (Ours)	23.86	0.694	0.749	0.295	**137.01 (6.78×)**	**2850.60* (2.18×)**
+ GroupFlow (Ours)	23.52	**0.709**	**0.771**	0.313	**276.91 (13.71×)**	**2461.14* (2.53×)**

On the 50 scenes in MonoDyGauBench, **SpeeDe3DGS** achieves **100+ FPS higher speed and better visual fidelity** than all non-neural motion baselines.