D^{2}GS: Deblurring Deformable 3D Gaussian Splatting for Motion-Blurred Causal Videos

Chaoran Feng, Jianbin Zhao, Wangbo Yu, Zhenyu Tang, Yuchen Li, Li Yuan†, and Yonghong Tian†, *Fellow, IEEE*

***Abstract*—Recent advancements in dynamic 3D gaussian splatting (3DGS) methods have yielded remarkable outcomes. However, these approaches rely on the assumption of sharp input images and existing dynamic 3DGS methods often struggle to generate high-quality dynamic view synthesis when faced with motion blur. Although some NeRF-based approaches have attempted to address this issue, they still struggle to generate high-quality results due to the inaccuracy in estimating continuous dynamic representations over the exposure time and the difficulty in incorporating effective supervision. To tackle this problem, we introduce Deformable Gaussian Splatting as the scene representation and propose a novel framework, D^{2}GS, to recover sharp and high-quality scenes from a monocular, motion-blurred casual video. Specifically, we separately model dynamic and static scenes and transform the continuous dynamic representation estimation over the exposure time into the estimation, simulating the blur generation process. Additionally, we integrate temporal-spatial consistency modeling with regularization terms to mitigate artifacts. Furthermore, we adopt a coarse-to-fine progressive training strategy during training for a faster and more stable optimization of the Gaussian point clouds. Experimental results on our dataset demonstrate that our method outperforms state-of-the-art approaches in generating sharp novel views from motion-blurred inputs. The video demos and partial codes are available in the supplementary materials.**

I. Introduction

Recently, 3D Gaussian Splatting (3DGS) [1] has emerged as a promising scene representation technique, significantly advancing the field of novel view synthesis (NVS). This approach has gained increasing attention, particularly for representing dynamic scenes, offering broad applications in the various AR/VR applications [2]–[6], including several studies [7]–[11] have focused on improving dynamic scene representation using 3D Gaussian Splatting. However, the efficacy of these dynamic 3DGS (namely 4DGS) methods diminishes when confronted with challenging inputs, such as those depicted in Figure 2, where motion-blurred images, prevalent in casually captured videos, present a notable obstacle.

Several 3DGS-based deblurring approaches [12]–[16] have primarily focused on tackling motion blur in static scenes. For example, BAD-GS [15] models the blur formation process by jointly optimizing Gaussian parameters and camera motion trajectories during the exposure period. Deblur-GS [12] simulates camera motion offsets to generate blurry views and refines 3DGS by minimizing the difference between the blurry observation and the generated blur. BAGS [16] estimates per-pixel convolution kernels to model blur and uses a coarse-to-fine optimization strategy to avoid suboptimal results. Despite these efforts, they still struggle to handle motion-blur in dynamic scenes, particularly when it comes to distinguishing object motion from camera motion. The core difficulty arises from real-world conditions, where prolonged camera exposure and irregular motion often result in blurred video captures. Existing 4DGS techniques typically depend on sharp inputs to disentangle object and camera motion or require additional regularizations, such as depth and optical flow, to infer motion and geometric details. Regrettably, motion blur degrades these critical data priors, and hampers the preservation of motion information, disrupts frame-to-frame coherence, complicating the precise modeling of camera trajectories and temporal representations [17].

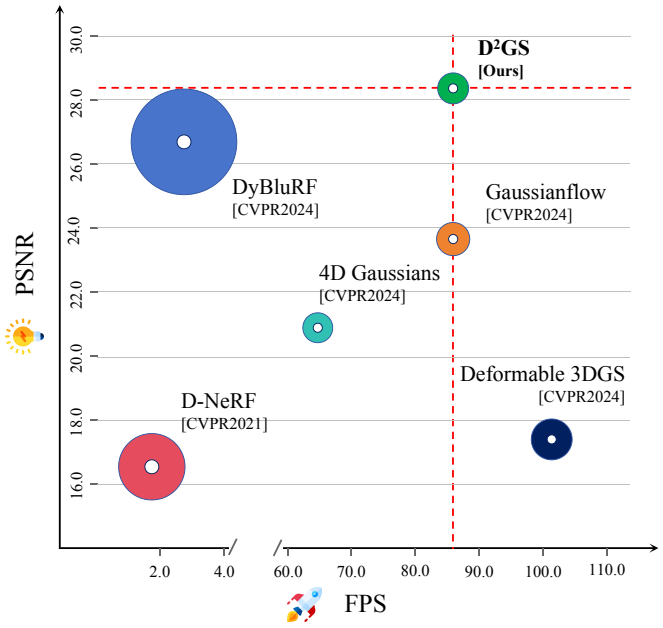

Fig. 1: **Comparison of proposed method with baselines.** Our method achieves higher PSNR with faster training and inference speed than the SoTA NeRF-based and GS-based methods on the DybluRF dataset [17].

Based on the above observation, we are interested in studying the following research question: *How to robustly reconstruct a dynamic scene based on the 4DGS representation from motion-blurred video capture under general real-world conditions?* One simple way is to retrofit a two-stage reconstruction method which utilizes the image or video deblurring method [18], [19] and then reconstruct the 4D scene with deblurred results. However, such naïve approach is inherently constrained by the quality of deblurred video frames, which

Chaoran Feng, Wangbo Yu, Zhenyu Tang, Yuchen Li, Li Yuan and Yonghong Tian are with Peking University and Peng Cheng Laboratory. Jianbin Zhao is with Dalian University of Technology.

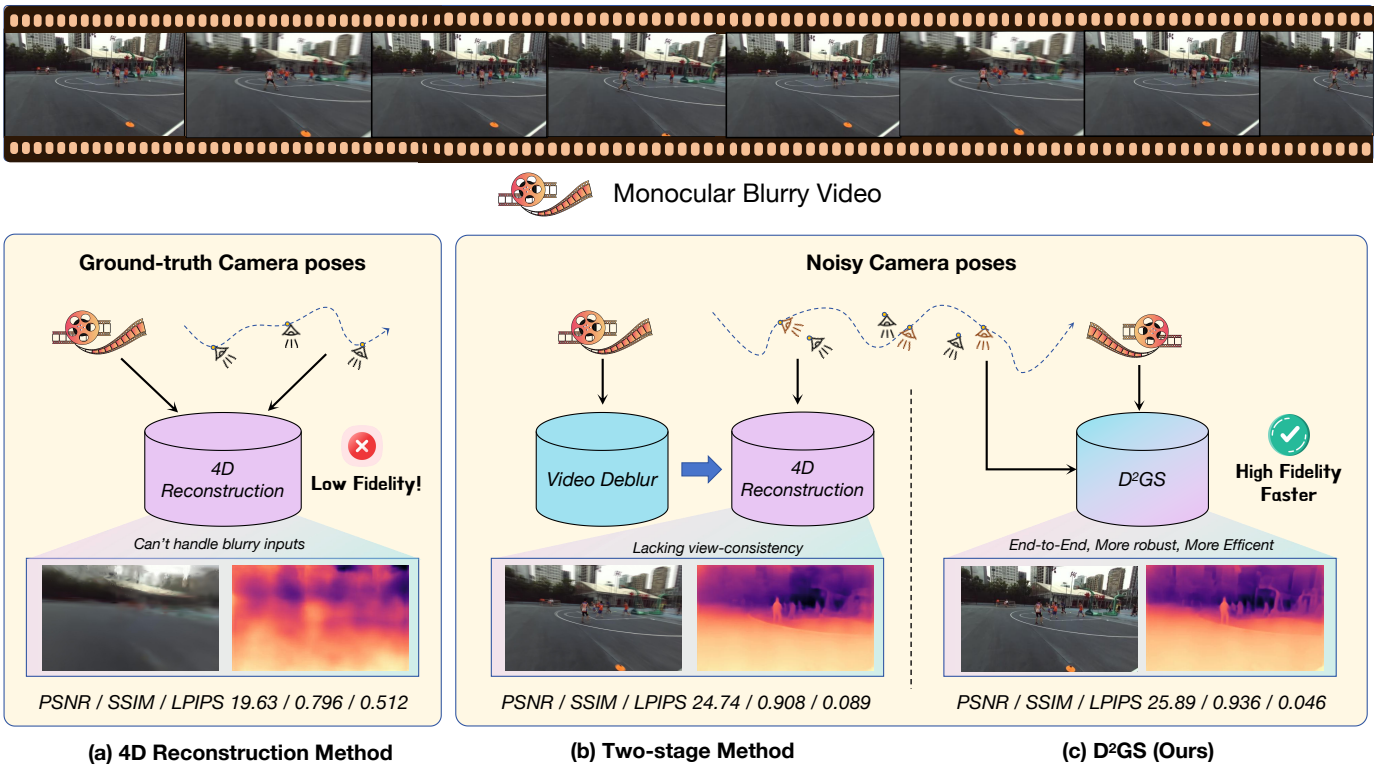

Fig. 2: **Comparison between conventional methods and proposed pipeline.** Existing methods rely heavily on sharp and high-quality inputs, making them unsuitable for motion-blurred video inputs. A straightforward approach is to combine a video deblurring model with 4DGS reconstruction, which fails to account for 3D geometric consistency. In contrast, our end-to-end model decouples deblurring and motion modeling, enabling robust and efficient reconstruction from blurry inputs.

inevitably introduce inconsistencies in geometry and texture. Such imperfections, in turn, give rise to inaccurate camera pose estimation, as these frames are presumed to faithfully depict the scene. In contrast, DyBluRF [17] has proposed reconstructing dynamic neural radiance field (NeRF [20]) directly using blurry inputs and 2D motion mask priors. Nevertheless, it heavily relies on volume rendering, which is significantly slow in practice, and cannot directly decompose the static and dynamic scene which mitigates ambiguities between camera and object movement during joint optimization by associating camera motion with static geometry.

To address these challenges, we propose ***D*^{2}*GS*** (***D***eblurring ***D***eformable ***G***aussian ***S***platting), a single-stage framework tailored for reconstructing a high-fidelity 4DGS representation from motion-blurred casual videos. Specifically, **D^{2}GS** integrates a realistic motion-aware camera trajectory model that considers exposure time and employs blur-aware reprojection regularization to jointly optimize camera poses and scene geometry, effectively mitigating the impact of object motion and camera jitter on trajectory modeling. Subsequently, it disentangles the modeling of static and dynamic scene components, incorporating three regularization terms to ensure spatiotemporal consistency in the dynamic reconstruction. One regularization term focuses on the temporal-spatial dependencies among the attributes of each Gaussian, another applies an entropy-based loss to enforce surface smoothness, and the final term captures the frequency details of the entire Gaussian scene. Finally, we introduce a coarse-to-fine training strategy that enhances training stability and accelerates convergence with blurry inputs, while simultaneously yielding better reconstructions results.

We comprehensively evaluate the performance of **D^{2}GS** in novel view synthesis. As illustrated in Figure 1, it achieves state-of-the-art quality in novel view synthesis while maintaining real-time rendering efficiency. To summarize, our contributions are as follows:

- We propose **D^{2}GS**, a novel 4DGS pipeline which separately reconstructs the static and dynamic scene into the whole scene from blurry monocular videos.
- We introduce a motion-aware camera trajectory model to simulate the blur generation during the exposure time, and then jointly optimizes camera poses and 4DGS.
- We integrate temporal-spatial consistency modeling with regularization terms to ensure physically plausible motion and consistent geometry, and adopt a coarse-to-fine training strategy to fit the blurry inputs well.
- Extensive experiments have demonstrated the effectiveness of the proposed method. Results on the DyBluRF datasets validate the SOTA performance of our approach in dynamic scene reconstruction.

II. Related Work

A. Novel view synthesis and Video Deblurring

Novel view synthesis (NVS) with deblurring methods have been widely explored. Deblur-NeRF [21] and DP-NeRF [22] simulate the ray marching from each viewpoint with NeRF-based model. BAD-Gaussians [15], Deblur-GS [12] and others [23]–[25] explicitly model the camera trajectory during the exposure time, assuming that the motion speed of captured cameras is uniform. EvaGaussians [26], LSE-NeRF [27] and EaDeblurGS [28] add the constraint from neuromorphic cameras to jointly optimize the deblurring formation and scene respresentation while the methods above still remain static. Moreover, video deblurring methods [18], [19], [29]–[38] effectively leverage temporal cues across consecutive frames to improve restoration quality. However, these methods either focus solely on the physical principles of motion blur generation or emphasize consistency across adjacent views, without jointly considering the 4D representation and camera motion, thereby limiting their capacity to comprehensively capture the underlying scene dynamics.

B. Dynamic Scene Representation

Recently, NeRF [20] and 3DGS [1] serve as the excellent 3D respresentation methods in the implicit and explicit domain, respectively. But expanding static scene representation to dynamic scenes is not a simple task. Some NeRF-based methods [39]–[51] have made progress based on deformation fields which modeling the whole scene as a canonical field and a deformation field, or reduce the dimensionality of the 4D space by decomposing it into a set of planar or hash grids. However, these methods requires dense sampling along rays, limiting the possibility of real-time rendering. Owing to the efficient training and inference speeds, numerous concurrent works based on 3DGS representation have been proposed. We review 3DGS-based methods [7], [9]–[11], [52]–[67], [67]–[69] for dynamic reconstruction, including those that deform the 3D canonical space, utilize dynamic 3D Gaussians, and leverage embeddings and spatial relationships of Gaussians.

1) Dynamic Reconstruction with the Deformable Field: DeformableGS [7] and 4DGS [53] reconstruct the dynamic scene with a tiny deformable multilayer perceptron (MLP) which inputs the center position of the canonical 3D Gaussians and timestamps. WassersteinGS [54] employ Wasserstein distance and deformation to ensure the smoothness and consistence of Gaussian primitives motion, reducing motion artifacts. Dreamscene4D [52] and S4D [55] decompose the static and dynamic scene, jointly modeling the object deformation and the background geometry.

2) Dynamic Reconstruction with Probabilistic Model: Real-time GS [56] and Rotor4DGS [9] enabling 4D Gaussian to be decomposed into a conditional 3D Gaussian and a 1D annotation Gaussian, decoupling the 3D and 4D reconstruction tasks. STGS [58] represents dynamic changes in 3D Gaussian over time through a temporal opacity and a polynomial function for each Gaussian primitive.

3) Dynamic Reconstruction with Time-Spatial Constraint: There are some flow-based and anchor-based Gaussian methods for dynamic reconstruction. Flow-based methods, GFlow [57], MotionGS [10], Shape-of-motion [61], GaussianFlow [59] and others [11], [60] compose the movement of 3D points through their corresponding Gaussians, using 2D physical priors to supervise the deformation of 3D Gaussians. DynamicGS [62] utilizes regularization to encourage that Gaussians and their neighbors deform with local rigidity.

However, the methods mentioned above are primarily designed for high-quality inputs and dynamic reconstruction, and they often struggle with deblurring inputs due to the ambiguity between camera and object movement in the absence of physical cues.

To overcome this challenge, we leverage the pre-trained Align3R model [70] to extract Structure-from-Motion (SfM) points, camera extrinics and intrinics. Additionally, we integrate the off-the-shell tracking model, Tracking Anything [71], to generate motion masks, followed by the application of Gaussianflow [59] as the dynamic reconstruction model.

III. Preliminary

A. Dynamic 3D Resprensentation

3DGS [1] represents the static scene with a series of sparse 3D Gaussians. Each Gaussian is parameterized by a covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$ and location $\mu \in \mathbb{R}^3$:

$$\mathbf{G}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}, \tag{1}$$

where $\mathbf{\Sigma}$ can be further factorized into a scaling matrix $\mathbf{S} \in \mathbb{R}^3$ and a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$, represented as $\mathbf{\Sigma} = \mathbf{RSS}^\top\mathbf{R}^\top$. To render the scene from a novel view under a specific camera pose, the covariance matrix in camera coordinates, denoted as $\mathbf{\Sigma}' = \mathbf{JW\Sigma W}^\top\mathbf{J}^\top$, where $\mathbf{J}$ represents the Jacobian of the affine approximation of the projective transformation, and $\mathbf{W}$ denotes the view transformation matrix. This transformation ensures that the covariance matrix is adjusted according to the camera's perspective, enabling the accurate rendering of the scene from a new viewpoint.

We extend Gaussianflow [59] as the dynamic reconstruction model, which is extended to model dynamic scenes from 3DGS by tracking the trajectory of each Gaussian using Dual-Domain Deformation Model (DDDM). We assume that only the rotation $\mathbf{R}$, radiance c, and position μ of each 3D Gaussian point change over time, while the scaling $\mathbf{S}$ and opacity α remain constant. The temporal variation of each Gaussian's attributes is modeled as the sum of its base attributes $\mathbf{A}_0 \in \mu_0, c_0, \mathbf{R}_0$ at the reference time frame t_0 and a time-dependent attribute residual $\mathbf{D}(t)$. The time-dependent residual $\mathbf{D}(t)$ is modeled as a combination of polynomial fitting in the time domain and Fourier series fitting in the frequency domain, expressed as:

$$\mathbf{A}(t) = \mathbf{A}_0 + \mathbf{D}(t), \tag{2}$$

where $D(t) = P_N(t) + F_L(t)$ is combined by a polynomial $P_N(t)$ with coefficients $\mathbf{a} = \{a_n\}_{n=0}^N$ and a Fourier series

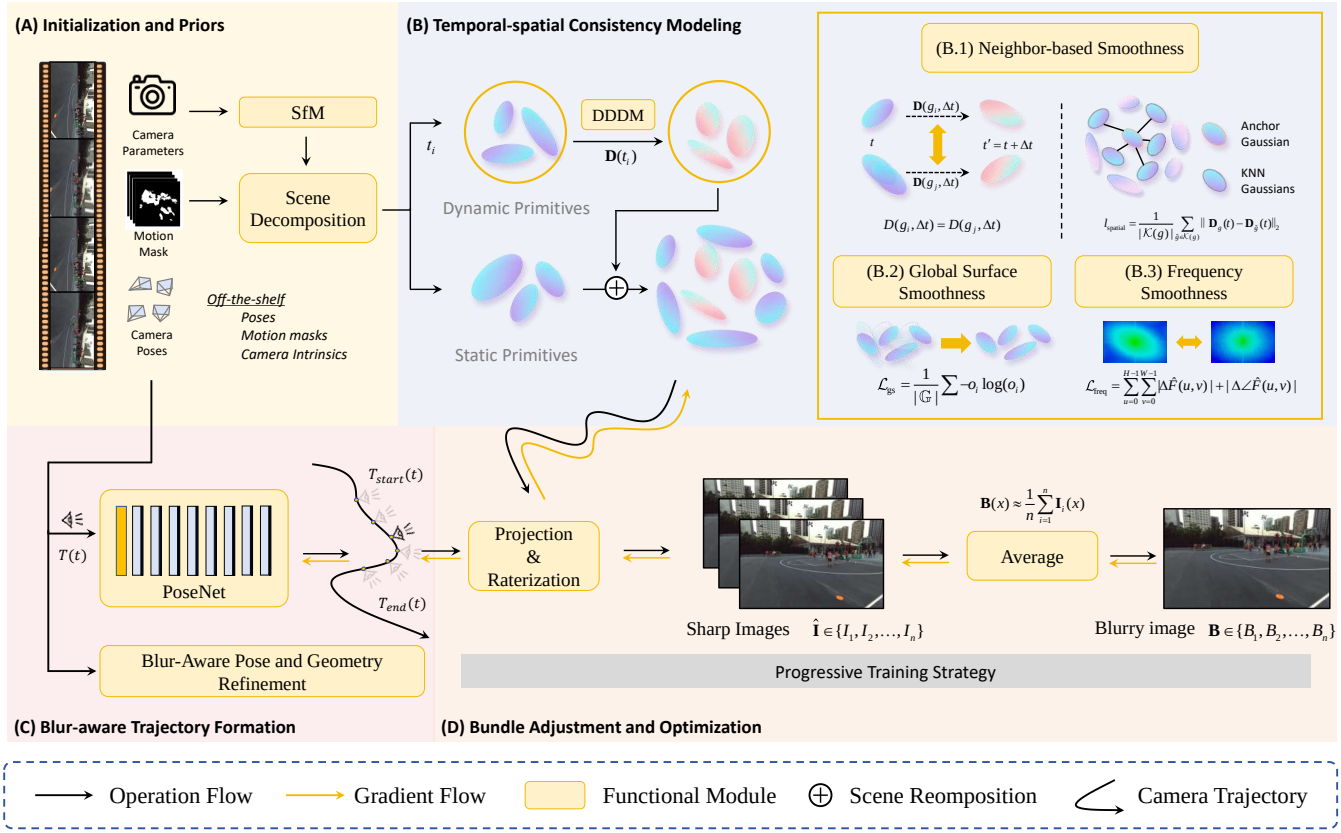

Fig. 3: **Overview of the proposed method.** Our method takes as input the estimated pose P_t and motion mask M_t derived from Align3R [70], along with the blurry video frame B_t at timestamp t. The system then processes these inputs to jointly optimize the dynamic and static scene representation based on Gaussianflow [59], with three smoothness regularization terms, to accurately render novel views across multiple viewpoints and timestamps. We model the blur formation with the pre-trained PoseNet to get the camera poses during exposure time τ. Finally, we employ a progressive training strategy to better capture the finer details of the whole scene and mitigate blurring artifacts from Gaussian floators.

$F_L(t)$ with coefficients $\mathbf{f} = \{f^l_{\sin}, f^l_{\cos}\}_{l=1}^{L}$. These are respectively denoted as:

$$P_N(t) = \sum_{n=0}^{N} a_n t^n, \tag{3}$$

$$F_L(t) = \sum_{l=1}^{L} \left(f^l_{\sin} \cos(lt) + f^l_{\cos} \sin(lt) \right). \tag{4}$$

By integrating both the static and dynamic components in this way, we can effectively model dynamic scenes, where the separation of static and dynamic representations enhances the robustness and stability of the optimization process.

B. Motion Blur Formation

The physical process of motion blur can be interpreted as the consequence of the camera or object motion within exposure time τ, expressed as:

$$\mathbf{B}(x) = \frac{1}{c} \int_0^{\tau} \mathbf{I}_t(\mathbf{x})\, dt \tag{5}$$

where $I_t(x)$ denotes the sharp image captured at timestamp t, and c denotes the normalization factor of exposure time. To approximate the integration, we uniformly discretize the exposure time τ into n timestamps and process the captured blurry image as the average of sharp images over the exposure period:

$$\mathbf{B}(x) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}_i(\mathbf{x}) \tag{6}$$

Here, as n increases, the simulated motion blur becomes more realistic in theory, but it also incurs higher computational cost. Therefore, a trade-off must be struck between the accuracy of motion blur modeling and the training efficiency.

IV. Method

In this section, we introduce our proposed reconstruction method, D^{2}GS, which models dynamic scenes using a video sequence of N motion-blurred frames $B_t \in \mathcal{B}^{N\times H\times W\times 3}$ and motion masks $M_t \in \mathcal{M}^{N\times H\times W}$. The overall pipeline is illustrated in Figure 3. Our goal is to reconstruct the sharp 4D Gaussian scene of the scene along with the corresponding camera trajectory $P_t \in \mathcal{P}^{N\times 4\times 4}$. In IV, we elaborate on modeling the physical image formation process of motion blur during the exposure time. We achieve this by jointly optimizing the scene geometry and camera poses using a multi-view correspondence

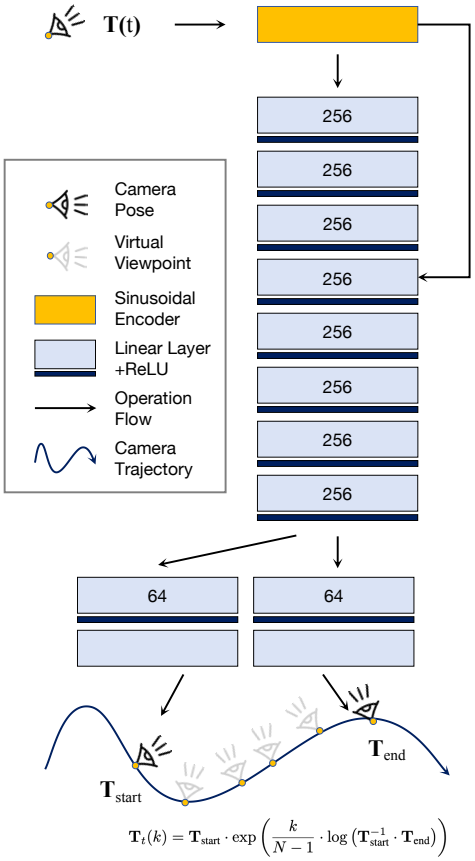

Fig. 4: **The structure of the pose network.** The tiny MLP layers predict the start and end camera poses from the estimated pose P_t during the exposure time τ.

loss to learn a globally consistent 3D solution. In Sec. IV-B, we introduce a temporal-spatial partitioning of the dynamic Gaussian representation $\mathbb{G}$. This partitioning separates the representation into two distinct subsets: the static Gaussian set $\mathbb{G}$sta, which captures the background, and the dynamic Gaussian set $\mathbb{G}$dyn, which models the moving objects in the scene. To resolve ambiguities arising from the interactions between camera motion and object movement, we employ three regularization terms. Finally, in Sec.IV-C, we propose a progressive training strategy to ensure stable training and improve performance, and then in Sec.IV-D, we provide details of the training and optimization process.

A. Motion-aware Trajectory Formation

To simulate the real image formation process described in Sec.III, we accumulate sharp images over the exposure time using the *Camera Trajectory Model*, which simulates the camera poses during the imaging process. However, simulating only the camera motion is insufficient, as we observe that errors in estimated poses—whether obtained from COLMAP [72] or Align3R [70]—can significantly impact the sharp reconstruction. Therefore, in this section, we propose a *Blur-Aware Pose and Geometry Refinement* strategy. This strategy enforces view-correspondence consistency and further mitigates the impact of motion blur on both trajectory modeling and the optimization of Gaussian attributes.

Camera Trajectory Model. Previous works such as BAD-Gaussians [15] formulate and optimize the corresponding poses of each latent sharp image during the camera exposure time τ by employing a camera motion trajectory represented through linear interpolation between adjacent camera poses. Specifically, for each blurry frame input B_t, we jointly optimize both the exposure start pose $\mathbf{T}_{\text{start}} \in \text{SE}(3)$ and exposure end pose $\mathbf{T}_{\text{end}} \in \text{SE}(3)$ over the interval τ. The virtual viewpoint $\mathbf{T}(t)$ at timestamp $t \in [0, \tau]$ is then expressed as follows:

$$\mathbf{T}(t) = \mathbf{T}_{\text{start}} \cdot \exp\left(\frac{t}{\tau} \cdot \log\left(\mathbf{T}_{\text{start}}^{-1} \cdot \mathbf{T}_{\text{end}}\right)\right) \tag{7}$$

However, in many real-world scenarios, the exposure time τ is not explicitly provided, and the poses at either the start or end of the exposure are often unavailable. To address this issue, we employ a pose prediction network inspired by [20], [73], denoted as $\mathcal{F}(\cdot)$, to model the camera trajectory and decouple latent sharp images, as illustrated in Figure 4. Specifically, the start and end poses are predicted as follows:

$$(\mathbf{T}_{\text{start}}, \mathbf{T}_{\text{end}}) = \mathcal{F}(\mathbf{T}(t)) \tag{8}$$

$$\mathbf{T}_t(k) = \mathbf{T}_{\text{start}} \cdot \exp\left(\frac{k}{N-1} \cdot \log\left(\mathbf{T}_{\text{start}}^{-1} \cdot \mathbf{T}_{\text{end}}\right)\right) \tag{9}$$

Here, $\frac{t}{\tau}$ is further discretized as $\frac{k}{N-1}$ for the sampled virtual sharp image at timestamp k in Eq. 6, and our poses $\mathcal{P}$, which includes both the start and end poses, is optimized to mitigate the cumulative error introduced by inaccuracies in the initial pose estimation.

However, directly training a single MLP network to represent longer periods of exposure time presents several challenges. The MLP training process would remain unstable, significantly slowing both the training and the interpretation of the scene representation learning. Moreover, the estimated pose $\mathbf{T}(t)$ directly affects the representational capacity of the MLP network, exacerbating the accumulation of errors throughout the pipeline.

In the next subsection, we address this issue by introducing a view-correspondence loss that enforces temporal-spatial consistency, inspired by [74]. This loss jointly optimizes the dynamic scene representation and camera pose, mitigating the errors in pose estimation and improving the overall stability and accuracy of the reconstruction process.

Blur-Aware Pose and Geometry Refinement. Specifically, for a pixel p in the blurry image B_{t_n} of reference view at time t_n with predicted depth $D_{t_n}(p)$, we project it into one of the neighbor view at the next timestamp t_{n+1} to obtain pixel p' and its depth value $D_{t_{n+1}}(p')$, Then we re-project p' back to the reference view as p'' and determine its depth $D_{t_{n+1}}(p'')$. Notely, the blurry image pair is without motion objects but only background. The view-correspondence loss is defined by minimizing the reprojection error as:

$$\begin{aligned} \xi_p^{t_n} &= \|p - p''\|_2, \\ \xi_d^{t_n} &= \frac{\|D_{t_n,i}(p) - D_{t_{n+1},i}(p'')\|_1}{D_{t_n,i}(p)}, \\ \mathcal{L}_{reproj} &= \sum_{t_n \in T, p \in B_{t_n}} (\xi_p^{t_n} + \beta \xi_d^{t_n}). \end{aligned} \tag{10}$$

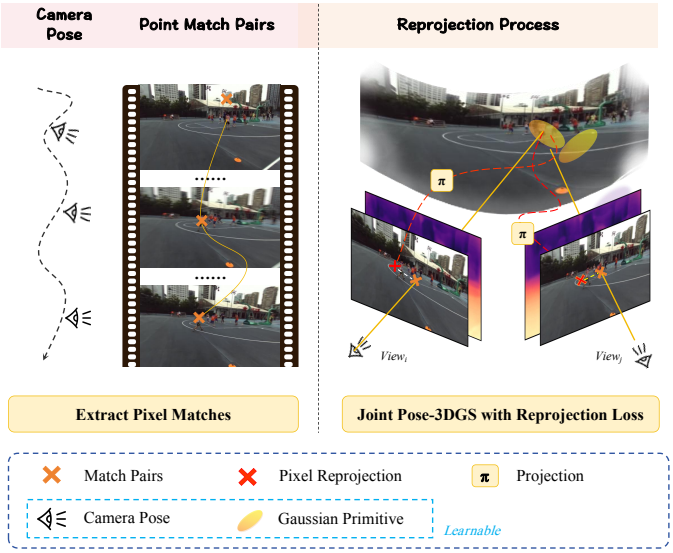

Fig. 5: The illustration of blur-aware pose and 4DGS optimization. We use the off-the-shell model to extract the pixel pairs from the randomly selected images and compute the view-correspond reprojection loss between sampled pixels.

where, $\xi_p^{t_n}$, $\xi_d^{t_n}$ and $\beta = 100$ represent the back-projection errors in pixel space, depth space and a parameter that balances the two re-projection metrics, respectively. Moreover, by given a blurry image and its next frame $(B_{t_n}, B_{t_{n+1}})$, we obtain the matched pixel pairs derived from the off-the-shelf feature matching model [75]. Then, we calculate the dynamic geometric consistency for each pixel in every frame to filter out unreliable pixels and correct the inaccurate poses.

We establish correspondences between training images to jointly optimize the static scene $\mathbb{G}_{\text{static}}$ and the dynamic scene $\mathbb{G}_{\text{dynamic}}$, alleviating the adverse effects of motion blur on the modeling of dynamic objects.

B. Temporal-spatial Consistency Modeling

Efficient prediction of the exposure process from a specific viewpoint is crucial, but relying solely on camera models to reconstruct sharp dynamic scenes is insufficient. To achieve more robust and high-quality reconstructions, more effective regularization techniques and more sophisticated methods are required. Moreover, the inherent limitations of the current reconstruction model in capturing fine details of moving objects and occluded regions across motion-blurred video frames present significant challenges.

In this section, we partition the dynamic Gaussian representation into two distinct subsets: the static scene and the dynamic scene. We introduce three regularization modules to address these challenges and ensure spatiotemporal consistency in the dynamic reconstruction process under blurred input conditions. Specifically, for the static Gaussian primitives $\mathbb{G}_{\text{sta}}$, we adopt the original 3DGS method [1], while for the dynamic scene, we utilize Gaussianflow [59] to model the dynamic Gaussian primitives $\mathbb{G}_{\text{dyn}}$.

Neighbor-based Temporal-spatial Smoothness. Firstly, we utilize a k-nearest neighbor (KNN) algorithm to partition the 4D Gaussian primitives of a dynamic Gaussian $g \in \mathbb{G}_{dyn}$ into distinct subsets. Furthermore, we apply regularization on the input timestamp t to promote temporal-spatial dependencies among the attributes of each Gaussian and its neighbors. The temporal and spatial regularization terms are defined as:

$$l_{time} = \frac{1}{|\mathbb{G}_{dyn}|} \sum_{g \in \mathbb{G}_{dyn}} ||\mathbf{D}(g,t) - \mathbf{D}(g,t+\epsilon)||_2, \tag{11}$$

$$l_{spatial} = \frac{1}{|\mathcal{K}(g)|} \sum_{\tilde{g} \in \mathcal{K}(g)} ||\mathbf{D}_g(t) - \mathbf{D}_{\tilde{g}}(t)||_2, \tag{12}$$

where, $\mathcal{K}(\cdot)$ denotes the KNN algorithm and $\mathbf{D}$ denotes the deformable term of each gaussian primitive as defined in Sec. III-A. The temporal-spatial loss is defined as follows:

$$\mathcal{L}_{\text{ts}} = \lambda_t l_{time} + \lambda_s l_{spatial} \tag{13}$$

where λ_t and λ_s are the weighting factors for the temporal and spatial smoothness terms, respectively.

Global Surface Smoothness. Assuming that each Gaussian primitive should ideally lie close to the object surface, its opacity o_i is expected to be near one in most cases [9]. To enforce this property, we leverage the entropy principle to encourage opacity values to be close to either zero or one. Specifically, Gaussian primitives with near-zero opacity are pruned during training by default. The corresponding entropy loss is defined as:

$$\mathcal{L}_{\text{gs}} = \frac{1}{|\mathbb{G}|} \sum -o_i \log(o_i), \tag{14}$$

where $|\mathbb{G}| = |\mathbb{G}_{\text{dyn}} + \mathbb{G}_{\text{sta}}|$ denotes the total number of Gaussian primitives. We observe that $\mathcal{L}_{\text{gs}}$ effectively condenses Gaussian points and suppresses noisy floaters, proving particularly beneficial when training with blurry input views.

Frequency-aware Loss. We observe that during the training process with blurry inputs, the high-frequency details, such as the trees and ground, tend to become over-smoothed or even lost. To mitigate this issue and complete training within a limited number of iterations, we introduce a frequency loss to better preserve these high-frequency details. Specifically, we apply a 2D discrete Fourier transform to convert both the rendered image I and the ground truth I_{gt} into their respective frequency representations, F and F_{gt}. $F(u,v)$ can be further expressed in terms of amplitude $\left|F(u,v)\right|$ and phase $\angle F(u,v)$, where (u,v) denotes the coordinates in the frequency spectrum. We then introduce a high-pass filter in the frequency domain to extract high-frequency information, denoted as $\hat{F}(u,v)$ and $\hat{F}_{gt}(u,v)$. We define $\Delta\left|\hat{F}(u,v)\right| = \left|\hat{F}(u,v)\right| - \left|\hat{F}_{gt}(u,v)\right|$ and $\Delta\angle\hat{F}(u,v) = \angle\hat{F}(u,v) - \angle\hat{F}_{gt}(u,v)$. Thus, the frequency loss $\mathcal{L}_{\text{freq}}$ and the total loss $\mathcal{L}_{\text{total}}$ can be formulated as follows:

$$\mathcal{L}_{\text{freq}} = \sum_{u=0}^{H-1} \sum_{v=0}^{W-1} \left|\Delta\left|\hat{F}(u,v)\right|\right| + \left|\Delta\angle\hat{F}(u,v)\right|, \tag{15}$$

Here, H, W and λ_{freq} denote the image height, width, and the hyperparameter to balance the loss.

C. Progressive Training Strategy

Training Gaussian primitives typically involves computing the photorealistic loss over the full image resolution. This results in a more complex loss landscape, as Gaussians are required to fit fine scene details from the early stages of training. Moreover, blurry images should be treated as a weak signal during the early stages of training, as using full-resolution inputs may interfere with the optimization of other components, ultimately destabilizing the training process. To address this, we propose a coarse-to-fine training strategy, which incorporates different decay schemes (linear decay, exponential decay, inverse square root decay, and cosine decay). The strategy begins by rendering at a low scene resolution r_0 and gradually increases the resolution over N training iterations until the full resolution $r_N = H \times W$ is achieved. This process is defined as follows:

$$r(t) = r_0 + (r_{N-1} - r_0) \cdot \frac{1}{2}\left(1 + \cos\left(\pi \cdot \frac{t}{T}\right)\right), \quad (16)$$

where cosine decay is used and t denotes the training iteration index, with $0 \leq t \leq N-1$. As the render resolution increases, more Gaussians can be utilized to better capture the finer details of the dynamic scene, enabling more accurate modeling of the camera trajectory. Furthermore, the proposed training strategy allows the representation of coarser scenes with fewer Gaussians, mitigating blurring artifacts from Gaussian floaters while ensuring faster rendering and high-quality reconstruction during training as we show in Sec. V-B.

D. Overall Training Pipeline

Loss Function. We use the photometric losses including an L_1 reconstruction loss and an SSIM loss as in the original 3DGS [1] and Gaussianflow [59] framework, and it combines the dynamic Gaussians and static Gaussians to generate the rendered image $\hat{B}_t$ and the blurry image $\hat{\mathbf{B}}_t$ for each viewpoint and timestamp, expressed as follows:

$$\mathcal{L}_{\text{render}} = \lambda_r \mathcal{L}_1(\hat{\mathbf{B}}_t, \mathbf{B}_t) + (1 - \lambda_r)\mathcal{L}_{ssim}(\hat{\mathbf{B}}_t, \mathbf{B}_t) \quad (17)$$

Assembling all loss terms, we get the overall loss function for the overall pipeline:

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \mathcal{L}_{\text{ts}} + \lambda_{\text{gs}}\mathcal{L}_{\text{gs}} + \lambda_{\text{reproj}}\mathcal{L}_{\text{reproj}} + \lambda_{\text{freq}}\mathcal{L}_{\text{freq}} \quad (18)$$

Implementation Details. We implemented D^{2}GS based on the official code of 3DGS [1], Gaussianflow [59] and DIFT [75]. Before the entire training process, we pre-train the PoseNet $\mathcal{F}$ in the static scene for 1,000 iterations and the rest settings of training follows the original 3DGS [1]. Then we set the initial rendering resolution $r_0 = 0.2 \times r_N$, the number of sampled pixel and the loss weight $\lambda_r = 0.8$, $\lambda_t = \lambda_s = 1.0$, $\lambda_{gs} = 0.01$, $\lambda_{\text{freq}} = 0.002$ and $\lambda_{reproj} = 0.001$. During the training process, we firstly train the model for a 1,000-iteration warmup the for the Camera Trajectory Model (see Sec. IV), 30,000 steps for the DDDM Model and then 30,000 steps for dynamic and static primitives. Moreover, we omit the densification process to streamline and simplify the subsequent optimization. The learning rate for the Camera Trajectory Model is set to $1.0e^{-4}$ with a weight decay to $5.0e^{-5}$ and the time shift $\epsilon = 0.08$ and the virtual viewpoint number is set to $N = 8$. The learning rate of the DDDM module is the same as the Gaussianflow setting. All experiments are conducted on a single NVIDIA A6000 with 48GB memory and more training details are in the supplementary materials.

V. Experiment

A. Experiment Settings

In this section, we compare our proposed method with several state-of-the-art techniques in the tasks of novel view synthesis, motion deblurring, and pose accuracy. We perform both qualitative and quantitative evaluations using the proposed synthetic dataset and the existing real dataset. The details of the experiments are as follows:

1) Datasets: We evaluate our proposed methods on six scenes with significant motion blur from the DybluRF dataset [17]. Additionally, to evaluate the effect of our camera trajectory model, we introduce a synthetic dataset of Pointodyssey [76] generated using Blender [77]. This dataset includes ground-truth camera poses, high-quality rendered images (540×960), and artificially generated blurry images (with 10 frames accumulated), covering six diverse indoor and outdoor scenes. These camera extrinsic and intrinsic parameters are calibrated using Align3R [70], consistent with the settings in the DybluRF dataset. We use the blurry images, estimated poses, and intrinsic parameters as the input to our proposed method.

2) Baselines: **D^{2}GS** performs 4DGS reconstruction and camera motion recovery for blurry formation using only a monocular camera. To the best of our knowledge, there are no existing deblurring 4DGS methods that do not rely on ground-truth poses, except the NeRF-based method DyBluRF [17]. Therefore, we conduct a comprehensive comparison of our method with several 4D-GS approaches, including Gaussianflow [59], Deformable3DGS [7], Shape-of-motion [61], and Real-Time GS [56]. Additionally, we compare our method against two-stage approaches, such as off-the-shelf video deblurring models, including RTVD [29], RWVD [18], and STDAN [30], combined with the 4DGS reconstruction model. In these methods, blurry images are first processed using the respective deblurring models, followed by camera pose estimation from the deblurred images using COLMAP [72]. Subsequently, RTVD [29], RWVD [18], and STDAN [30] are trained using the generated deblurred images and estimated poses. Both the quantitative and qualitative comparisons in the novel view synthesis (NVS), deblurring view synthesis (DVS) and camera trajectory evaluation tasks, are performed on the synthetic dataset. Since there are no paired ground truth poses for the real dataset, we only perform quantitative results of NVS and DVS, and qualitative comparisons on the real dataset.

3) Metrics: The metrics are utilized for the NVS and DVS tasks, include the commonly employed Peak Signal-to-Noise Ratio (PSNR) [79], Structural Similarity Index Measure (SSIM) [80], and VGG-based Learned Perceptual Image Patch Similarity (LPIPS) [81] between rendered views and ground-truth views of the whole dataset. Additionally, for camera

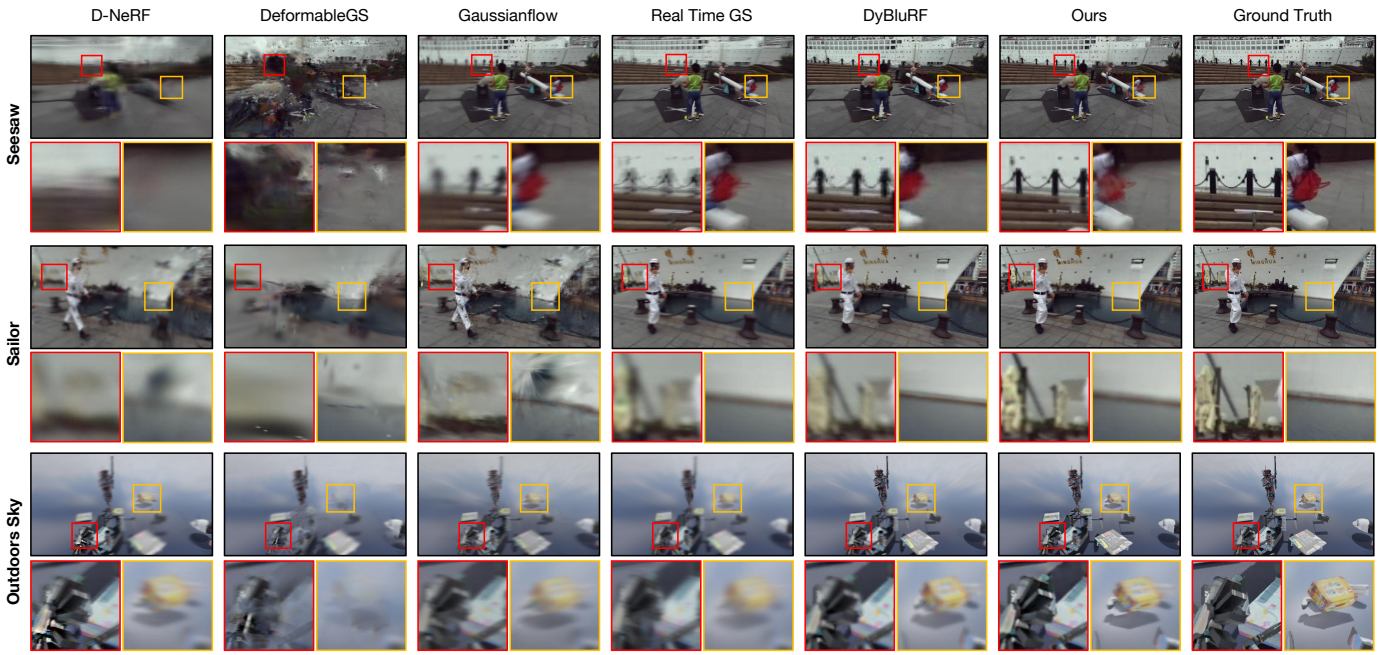

Fig. 6: **Qualitative results of novel view synthesis.** Compared with 4D reconstruction-based methods, our approach produces more realistic rendering results with fine-grained details.

TABLE I: **Quantitative evaluations of novel view synthesis on the DeBluRF dataset and proposed dataset.** Each baseline method is trained with its public code under the original settings and evaluated with the same evaluation protocol and ATE is in the ground truth scale. The best results are highlighted in **bold**. '↓' or '↑' indicate lower or higher values are better.

Novel View Synthesis	Pose	DyBluRF Dataset [17]			Proposed Synthetic Dataset			
	Opt.	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	ATE ↓
D-NeRF [39]	×	16.83	0.654	0.663	18.54	0.682	0.581	1.708
DeformableGS [7]	×	18.18	0.690	0.429	20.17	0.759	0.407	1.708
GaussianFlow [59]	×	23.93	0.838	0.142	23.26	0.827	0.255	1.708
Real-Time GS [56]	×	22.67	0.812	0.165	21.70	0.796	0.285	1.708
DyBluRF [17]	✓	27.26	0.893	0.118	29.34	0.932	0.098	0.457
D^{2}GS	✓	**28.34**	**0.906**	**0.098**	**30.27**	**0.943**	**0.077**	**0.353**

trajectory evaluations, we employ Absolute Trajectory Error (ATE) to assess accuracy in the proposed synthetic dataset. Furthermore, we use Relative Pose Error (RPE), which is divided into RPE$_t$ (translational error) and RPE$_r$ (rotational error), to evaluate the precision of trajectory estimation in terms of both translation and rotation.

B. Experiment Results

In this section, we compare proposed method with the 5 state-of-the-art (SOTA) dynamic reconstruction methods (i.e. D-NeRF [39], DeformableGS [7], Gaussianflow [59], Real-TimeGS [56] and DyBluRF [17]) and their variants with video-deblur models (i.e. RTVD [29], RWVD [18] and STDAN [30]). The quantitative results for novel view synthesis, deblurring view synthesis, and trajectory precision are summarized in Table. II and Table. III, respectively. Furthermore, qualitative results are presented in Fig. 6, Fig. 7 and Fig. 8, while the rendering speed comparisons for 720 × 1280 images are shown in Fig 2.

1) Evaluation of novel view synthesis: Quantitatively, the experimental results in Table II demonstrate the superiority of deblurring dynamic reconstruction methods over other baselines in both novel view synthesis rendering quality and pose estimation accuracy. Furthermore, our proposed method outperforms deblurring baselines across all evaluation metrics for both synthetic and real scenes, indicating that it more effectively decouples the 3D scene representation from the camera motion information.

Qualitatively, the effectiveness of our method is demonstrated in Fig 6, where it is evident that our approach generates more photo-realistic and detailed renderings compared to the baseline. The synthesized views, in both synthetic and real scenes, exhibit sharper edges and better preservation of finer textures. In contrast, baseline methods suffer from various artifacts, such as blurriness and texture inconsistencies. Furthermore, the visualization of the estimated camera trajectories in Fig 7 shows that our trajectory aligns more closely with the ground truth, achieving a significantly smaller ATE compared to the other baselines.

However, evaluating only on the blurry 3D dataset is insufficient. Therefore, we extend our framework to model sharp dynamic scenes, using the DyBluRF dataset [17]. In

TABLE II: **Quantitative comprisons between two-stages methods and end-to-end models.** Notely, in the two-stage method each blurry video is pre-processed with video deblurring methods before reconstruction. 'None' indicates that no deblurring methods are applied and the best results are highlighted in **bold** while the second results are in underline.

Reconstruction Methods	Deblurring Methods	DyBluRF Dataset [17]			Proposed Synthetic Dataset		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
GaussianFlow [59]	None	23.93	0.838	0.142	23.26	0.827	0.255
GaussianFlow [59]	RTVD [29]	**24.81**	0.853	0.138	26.10	0.882	0.115
GaussianFlow [59]	RWVD [18]	24.65	**0.869**	**0.132**	**26.85**	0.895	0.110
GaussianFlow [59]	STDAN [30]	24.29	0.846	0.139	26.50	**0.903**	**0.105**
DyBluRF [17]	None	27.26	0.893	0.118	29.34	0.932	0.098
D^{2}GS	None	**28.34**	**0.906**	**0.098**	**30.27**	**0.943**	**0.077**

TABLE III: Quantitative evaluations of deblurring view synthesis on the DeBluRF dataset and proposed dataset.

Deblur View Synthesis	Model Type	DyBluRF Dataset [17]			Proposed Synthetic Dataset		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
RTVD [29]	Deblurring	33.07	0.969	0.068	35.20	0.983	0.045
RWVD [18]	Deblurring	32.81	0.965	0.062	35.08	0.978	0.047
STDAN [30]	Deblurring	**33.79**	**0.978**	**0.039**	**35.87**	**0.985**	**0.035**
GaussianFlow [59]	Reconstruction	28.95	0.938	0.173	31.20	0.948	0.132
DeformableGS [78]	Reconstruction	28.73	0.931	0.180	31.11	0.947	0.138
Real-Time GS [56]	Reconstruction	**29.75**	**0.943**	**0.155**	**31.86**	**0.953**	**0.123**
DyBluRF [17]	Deblur+Recon.	28.90	0.936	0.174	32.16	0.955	0.124
D^{2}GS	Deblur+Recon.	**31.55**	**0.961**	**0.078**	**33.24**	**0.968**	**0.075**

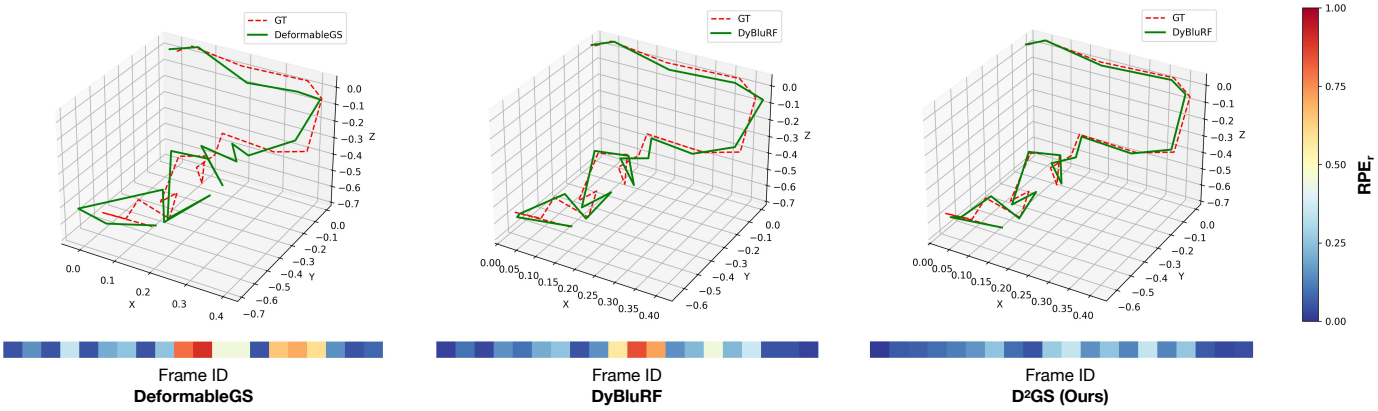

Fig. 7: **Qualitative comparison of pose refinement on the proposed synthetic dataset.** We visualise the trajectory and RPE$_r$ in the synthetic scene *ourdoors grass*. We clip and normalize the RPE$_r$ by a quarter of the max RPE$_r$ across all results.

4D Reconstruction	PSNR↑	SSIM↑	LPIPS↓
D-NeRF [39]	21.94	0.792	0.262
Gaussianflow [59]	23.34	0.810	0.237
DyBluRF [17]	25.31	0.855	0.128
Ours	**26.52**	**0.872**	**0.096**

TABLE IV: Quantitative results of novel view synthesis on the non-blur DyBluRF dataset.

this case, sharp images are used as input, and we evaluate our proposed method under the original settings. Our results, as shown in Table IV, demonstrate that our model outperforms existing state-of-the-art methods. Additionally, Fig.9 illustrates an example where our approach on dynamic scenes surpasses D-NeRF [21] and Gaussianflow [59]. Notably, the performance gap widens in sharp scenarios, with PSNR improvements of nearly 1.21 dB over baselines, emphasizing the method's efficacy in handling sharp 4D reconstruction in real-world settings. This highlights that our model not only achieves higher accuracy but also exhibits superior generalization when handling dynamic scenes.

2) Evaluation of deblurring view synthesis: For the deblurring task, we additionally compare with several state-of-the-art video deblurring methods [18], [29], [30], alongside 4D reconstruction methods [7], [56], [59]. As shown in Table III, GS-based methods demonstrate the ability to decouple the blur task due to 3DGS overfitting to the training viewpoint. Moreover, our proposed method outperforms 4D reconstruction-based approaches (e.g., GaussianFlow [59], DeformableGS [7], Real-TimeGS [56]) and achieves results

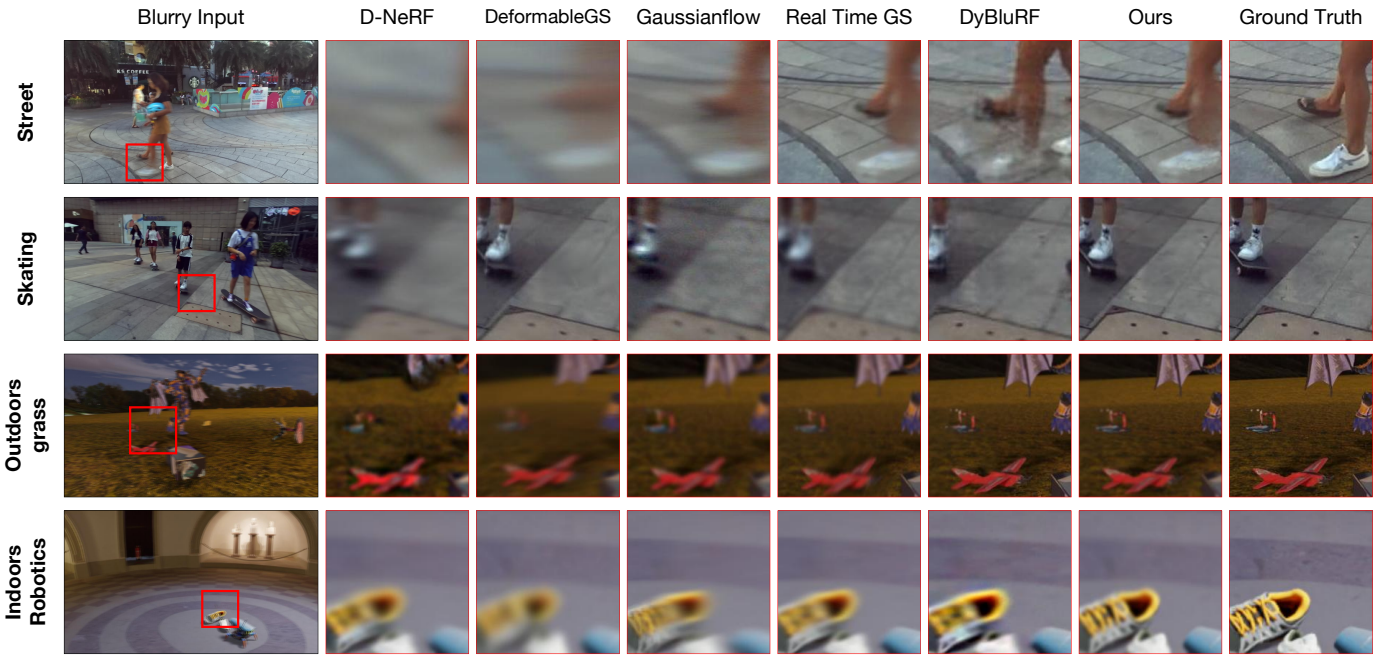

Fig. 8: **Qualitative results of deblurring view synthesis.** Compared with 4D reconstruction-based methods, our approach produces more realistic deblurring results with fine-grained details.

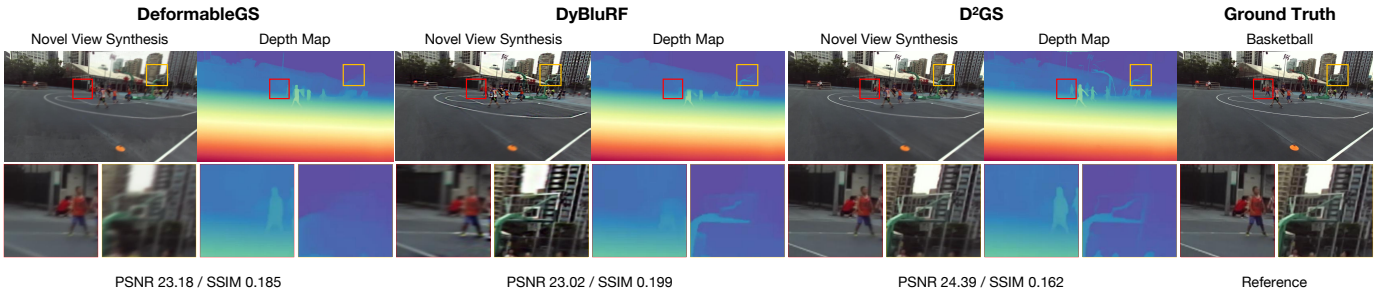

Fig. 9: Qualitative Results of novel view synthesis in the DyBluRF dataset with sharp inputs.

comparable to deblurring-specific methods (e.g., RTVD [29], RWVD [18], STDAN [30]). In comparison to the former, **D^{2}GS** demonstrates superior scene reconstruction, leading to enhanced performance. It is important to note that the deblur-specific methods are trained on large amounts of paired data in a fully supervised manner, whereas **D^{2}GS** is optimized solely using a self-supervised approach on a given blurred video. While the extensive data prior benefits the baseline methods, **D^{2}GS** offers the advantage of being easier to use, without requiring paired data collection.

Qualitatively, the deblurring effectiveness of our method is shown in Fig 8, where it is evident that our approach generates more photo-realistic and detailed renderings while the baseline methods suffer from various artifacts, such as blurriness and texture inconsistencies.

3) Accuracy of the Camera Motion: To validate the performance of different methods under inaccurate pose initialization from Align3R [70] with blurry images from a monocular camera, we compare the accuracy of camera poses with the 4D reconstruction methods above at the end of trainging process. The effectiveness of our method is demonstrated in Fig. 7, where it is evident that our approach recover more accurate and

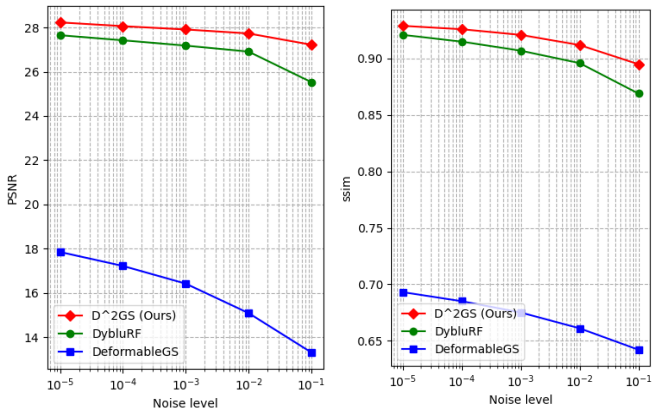

Fig. 10: Ablation study on the robustness of pose refinement module. We add different perturbation to camera poses during the training process and evaluate robustness of this module.

motion-aware camera poses compared to the baseline. And the quantatitive result of our method demostrates the superiority of 4D reconstruction methods over other baselines about the accuracy of pose optimization, as shown in Table. II.

TABLE V: **Ablation study of each component of our approach.** We sequentially add regularization terms ($\mathcal{L}_{\text{render}}$, $\mathcal{L}_{\text{ts}}$, $\mathcal{L}_{\text{gs}}$, $\mathcal{L}_{\text{reproj}}$, and $\mathcal{L}_{\text{freq}}$) and evaluate their effects on novel view synthesis, deblurring view synthesis, and pose estimation. The complete model achieves the best performance across all metrics.

Regularization Terms					Novel View Synthesis			Deblurring View Synthesis			Pose Evaluation		
$\mathcal{L}_{\text{render}}$	$\mathcal{L}_{\text{ts}}$	$\mathcal{L}_{\text{gs}}$	$\mathcal{L}_{\text{reproj}}$	$\mathcal{L}_{\text{freq}}$	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	RPE_t ↓	RPE_r ↓	ATE ↓
✓					26.35	0.878	0.102	31.05	0.937	0.114	0.127	1.18	0.912
✓	✓				27.48	0.892	0.088	32.05	0.946	0.102	0.083	1.03	0.652
✓	✓	✓			27.12	0.889	0.094	31.68	0.942	0.108	0.075	0.96	0.625
✓	✓	✓	✓		<u>29.04</u>	<u>0.937</u>	<u>0.058</u>	**33.44**	<u>0.965</u>	<u>0.079</u>	**0.039**	**0.56**	**0.350**
✓	✓	✓		✓	27.95	0.903	0.082	32.62	0.952	0.095	0.064	0.89	0.574
✓	✓	✓	✓	✓	**30.26**	**0.952**	**0.047**	<u>33.24</u>	**0.968**	**0.075**	<u>0.041</u>	<u>0.58</u>	<u>0.353</u>

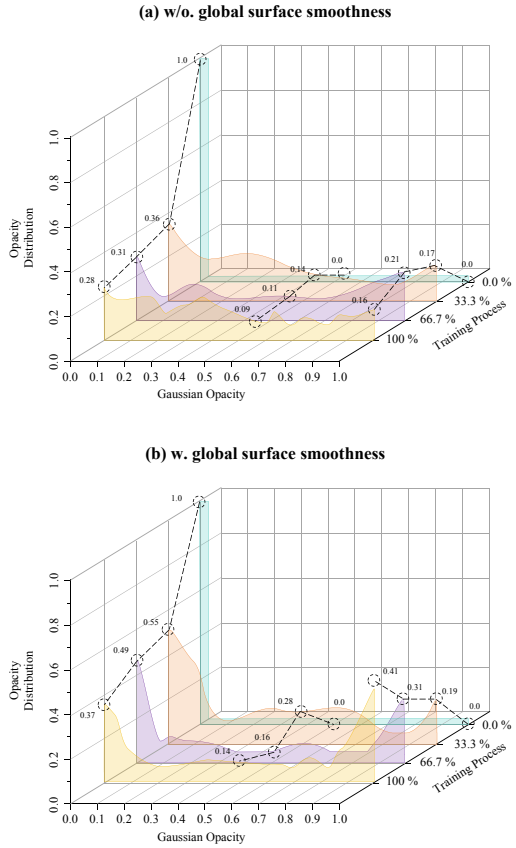

Fig. 11: Qualitative Results of effectiveness of global surface smoothness.

Scaling Method	PSNR	Num. Gaussians	Training Time
(a) None	23.34dB	0.95M	23m27s
(b) Linear Decay	23.31dB	0.61M	31m41s
(c) Exp. Decay	23.36dB	0.61M	31m32s
(d) Cosine Decay	23.41dB	0.75M	29m49s

TABLE VI: **Ablation study of the progressive training strategy.** Specially, training time is only calculated during the dynamic reconstruction period and more details are in the supplementary materials.

C. Ablation Study

We assess the effectiveness of our proposed progressive training optimization strategy. Subsequently, we validate the accuracy of the camera trajectory model. Finally, we evaluate the impact of the regularization terms.

1) The Effectiveness of Progressive Training: As explained previously in Sec. IV-C, progressive scaling of the scene while training provides stable optimizations. We include progressive scaling that increases the resolution of the rendering in a cosine schedule. We achieve gains in PSNR with fewer floating artifacts due to a more stable optimization while significantly reducing the training time. The results are summarized in the Table. VI. Notably, we try different types of strategies for our progressive training, such as (a) with the full resolution (origin), (b) with the linear decay, (c) with the exponential decay and (d) with the cosine decay.

2) The Robustness of Pose Refinement: We introduce varying degrees of perturbation to the initial camera poses to validate the robustnness of the motion-aware trajectory formation module (Sec. IV). Specifically, we parameterize the camera poses p using the SE(3) Lie algebra, which is the same as BARF [82]. For each scene, we synthetically perturb the camera poses with additive noise $\delta_p \sim \mathcal{N}(0, nI)$, where n is the noise level. Each method is then initialized with the noised poses, and optimization is performed thereafter. The results are shown in Fig. 10.

Notably, DeformableGS [7], which lacks the ability to optimize camera poses, suffers a significant performance degradation as the magnitude of pose perturbations increases. This highlights the critical importance of jointly optimizing camera poses alongside scene reconstruction. Furthermore, our proposed framework demonstrates superior robustness across the full spectrum of pose perturbations compared to existing methods (i.e. DyBluRF [17]). Even under significant perturbations, the degradation in novel view synthesis fidelity and trajectory estimation performance is considerably less pronounced compared to the other methods.

3) The Effectness of Regularization Terms: The effect of time-spatial regularization loss $\mathcal{L}_{\text{ts}}$ (Eq. 13), global surface smoothness loss $\mathcal{L}_{\text{gs}}$ (Eq. 14), frequency-aware loss $\mathcal{L}_{\text{freq}}$ (Eq. 15) and view-correspondence reprojection loss $\mathcal{L}_{\text{reproj}}$ (Eq. 10) in Table. V. We observe that the performance drops significantly without these regularization loss as shown in Table. V. Specifically, by smoothing the global surface, $\mathcal{L}_{gs}$ alleviates the notable floaters and artifacts in the whole scene. Furthermore, $\mathcal{L}_{reproj}$ refines the camera poses and significantly improve the blurry formation process during the training. Moreover, the loss $\mathcal{L}_{ts}$ and $\mathcal{L}_{freq}$ regularize the time-spatial consistence of Gaussian primitives and immigrate the artifacts of the whole scene.

VI. Conclusion

In this paper, we propose Deblurring Deformable Gaussian Splatting ($\mathbf{D^2GS}$), a novel framework designed to model sharp and high-quality 4D Gaussian representations from motion-blurred casual videos by motion blur. $\mathbf{D^2GS}$ employs a motion-aware camera trajectory model and decouples static and dynamic scene components using motion masks, ensuring precise reconstruction of scene geometry and camera poses despite motion blur. Additionally, our approach incorporates temporal-spatial consistency regularizations and a coarse-to-fine training strategy to enhance reconstruction quality. Experimental evaluations demonstrate that $\mathbf{D^2GS}$ achieves state-of-the-art performance in novel view synthesis, deblurring view synthesis, and camera motion accuracy. Furthermore, our framework not only excels in reconstructing blurred video inputs but also effectively reconstructs the sharp 4D models.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[2] Z. Tang, J. Zhang, X. Cheng, W. Yu, C. Feng, Y. Pang, B. Lin, and L. Yuan, "Cycle3d: High-quality and consistent image-to-3d generation via generation-reconstruction cycle," *arXiv preprint arXiv:2407.19548*, 2024.

[3] J. Wang, Y. Ma, J. Guo, Y. Xiao, G. Huang, and X. Li, "Cove: Unleashing the diffusion feature correspondence for consistent video editing," *arXiv preprint arXiv:2406.08850*, 2024.

[4] Z. Fan, W. Cong, K. Wen, K. Wang, J. Zhang, X. Ding, D. Xu, B. Ivanovic, M. Pavone, G. Pavlakos, Z. Wang, and Y. Wang, "Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds," 2024.

[5] K. Feng, Y. Ma, B. Wang, C. Qi, H. Chen, Q. Chen, and Z. Wang, "Dit4edit: Diffusion transformer for image editing," *arXiv preprint arXiv:2411.03286*, 2024.

[6] J. Zhang, Z. Tang, Y. Pang, X. Cheng, P. Jin, Y. Wei, W. Yu, M. Ning, and L. Yuan, "Repaint123: Fast and high-quality one image to 3d generation with progressive controllable 2d repainting," *arXiv preprint arXiv:2312.13271*, 2023.

[7] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 331–20 341.

[8] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," *arXiv preprint arXiv:2308.09713*, 2023.

[9] Y. Duan, F. Wei, Q. Dai, Y. He, W. Chen, and B. Chen, "4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11.

[10] X. Guo, W. Zhang, R. Liu, P. Han, and H. Chen, "Motiongs: Compact gaussian splatting slam by motion filter," *arXiv preprint arXiv:2405.11129*, 2024.

[11] A. Kratimenos, J. Lei, and K. Daniilidis, "Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting," in *European Conference on Computer Vision*. Springer, 2025, pp. 252–269.

[12] W. Chen and L. Liu, "Deblur-gs: 3d gaussian splatting from camera motion blurred images," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 7, no. 1, pp. 1–15, 2024.

[13] O. Seiskari, J. Ylilammi, V. Kaatrasalo, P. Rantalankila, M. Turkulainen, J. Kannala, E. Rahtu, and A. Solin, "Gaussian splatting on the move: Blur and rolling shutter compensation for natural camera motion," in *European Conference on Computer Vision*. Springer, 2024, pp. 160–177.

[14] B. Lee, H. Lee, X. Sun, U. Ali, and E. Park, "Deblurring 3d gaussian splatting," in *European Conference on Computer Vision*. Springer, 2024, pp. 127–143.

[15] L. Zhao, P. Wang, and P. Liu, "Bad-gaussians: Bundle adjusted deblur gaussian splatting," *arXiv preprint arXiv:2403.11831*, 2024.

[16] C. Peng, Y. Tang, Y. Zhou, N. Wang, X. Liu, D. Li, and R. Chellappa, "Bags: Blur agnostic gaussian splatting through multi-scale kernel modeling," in *European Conference on Computer Vision*. Springer, 2024, pp. 293–310.

[17] H. Sun, X. Li, L. Shen, X. Ye, K. Xian, and Z. Cao, "Dyblurf: Dynamic neural radiance fields from blurry monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7517–7527.

[18] Z. Zhong, Y. Gao, Y. Zheng, B. Zheng, and I. Sato, "Real-world video deblurring: A benchmark dataset and an efficient recurrent neural network," *International Journal of Computer Vision*, vol. 131, no. 1, pp. 284–301, 2023.

[19] J. Pan, B. Xu, J. Dong, J. Ge, and J. Tang, "Deep discriminative spatial and temporal network for efficient video deblurring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 191–22 200.

[20] 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," in *European Conference on Computer Vision (ECCV)*, 2020. [Online]. Available: https://www.matthewtancik.com/nerf

[21] L. Ma, X. Li, J. Liao, Q. Zhang, X. Wang, J. Wang, and P. V. Sander, "Deblur-NeRF: Neural Radiance Fields from Blurry Images," in *Computer Vision and Pattern Recognition (CVPR)*, 2022. [Online]. Available: https://limacv.github.io/deblurnerf/

[22] D. Lee, M. Lee, C. Shin, and S. Lee, "Dp-nerf: Deblurred neural radiance field with physical scene priors," in *Computer Vision and Pattern Recognition (CVPR)*, 2023.

[23] K. Chen, J. Zhang, Z. Hao, Y. Zheng, T. Huang, and Z. Yu, "Usp-gaussian: Unifying spike-based image reconstruction, pose correction and gaussian splatting," *arXiv preprint arXiv:2411.10504*, 2024.

[24] D. Lee, J. Park, and K. M. Lee, "GS-blur: A 3d scene-based dataset for realistic image deblurring," in *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. [Online]. Available: https://openreview.net/forum?id=Awu8YlEofZ

[25] M. Niu, Y. Zhan, Q. Zhu, Z. Li, W. Wang, Z. Zhong, X. Sun, and Y. Zheng, "Bundle adjusted gaussian avatars deblurring," *arXiv preprint arXiv:2411.16758*, 2024.

[26] W. Yu, C. Feng, J. Tang, X. Jia, L. Yuan, and Y. Tian, "Evagaussians: Event stream assisted gaussian splatting from blurry images," *arXiv preprint arXiv:2405.20224*, 2024.

[27] W. Z. Tang, D. Rebain, K. G. Derpanis, and K. M. Yi, "Lse-nerf: Learning sensor modeling errors for deblured neural radiance fields with rgb-event stereo," *arXiv preprint arXiv:2409.06104*, 2024.

[28] Y. Weng, Z. Shen, R. Chen, Q. Wang, and J. Wang, "Eadeblur-gs: Event assisted 3d deblur reconstruction with gaussian splatting," *arXiv preprint arXiv:2407.13520*, 2024.

[29] H. Son, J. Lee, S. Cho, and S. Lee, "Real-time video deblurring via lightweight motion compensation," in *Computer Graphics Forum*, vol. 41, no. 7. Wiley Online Library, 2022, pp. 177–188.

[30] H. Zhang, H. Xie, and H. Yao, "Spatio-temporal deformable attention network for video deblurring," in *ECCV*, 2022.

[31] Y. Wang, Y. Lu, Y. Gao, L. Wang, Z. Zhong, Y. Zheng, and A. Yamashita, "Efficient video deblurring guided by motion magnitude," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[32] Z. Zhong, X. Sun, Z. Wu, Y. Zheng, S. Lin, and I. Sato, "Animation from blur: Multi-modal blur decomposition with motion guidance," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*. Springer, 2022, pp. 599–615.

[33] Z. Zhong, M. Cao, X. Ji, Y. Zheng, and I. Sato, "Blur interpolation transformer for real-world motion from blur," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5713–5723.

[34] Z. Zhong, Y. Gao, Y. Zheng, and B. Zheng, "Efficient spatio-temporal recurrent neural network for video deblurring," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer, 2020, pp. 191–207.

[35] S. Yang, K.-P. Ning, Y.-Y. Liu, J.-Y. Yao, Y.-H. Tian, Y.-B. Song, and L. Yuan, "Is parameter collision hindering continual learning in llms?" *arXiv preprint arXiv:2410.10179*, 2024.

[36] J. Gast and S. Roth, "Deep video deblurring: The devil is in the details," in *ICCV Workshop on Learning for Computational Imaging (ICCVW)*, Seoul, Korea, Nov. 2019.

[37] J. Pan, B. Xu, J. Dong, J. Ge, and J. Tang, "Deep discriminative spatial and temporal network for efficient video deblurring," in *The IEEE*

Conference on Computer Vision and Pattern Recognition(CVPR), Feb 2023.

[38] W. Shang, D. Ren, Y. Yang, H. Zhang, K. Ma, and W. Zuo, "Joint video multi-frame interpolation and deblurring under unknown exposure time," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 935–13 944.

[39] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[40] 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.

[41] X. Zhao, Z. An, Q. Pan, and L. Yang, "Nerf2: Neural radio-frequency radiance fields," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023, pp. 1–15.

[42] Z. Yan, C. Li, and G. H. Lee, "Nerf-ds: Neural radiance fields for dynamic specular objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8285–8295.

[43] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-Planes: Explicit Radiance Fields in Space, Time, and Appearance," in *Computer Vision and Pattern Recognition (CVPR)*, 2023. [Online]. Available: https://sarafridov.github.io/K-Planes/

[44] J.-W. Liu, Y.-P. Cao, W. Mao, W. Zhang, D. J. Zhang, J. Keppo, Y. Shan, X. Qie, and M. Z. Shou, "Devrf: Fast deformable voxel radiance fields for dynamic scenes," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 762–36 775, 2022.

[45] M. Mihajlovic, S. Prokudin, S. Tang, R. Maier, F. Bogo, T. Tung, and E. Boyer, "Splatfields: Neural gaussian splats for sparse 3d and 4d reconstruction," in *European Conference on Computer Vision*. Springer, 2024, pp. 313–332.

[46] J. Lin, "Dynamic nerf: A review," *arXiv preprint arXiv:2405.08609*, 2024.

[47] Y. Zhan, Z. Li, M. Niu, Z. Zhong, S. Nobuhara, K. Nishino, and Y. Zheng, "Kfd-nerf: Rethinking dynamic nerf with kalman filter," *arXiv preprint arXiv:2407.13185*, vol. 3, 2024.

[48] Z. Yan, C. Li, and G. H. Lee, "Nerf-ds: Neural radiance fields for dynamic specular objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8285–8295.

[49] G.-W. Yang, W.-Y. Zhou, H.-Y. Peng, D. Liang, T.-J. Mu, and S.-M. Hu, "Recursive-nerf: An efficient and dynamically growing nerf," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 12, pp. 5124–5136, 2022.

[50] J.-W. Liu, Y.-P. Cao, W. Mao, W. Zhang, D. J. Zhang, J. Keppo, Y. Shan, X. Qie, and M. Z. Shou, "Devrf: Fast deformable voxel radiance fields for dynamic scenes," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 762–36 775, 2022.

[51] 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.

[52] W.-H. Chu, L. Ke, and K. Fragkiadaki, "Dreamscene4d: Dynamic multi-object scene generation from monocular videos," *arXiv preprint arXiv:2405.02280*, 2024.

[53] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 310–20 320.

[54] J. Deng and Y. Luo, "Gaussians on their way: Wasserstein-constrained 4d gaussian splatting with state-space modeling," *arXiv preprint arXiv:2412.00333*, 2024.

[55] B. He, Y. Chen, G. Lu, Q. Wang, Q. Gu, R. Xie, L. Song, and W. Zhang, "S4d: Streaming 4d real-world reconstruction with gaussians and 3d control points," 2024. [Online]. Available: https://arxiv.org/abs/2408.13036

[56] Z. Yang, H. Yang, Z. Pan, and L. Zhang, "Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting," *arXiv preprint arXiv:2310.10642*, 2023.

[57] S. Wang, X. Yang, Q. Shen, Z. Jiang, and X. Wang, "Gflow: Recovering 4d world from monocular video," *arXiv preprint arXiv:2405.18426*, 2024.

[58] Z. Li, Z. Chen, Z. Li, and Y. Xu, "Spacetime gaussian feature splatting for real-time dynamic view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8508–8520.

[59] Y. Lin, Z. Dai, S. Zhu, and Y. Yao, "Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 136–21 145.

[60] Y.-H. Huang, Y.-T. Sun, Z. Yang, X. Lyu, Y.-P. Cao, and X. Qi, "Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4220–4230.

[61] Q. Wang, V. Ye, H. Gao, J. Austin, Z. Li, and A. Kanazawa, "Shape of motion: 4d reconstruction from a single video," *arXiv preprint arXiv:2407.13764*, 2024.

[62] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," *arXiv preprint arXiv:2308.09713*, 2023.

[63] J. Lu, J. Deng, R. Zhu, Y. Liang, W. Yang, X. Zhou, and T. Zhang, "Dn-4dgs: Denoised deformable network with temporal-spatial aggregation for dynamic scene rendering," *Advances in Neural Information Processing Systems*, vol. 37, pp. 84 114–84 138, 2025.

[64] J. Xu, Z. Fan, J. Yang, and J. Xie, "Grid4d: 4d decomposed hash encoding for high-fidelity dynamic gaussian splatting," *arXiv preprint arXiv:2410.20815*, 2024.

[65] Z. Yang, Z. Pan, X. Zhu, L. Zhang, Y.-G. Jiang, and P. H. Torr, "4d gaussian splatting: Modeling dynamic scenes with native 4d primitives," *arXiv preprint arXiv:2412.20720*, 2024.

[66] A. Bond, J.-H. Wang, L. Mai, E. Erdem, and A. Erdem, "Gaussianvideo: Efficient video representation via hierarchical gaussian splatting," *arXiv preprint arXiv:2501.04782*, 2025.

[67] D. Sun, H. Guan, K. Zhang, X. Xie, and S. K. Zhou, "Sdd-4dgs: Static-dynamic aware decoupling in gaussian splatting for 4d scene reconstruction," 2025. [Online]. Available: https://arxiv.org/abs/2503.09332

[68] J. Yan, R. Peng, L. Tang, and R. Wang, "4d gaussian splatting with scale-aware residual field and adaptive optimization for real-time rendering of temporally complex dynamic scenes," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 7871–7880.

[69] Y. Lu, Y. Zhou, D. Liu, T. Liang, and Y. Yin, "Bard-gs: Blur-aware reconstruction of dynamic scenes via gaussian splatting," 2025. [Online]. Available: https://arxiv.org/abs/2503.15835

[70] J. Lu, T. Huang, P. Li, Z. Dou, C. Lin, Z. Cui, Z. Dong, S.-K. Yeung, W. Wang, and Y. Liu, "Align3r: Aligned monocular depth estimation for dynamic videos," *arXiv preprint arXiv:2412.03079*, 2024.

[71] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng, "Track anything: Segment anything meets videos," 2023.

[72] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion Revisited," in *Computer Vision and Pattern Recognition (CVPR)*, 2016. [Online]. Available: https://github.com/colmap/colmap

[73] Q. Ma, D. P. Paudel, A. Chhatkuli, and L. Van Gool, "Continuous pose for monocular cameras in neural implicit representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5291–5301.

[74] P. Truong, M.-J. Rakotosaona, F. Manhardt, and F. Tombari, "Sparf: Neural radiance fields from sparse and noisy poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4190–4200.

[75] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan, "Emergent correspondence from image diffusion," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=ypOiXjdfnU

[76] Y. Zheng, A. W. Harley, B. Shen, G. Wetzstein, and L. J. Guibas, "Pointodyssey: A large-scale synthetic dataset for long-term point tracking," in *ICCV*, 2023.

[77] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[78] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 331–20 341.

[79] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[80] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[81] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in

Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.

[82] 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*, 2021, pp. 5741–5751.

Chaoran Feng is a master student of Computer Science in Peking University. He pursued his undergraduate studies at the School of Future Technology, Dalian University of Technology, majoring in Artificial Intelligence. His research involves 3D reconstruction for large scale scene and sparse views, 3D generation, SLAM, and spiking neural network.

Jianbin Zhao is currently an undergraduate student at the School of Future Technology, Dalian University of Technology, majoring in Artificial Intelligence. His research interests include 3D reconstruction from sparse views and computer vision.

Zhenyu Tang is a second-year master's student at the School of Electronic and Computer Engineering, Peking University, under the supervision of Prof. Li Yuan. His research interests include multimodal understanding as well as the generation of 3D and 4D for objects and scenes.

Wangbo Yu is currently a Ph.D. student at the School of Computer Science, Peking University. He received a B.E. degree in telecommunications engineering from Xidian University in 2021. His research interests include low-level computer vision, 3D vision and Generative models.

Yuchen Li is currently a master's student at the School of Electronic and Computer Engineering, Peking University. He received a B.S. degree in Mathematical Finance from Tongji University. His research interests include computer vision and video generative models.

Li Yuan received the B.E. degree from University of Science and Technology of China, in 2017, and the PhD degree from National University of Singapore, in 2021. He is currently a tenure-track assistant professor with School of Electrical and Computer Engineering with Peking University. He has published more than 40 papers on top conferences/journals. His research interests include deep learning, image processing, and computer vision.

Yonghong Tian (Fellow, IEEE) is currently the Dean of the School of Electronics and Computer Engineering, a Boya Distinguished Professor with the School of Computer Science, Peking University, China, and the Deputy Director of the Artificial Intelligence Research, Peng Cheng Laboratory, Shenzhen, China. He is the author or coauthor of over 350 technical papers in refereed journals and conferences. His research interests include neuromorphic vision, distributed machine learning, and AI for science. He is a TPC Member of more than ten conferences, such as CVPR, ICCV, ACM KDD, AAAI, ACM MM, and ECCV. He is a Senior Member of CIE and CCF and a member of ACM. He was a recipient of the Chinese National Science Foundation for Distinguished Young Scholars in 2018, two National Science and Technology Awards, and three ministerial-level awards in China. He received the 2015 Best Paper Award for *EURASIP Journal on Image and Video Processing*, the Best Paper Award from IEEE BigMM 2018, and the 2022 IEEE SA Standards Medallion and SA Emerging Technology Award. He served as the TPC Co-Chair for BigMM 2015, the Technical Program Co-Chair for IEEE ICME 2015, IEEE ISM 2015, and IEEE MIPR 2018/2019, and the General Co-Chair for IEEE MIPR 2020 and ICME 2021. He was/is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from January 2018 to December 2021, IEEE TRANSACTIONS ON MULTIMEDIA from August 2014 to August 2018, *IEEE Multimedia Magazine* from January 2018 to August 2022, and IEEE ACCESS from January 2017 to December 2021. He co-initiated the IEEE International Conference on Multimedia Big Data (BigMM).