A Pose-Free Approach for 4D Gaussian Splatting to Reconstruct Dynamic Scenes

Huosen OU and Yiding JI

***Abstract*— This work develops PF-4DGS, an novel framework for 4D Gaussian splatting that addresses the challenges associated with the reliance on accurate prior knowledge of camera poses in dynamic scene modeling. Our approach employs a pose-free optimization strategy that simultaneously estimates camera parameters and reconstructs the scene within a unified framework. We introduce a stable initialization technique and an efficient joint optimization loop that simultaneously improves scene reconstruction and camera tracking. Comprehensive evaluations on real-world datasets demonstrate that PF-4DGS achieves accuracy comparable to leading methods, even without prior camera pose information. This advancement marks a huge breakthrough in Gaussian splatting and promotes the application of this technique in dynamic environments.**

***Index Terms*— 4D Gaussian Splatting, pose-free reconstruction, joint optimization, dynamic scenes, computer vision**

I. Introduction

Novel view synthesis (NVS) represents a core challenge in computer vision and graphics, emphasizing the creation of photorealistic novel views of a scene from arbitrary, previously unobserved viewpoints. Recent advancements in scene representation [1], [2] and rendering techniques have substantially enhanced the quality of NVS. The emergence of Neural Radiance Fields (NeRF) [3] represented a transformative landmark in the field, and it revolutionized NVS by representing scenes as continuous neural volumetric functions. The obvious strength of NeRF lies in its capacity to implicitly encode scene properties through a neural network that predicts color and density for any 3D location and viewing angle. However, the computational demands of NeRF and its extensions, combined with the necessity for precise camera pose information, pose significant challenges for real-time applications and dynamic scene reconstruction.

To mitigate the computational inefficiency inherent in NeRF, recent research has explored alternative scene representations. One promising approach is 3D Gaussian Splatting (3DGS) [5], which substitutes the dense neural network with spatially-distributed 3D Gaussian primitives. This representation models the scene as a series of continuous ellipsoidal primitives, each parameterized by its position, orientation, size, and attributes such as color and opacity. 3DGS offers notable improvements in speed and scalability, also facilitates real-time rendering and reconstruction. Despite those merits, current 3DGS methods are predominantly tailored for static scenes and require accurate prior camera poses for initialization and optimization.

Fig. 1. Our PF-4DGS framework achieves real-time performance, rendering dynamic scenes at 42 frames per second (1352 × 1014 resolution) on the Neu3D benchmark [4], while maintaining high visual fidelity.

The extension of scene representations to dynamic environments has attracted substantial research interest. Neural Radiance Fields have been adapted for dynamic scenes with techniques such as D-NeRF [6] and NSFF [7], which incorporate temporal components into the volumetric representation. Meanwhile, 3D Gaussian Splatting has evolved to process dynamic scenes through the introduction of a temporal dimension, leading to 4D Gaussian Splatting (4DGS) [8]. This approach models dynamic scenes as collections of Gaussians that evolve in both space and time, allowing high-fidelity reconstruction and rendering of spatiotemporal phenomena. Although 4DGS demonstrates great potential, its reliance on accurate camera calibration restricts the deployment in uncontrolled environments where obtaining reliable pose information remains challenging.

In this work, we introduce a creative approach for pose-free 4D Gaussian Splatting to bridge the above mentioned research gaps. In contrast with existing approaches, our method eliminates the dependency on predefined camera poses through the simultaneous optimizing the scene representation and camera parameters. Specifically, we integrate a robust initialization strategy and an efficient optimization

Huosen OU and Yiding Ji are both with Robotics and Autonomous Systems Thrust, Systems Hub,The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. (Email addresses: hou061@connect.hkust-gz.edu.cn, jiyiding@hkust-gz.edu.cn.)

This work is supported by the National Natural Science Foundation of China grants 62303389 and 62373289; Guangdong Basic and Applied Basic Research Funding grants 2022A151511076 and 2024A1515012586; Guangdong Scientific Research Platform and Project Scheme grant 2024KTSCX039; Guangzhou Basic and Applied Basic Research Scheme grants 2023A04J1067; Guangzhou-HKUST(GZ) Joint Funding Program grants 2023A03J0678, 2023A03J0011, 2024A03J0618 and 2024A03J0680.

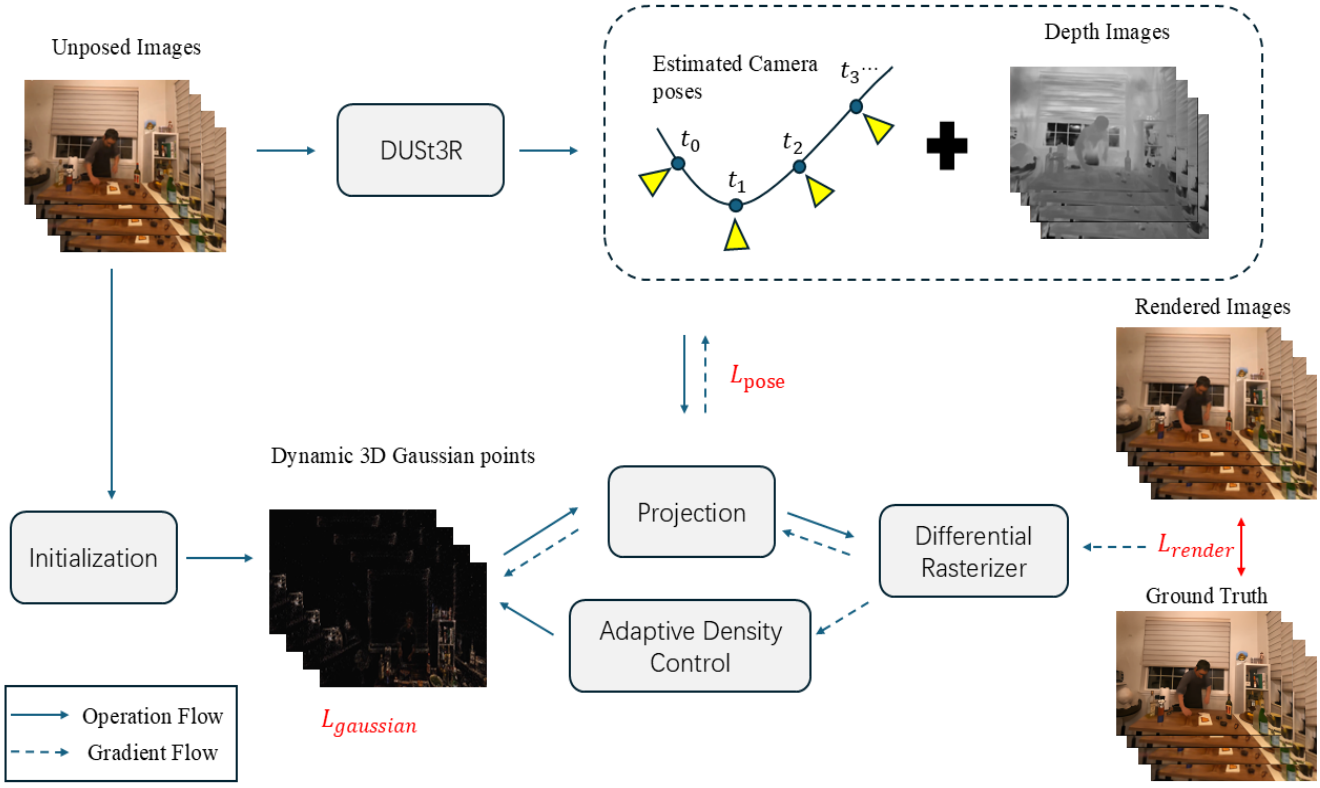

Fig. 2. The framework of PF-4DGS approach.

scheme that refines both the 4D Gaussian representation and the camera poses iteratively. This joint optimization framework effectively manages unconstrained dynamic scenes, achieving high-quality novel view synthesis without necessitating pose information. Overall, PF-4DGS shows a significant step forward for 4D scene reconstruction, removing the critical pose dependency barrier and making high-fidelity dynamic modeling feasible in real-world settings. Figure 1 presents a visual demonstration of PF-4DGS. The principal contributions are threefold:

- We propose PF-4DGS, a pose-free framework for 4D Gaussian Splatting that facilitates dynamic scene reconstruction without pre-registered camera poses.
- A robust initialization strategy and a joint optimization scheme that concurrently refines the scene representation and camera parameters are proposed.
- We conduct comprehensive evaluations on multiple real-world benchmarks to validate our method under diverse conditions, which turns out to show highly competitive performance in comparison to SOTA techniques.

The subsequent sections are structured as: Section II reviews related works in dynamic scene reconstruction and pose estimation. Section III explains the pose-free 4D Gaussian splatting framework for dynamic scene reconstruction. Section IV evaluates our method through experiments on real-world scenarios and highlights its superior performance over several benchmarks. Eventually, Section V concludes our work and proposes potential future study trends.

II. Related Works

Reconstruction of dynamic scenes and NVS are vibrant domains in graphics and computer vision. This section reviews recent advances in scene representation, camera pose estimation, and pose-free optimization frameworks, emphasizing methods that are most relevant to our approach.

A. Scene Representation for Novel View Synthesis

Effective scene representation is pivotal in NVS. NeRF [3] pioneered the utilization of neural networks to model scenes as continuous volumetric functions, leading to high quality static scenes. However, the computational inefficiency of NeRF and its dependence on dense sampling have spurred the development of alternative representations. Structured-NeRF [9] employs a hierarchical scene graph representation to encode both geometric properties and semantic attributes of objects, enabling enhanced scene comprehension that significantly improves NVS performance. Approaches such as PlenOctrees [10] and Instant-NGP [11] enhance NeRF's performance by employing spatial data structures like octrees and hash grids, thus allowing real-time rendering. For dynamic scenes, extensions such as D-NeRF [6] and TiNeuVox [12] adapt NeRF to model temporal variations.

A more recent and promising alternative is 3DGS [5], which represents scenes as collections of ellipsoidal Gaussians. This representation offers a continuous and efficient framework that supports real-time rendering. Building on

this concept, 4D Gaussian Splatting (4DGS) extends the model to dynamic scenes by integrating a temporal dimension. Although these methods demonstrate impressive results, they are heavily dependent on accurate camera pose, which severely limits their applicability in real-world settings where reliable pose data is often inaccessible.

B. Camera Pose Estimation

Precise estimation of camera pose is essential for scene reconstruction. Conventional approaches including Structure-from-Motion (SfM) [13] and Simultaneous Localization and Mapping (SLAM) [14] estimate camera poses by establishing correspondences across multiple views. Recently, COLMAP [13] provides robust pipelines for static scenes, however, its effectiveness in dynamic environments remains limited.

Additionally, learning-based approaches have emerged for pose estimation. Techniques such as Superglue [15] and NerfingMVS [16] leverage deep learning to infer camera poses directly from images, which improves robustness in challenging conditions. However, these methods still require substantial training data or strong priors, restricting their applicability in dynamic and unconstrained settings.

C. Pose-Free Reconstruction Frameworks

Recently, pose-free or joint optimization frameworks have been proposed to eliminate the reliance on predefined camera poses. For instance, BARF [17] introduces a bundle-adjusting NeRF that optimizes both camera poses and scene representation simultaneously. Similarly, iNeRF [18] inverts NeRF to estimate poses, facilitating pose-free reconstruction in static contexts. More recent works including NoPe-NeRF [19] and NeRF$--$zhe [20], extend the framework to dynamic scenes. Despite their potential, these methods are often computationally expensive and require strong initialization.

In summary, while recent advances have substantially improved scene representation and pose estimation techniques, current methods remain fundamentally limited—either restricted to static environments or critically dependent on accurate camera pose information.

III. Preliminary and Framework

In this section, we propose the PF-4DGS algorithm with mathematical formulation presented in a step-by-step manner. Our method initiates from the initialization of 3D Gaussians and camera poses, then performs a joint optimization of Gaussian and pose parameters, rendering, and involves the computation of loss functions. The comprehensive framework is depicted in Figure 2, and the detailed technical procedures are presented in Algorithm 1.

Given a group of unposed images, we leverage the DUSt3R [21] method from to obtain initial estimates of the camera poses and corresponding depth images. Following the initialization step, we derive the 3D Gaussian points. These Gaussian points are then projected using the estimated camera poses. Using a differential rasterizer, we generate the rendered images. Subsequently, we jointly optimize the rendered images ($\mathcal{L}_{\text{render}}$), the estimated camera poses ($\mathcal{L}_{\text{pose}}$), and the Gaussian parameters ($\mathcal{L}_{\text{gaussian}}$) in an iteratively manner, which aims to reconstruct the dynamic scene and generate novel views. The detailed steps are stated below.

A. Step 1: Input and Initialization

The input of the algorithm consists of a set of images $\{I_k\}_{k=1}^{K}$ sampled from dataset $\mathcal{D}$, which are captured from different viewpoints with unknown camera poses.

Gaussian Initialization. We initialize a set of N 3D Gaussian variables $\mathcal{G}_0 = \mathcal{G} = \{G_i\}_{i=1}^{N}$, where each variable G_i is parameterized in the following manner:

$$G_i = \{\mu_i, \Sigma_i, \mathbf{c}_i, \alpha_i\}, \tag{1}$$

where $\mu_i \in \mathbb{R}^3$ is the mean position of the Gaussian variable, $\Sigma_i \in \mathbb{R}^{3\times3}$ is the covariance matrix that controls shape and orientation; $\mathbf{c}_i \in \mathbb{R}^3$ is the color frame (e.g., RGB) and $\alpha_i \in [0, 1]$ is the opacity degree of the image.

Pose Initialization. Then the initial camera poses $\mathcal{P}_0 = \{(R_k, \mathbf{t}_k)\}_{k=1}^{K}$ are estimated such that:

$$R_k \in SO(3), \quad \mathbf{t}_k \in \mathbb{R}^3. \tag{2}$$

where we leverage DUSt3R [21] from to complete these steps and generate the related depth images simultaneously.

B. Step 2: Joint Optimization

2.1 Camera Projection: The 3D-to-2D projection of each Gaussian primitive G_i onto the k-th camera plane is computed through the following transformation:

$$\mu_i^{\text{proj}} = K \cdot (R_k \cdot \mu_i + \mathbf{t}_k), \tag{3}$$

where $K \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix, and $\mu_i^{\text{proj}} \in \mathbb{R}^2$ is the projected 2D position of the Gaussian center.

The covariance matrix of the projected Gaussian is:

$$\Sigma_i^{\text{proj}} = J_k \cdot \Sigma_i \cdot J_k^{\top}, \tag{4}$$

where J_k is the Jacobian of the projection function with respect to the Gaussian parameters.

2.2 Rendering. The rendered color at pixel p in the k-th image is computed by aggregating contributions of all G_i:

$$I_k(p) = \sum_{i=1}^{N} \alpha_i \cdot \mathcal{N}(\mathbf{p} \mid \mu_i^{\text{proj}}, \Sigma_i^{\text{proj}}) \cdot \mathbf{c}_i, \tag{5}$$

where $\mathcal{N}(\mathbf{p} \mid \mu_i^{\text{proj}}, \Sigma_i^{\text{proj}})$ is the Gaussian kernel value at pixel p, and $\alpha_i \cdot \mathbf{c}_i$ is the weighted color contribution of G_i.

C. Step 3: Loss Computation

3.1 Compute Photometric Loss. A photometric loss between rendered images and input images is computed to guide the optimization procedure of the next step:

$$\mathcal{L}_{\text{render}} = \sum_{k=1}^{K} \sum_{p\in\mathcal{P}_k} \|I_k(p) - I_k^{\text{target}}(p)\|^2, \tag{6}$$

where $\mathcal{P}_k$ is the set of all pixels in the k-th image and I_k^{target} is the ground-truth image for the k-th view.

3.2 Compute Pose Regularization Loss. To ensure stable optimization of camera poses, a regularization term is added:

$$\mathcal{L}_{\text{pose}} = \lambda_{\text{pose}} \sum_{k=1}^{K} \|R_k^\top R_k - I\| + \|\mathbf{t}_k\|^2, \tag{7}$$

where λ_{pose} is the weight for the pose regularization term, and I is the identity matrix to constrain $R_k \in SO(3)$.

3.3 Compute Regularization Loss. The Gaussian parameters are regularized for a compact scene representation:

$$\mathcal{L}_{\text{gaussian}} = \lambda_{\text{gaussian}} \sum_{i=1}^{N} \|\Sigma_i\|_F^2, \tag{8}$$

where $\|\Sigma_i\|_F^2$ is the Frobenius norm of the covariance matrix, and $\lambda_{\text{gaussian}}$ is the regularization weight.

D. Step 4: Optimization Objective

The final optimization objective combines all loss terms:

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{gaussian}}. \tag{9}$$

The optimization procedure jointly updates:

- Gaussian parameters $\mathcal{G} = \{\mu_i, \Sigma_i, \mathbf{c}_i, \alpha_i\}_{i=1}^{N}$.
- Camera poses $\{(R_k, \mathbf{t}_k)\}_{k=1}^{K}$.

E. Step 5: Output

The output of the optimization process includes:

- **Optimal 4D Gaussian representation:** a set of Gaussians $\mathcal{G}^*$ representing the spatiotemporal structure of the scene.
- **Refined camera poses:** Accurate camera poses $\mathcal{P}^* = \{(R_k, \mathbf{t}_k)\}_{k=1}^{K}$ for all input views.

These outputs facilitate high-quality NVS and dynamic scene reconstruction, which are reflected in the experiment section.

Algorithm 1: PF-4DGS for Dynamic Scene Reconstruction

Input: Dataset $\mathcal{D}$, initial Gaussians $\mathcal{G}_0$, initial poses $\mathcal{P}_0$, learning rate α, maximum iterations T
Output: Optimal Gaussians $\mathcal{G}^*$ and poses $\mathcal{P}^*$

Initialize parameters: $\mathcal{G} \leftarrow \mathcal{G}_0$, $\mathcal{P} \leftarrow \mathcal{P}_0$;
for $t = 1, 2, \ldots, T$ **do**
  Compute photometric loss $\mathcal{L}_{\text{render}}$;
  Compute pose regularization loss $\mathcal{L}_{\text{pose}}$;
  Compute gaussian regularization loss $\mathcal{L}_{\text{gaussian}}$;
  Compute total loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{render}} + \mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{gaussian}}$;
  Compute gradients:
    - $\nabla_{\mathcal{G}} \mathcal{L}$ (gradient w.r.t. Gaussian parameters)
    - $\nabla_{\mathcal{P}} \mathcal{L}$ (gradient w.r.t. poses)
  Update parameters using gradient descent:
    - $\mathcal{G} \leftarrow \mathcal{G} - \alpha \nabla_{\mathcal{G}} \mathcal{L}$
    - $\mathcal{P} \leftarrow \mathcal{P} - \alpha \nabla_{\mathcal{P}} \mathcal{L}$
  if *stopping criterion (e.g.,* $\|\nabla \mathcal{L}\| < \epsilon$*) is met* **then**
    break;
return Optimal parameters $\mathcal{G}^*$ and $\mathcal{P}^*$;

IV. EXPERIMENTS

A. Platforms and Datasets

We implemented our proposed methods on a single NVIDIA GeForce RTX 3090, utilizing PyTorch as the experimental environment. Considering the practical requirements, we employed real-world datasets of HyperNeRF [22] and Neu3D [4]. The configuration settings for these datasets were informed by the foundational work on 4DGS [8]. The selected datasets enable a thorough evaluation of our methods' performance in practical scenarios.

B. Evaluation Metrics

The evaluation metrics employed in our experiments include the multiscale structural similarity index (MS-SSIM), peak signal-to-noise ratio (PSNR), the structural dissimilarity index measure (D-SSIM), the learned perceptual image patch similarity (LPIPS), training times, storage requirements and frames per second (FPS). These metrics provide a comprehensive assessment of our PF-4DGS method.

C. Results

Tables I and II include the quantitative experimental results for the HyperNeRF and Neu3D datasets, respectively. The **best** and **second best** results are indicated in **green** and **blue**, separately. Note that the rendering convergence speed is influenced by image resolution. For a more consistent comparison, we standardized the resolution to 960×540 and 1352×1014 for the respective datasets.

The numbers in both tables quantify the performance by different methods. Evidently, some NeRF-based methods, namely NeRFPlayer [23], HyperReel [24], and HexPlane-all [25], converge in an unexpectedly slow speed, even after several hours. Additionally, certain grid-based NeRF methods struggle to render objects with intricate details, such as TiNeuVox-B [12], HexPlane-all [25], KPlanes [26], and MSTH [27]. Through certain benchmarks demonstrate promising results in specific evaluation metrics, such as MSTH [27] converging faster than PF-4DGS on the Neu3D dataset and 3D-GS [5] achieving higher FPS on the same dataset, their overall performance is inferior in other metrics.

Figure 3 illustrates the effects of various benchmarks on the HyperNeRF dataset, with tested objects including a broom, banana, chicken, and 3D printer. The related pictures obviously indicate that static 3D-GS [5] struggles with dynamic scene reconstruction. Among the methods evaluated, both 4DGS [8] and our proposed approach demonstrate superior results. Notably, upon closer inspection of the broom and chicken targets, the rendered images of the slipper, corner, and sleeve exhibit enhanced clarity using our method.

D. Limitations and Improvements

Although PF-4DGS demonstrates promising results, several limitations have been identified. Firstly, the model struggles with occlusions and fast-moving objects, causing artifacts or inaccuracies in reconstructed scenes. We will integrate optical flow for improving the model's robustness. For example, optical flow may initialize or regularize the

TABLE I

COMPARABLE RESULTS ARE REPORTED ON THE HYPERNERF DATASET.

Model	PSNR(dB)	MS-SSIM	Times	FPS	Storage(MB)
Nerfies [28]	22.1	0.804	∼hours	<1	-
HyperNeRF [22]	22.3	0.824	31.5 hours	<1	-
TiNeuVox-B [12]	24.5	0.841	31 mins	1	49
3D-GS [5]	19.8	0.684	41 mins	55	51
FFDNeRF [29]	24.2	0.842	-	0.05	440
4DGS [8]	25.2	0.845	1 hours	34	61
Ours	25.8	0.853	28 mins	50	45

TABLE II

COMPARABLE RESULTS ON NEU3D DATASET. HERE, THE SCENE RESOLUTION IS LIMITED TO 1352 X 1014.

Model	PSNR(dB)	D-SSIM	LPIPS	Time	FPS	Storage(MB)
NeRFPlayer [23]	30.68	0.035	0.110	6 hours	0.045	-
HyperReel [24]	31.11	0.034	0.099	9 hours	2.0	358
Hexplane-all [25]	31.68	0.015	0.074	12 hours	0.2	251
KPlanes [26]	31.65	-	-	1.9 hours	0.3	307
Im4D [30]	32.55	-	0.209	29 mins	∼5	94
MSTH [27]	32.39	0.015	0.057	20 mins	2.0	137
4DGS [8]	31.15	0.016	0.049	40 mins	30	90
Ours	32.64	0.013	0.036	26 mins	42	86

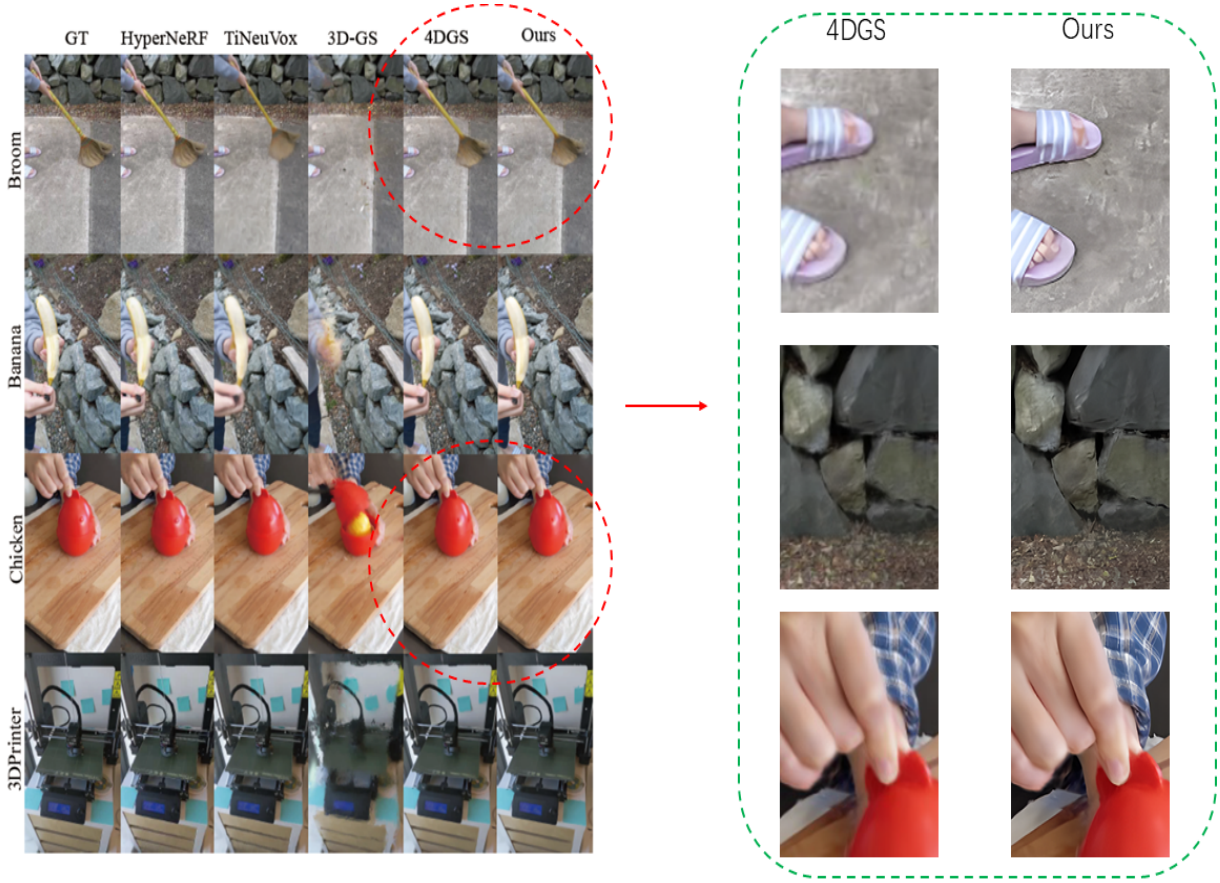

Fig. 3. Comparable visualized results that different methods obtained in HyperNeRF datasets.

Gaussian updates, ensuring smoother transitions and better handling of motion blur or occlusions.

Additionally, performance tends to degrade when input data is sparse or of low resolution, as the model relies heavily on adequate viewpoint coverage and spatial detail. To mitigate the effects of sparse inputs, regularization techniques, such as Gaussian and pose regularization terms, are essential for stabilizing the optimization process. Techniques like view synthesis and multi-view consistency can also be employed to extrapolate missing information from existing views. For low-resolution inputs, integrating super-resolution techniques to up-sample images prior to processing may

improve the model's competence to capture fine details. Alternatively, hierarchical optimization strategies, such as coarse-to-fine Gaussian updates, can refine scene representations even when working with low-resolution data. While the attention mechanism used in our model is not inherently interpretable, visualization techniques like saliency maps can provide insights into its decision-making process.

Finally, model compression techniques such as quantization and pruning facilitate deployment on edge devices, although careful tuning is necessary to maintain accuracy. In summary, experimental results demonstrate the applicability of PF-4DGS across various scenarios. Improved computational efficiency would further adapt it to real-time systems.

V. Conclusions

In this paper, we proposed PF-4DGS, an advanced pose-free 4DGS framework tailored for dynamic scene reconstruction as well as novel view synthesis. Our approach effectively eliminates the need for precomputed camera poses, jointly optimizing Gaussian parameters and pose estimation, which results in high-quality reconstructions and robust performance across a variety of scenarios, all while maintaining fast convergence speeds. Various experiments on the HyPerNeRF and Neu3D datasets indicate that PF-4DGS transcends existing benchmarks in the field of accuracy, rendering quality, and computational efficiency. Since it is still challenging to achieve excellent performance in scenes containing occlusions and fast-moving objects, we will extend our framework to process such scenarios and develop more compact and robust methods.

References

[1] H. Li, Y. Mai, M. Gao, J. He, Z. Liu, and H. Wang, "Large-scale lidar-based loop closing via combination of equivariance and invariance on SE (3)," *IEEE/ASME Transactions on Mechatronics*, 2025.

[2] J. Ham, M. Kim, S. Kang, K. Joo, H. Li, and P. Kim, "San francisco world: Leveraging structural regularities of slope for 3-dof visual compass," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 382–389, 2025.

[3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[4] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe, *et al.*, "Neural 3d video synthesis from multi-view video," in *IEEE/CVF conference on computer vision and pattern recognition*, pp. 5521–5531, 2022.

[5] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D gaussian splatting for real-time radiance field rendering.," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 139–1, 2023.

[6] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *IEEE/CVF conf. on computer vision and pattern recognition*, pp. 10318–10327, 2021.

[7] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 6498–6508, 2021.

[8] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4D Gaussian splatting for real-time dynamic scene rendering," in *IEEE/CVF conference on computer vision and pattern recognition*, pp. 20310–20320, 2024.

[9] Z. Zhong, J. Cao, S. Gu, S. Xie, L. Luo, H. Zhao, G. Zhou, H. Li, and Z. Yan, "Structured-NeRF: Hierarchical Scene Graph with Neural Representation," in *European Conference on Computer Vision*, pp. 184–201, 2024.

[10] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *IEEE/CVF international conference on computer vision*, pp. 5752–5761, 2021.

[11] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[12] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, and Q. Tian, "Fast dynamic radiance fields with time-aware neural voxels," in *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–9, 2022.

[13] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE/CVF conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

[14] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[15] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4938–4947, 2020.

[16] Y. Wei, S. Liu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5610–5619, 2021.

[17] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5741–5751, 2021.

[18] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1323–1330, 2021.

[19] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "Nope-nerf: Optimising neural radiance field with no pose prior," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4160–4169, 2023.

[20] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "Nerf−−: Neural radiance fields without known camera parameters," *arXiv:2102.07064*, 2021.

[21] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

[22] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *arXiv preprint arXiv:2106.13228*, 2021.

[23] L. Song, A. Chen, Z. Li, Z. Chen, L. Chen, J. Yuan, Y. Xu, and A. Geiger, "Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2732–2742, 2023.

[24] B. Attal, J.-B. Huang, C. Richardt, M. Zollhoefer, J. Kopf, M. O'Toole, and C. Kim, "HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16610–16620, 2023.

[25] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 130–141, 2023.

[26] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12479–12488, 2023.

[27] F. Wang, Z. Chen, G. Wang, Y. Song, and H. Liu, "Masked space-time hash encoding for efficient dynamic scene reconstruction," *Advances in neural information processing systems*, vol. 36, pp. 70497–70510, 2023.

[28] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *IEEE/CVF international conference on computer vision*, pp. 5865–5874, 2021.

[29] X. Guo, J. Sun, Y. Dai, G. Chen, X. Ye, X. Tan, E. Ding, Y. Zhang, and J. Wang, "Forward flow for novel view synthesis of dynamic scenes," in *IEEE/CVF International Conference on Computer Vision*, pp. 16022–16033, 2023.

[30] H. Lin, S. Peng, Z. Xu, T. Xie, X. He, H. Bao, and X. Zhou, "High-fidelity and real-time novel view synthesis for dynamic scenes," in *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–9, 2023.