Charge: A Comprehensive Benchmark and Dataset for Dynamic Novel View Synthesis

Michal Nazarczuk Thomas Tanay Zhensong Zhang Eduardo Pérez-Pellitero
Huawei Noah's Ark Lab
[michal.nazarczuk1; thomas.tanay; zhangzhensong; e.perez.pellitero]@huawei.com

Figure 1. Charge is a new dataset for dynamic novel view synthesis characterised by high-quality renderings and rich annotations. The example includes left-to-right: object segmentation, normals, RGB, depth, and optical flow.

Abstract

This paper presents a new dynamic dataset for novel view synthesis, generated from a high-quality, animated film with stunning realism and intricate detail. Our dataset captures a variety of dynamic scenes, complete with detailed textures, lighting, and motion, making it ideal for training and evaluating cutting-edge 4D scene reconstruction and novel view generation models. In addition to high-fidelity RGB images, we provide multiple complementary modalities, including depth, surface normals, object segmentation and optical flow, enabling a deeper understanding of scene geometry and motion. The dataset is organised into three distinct benchmarking scenarios: a dense multi-view camera setup, a sparse camera arrangement, and monocular video sequences, enabling a wide range of experimentation and comparison across varying levels of data sparsity. With its combination of visual richness, high-quality annotations, and diverse experimental setups, this dataset offers a unique resource for pushing the boundaries of view synthesis and 3D vision. The dataset is available at https://charge-benchmark.github.io.

1. Introduction

Access and availability of high-quality, large-scale datasets for the training and evaluation of algorithms is arguably one of the main engines for progress and innovation in computer vision. With the advent of digital photography and the internet, an abundance of images readily available has been a good example of this: it unlocked unprecedented development and breakthroughs in a number of tasks, *e.g.* image classification, low-level vision, specially when leveraging forms of self-supervision that alleviate for the need of costly manual data annotation.

Recently, the seminal Neural Radiance Fields (NeRFs) by Mildenhall *et al.* [17] reconstructs the 3D opacity and radiance of a scene by fitting an implicit representation supervised via differentiable volumetric rendering and pho-

Table 1. A summary and comparison of datasets used in dynamic novel view synthesis. The top section includes datasets used for multi-view evaluation whereas the middle section focuses on monocular evaluation data. † - 2 camera rig with alternating frames assigned to training and test trajectory.

Dataset	Dense	Sparse	Mono	Depth	FPS	# seq	# train cams	# test cams	# frames
Neural 3D Video [11]	✓	✗	✗	✗	30	6	20	1	56 700
Technicolor [22]	✓	✗	✗	✗	30	5	15	1	25 696
Google Immersive [3]	✓	✗	✗	✗	30	7	45	1	157 320
Nvidia Dynamic Scene [30]	✓	✗	✗	✓	60	8	1	11	2 304
D-NeRF [21]	✗	✗	✓	✗	60	8	1	1	1 410
Nerfies [19]	✗	✗	✓	✗	5	4	1†	1†	1 680
HyperNeRF [20]	✗	✗	✓	✗	15	17	1†	1†	2 152
DyCheck [7]	✗	✗	✓	✓	60	7+7	1	2/0	8 746
Ours	✓	✓	✓	✓	96	8	25+9+4	16+10+16	185 600

tometric loss on a dense set of posed 2D images. NeRFs have an undeniable appeal due to the simplicity of their capture, *i.e*. coupled with *off-the-shelf* Structure-from-Motion (SfM) camera pose estimators such as COLMAP [23], they can reconstruct 3D from a handful of 2D images. Such paradigm is exciting as it can potentially overcome some of the hardware bottlenecks around capturing 3D data at scale thanks to its reliance on a single camera to obtain 2D imagery and SfM for the camera poses.

Differentiable rendering has received significant number of follow-up works, *e.g*. dense, sparse, varying appearance. Most of these assume static scenes and fixed radiance fields. Extending static to dynamic scene reconstruction is challenging. There has been recent interest in addressing this problem, with early paradigms extending NeRF [20, 21] or 3DGS [14, 28, 33] with temporal deformation.

Capturing datasets for novel view synthesis of dynamic scenes is inherently difficult. As opposed to their static counterpart, camera pose estimation on dynamic scenes is a challenging and largely unsolved problem, *i.e*. movement in the scene can introduce matching errors that then degrade the estimated camera poses. In practice, this means that to obtain multiple views of the scene complex, synchronized multi-camera systems are needed [12, 16]. Such connected rigs impose constrains on the distribution of the camera poses, restrict the motion diversity due to the limitations on physical spaces where they can be deployed (*i.e*. indoors, volumetric studios), and ultimately can not be pragmatically moved within the scene (in part due to weight, fragility and build, but also because of potential problems with camera calibration). The need for testing views results in a reduction of the number of training views available, which can result in poor view-sampling for evaluation. For approaches where separated, non-synchronized cameras are used for the testing views [7], problems can arise in the camera calibration in dynamic scenes. This, coupled with the lack of precise temporal frame alignment, degrades the ability to evaluate accurately the performance of 4D reconstruction methods.

Given the current technical limitations to capture dynamic 3D data, we find inspiration on the *MPI Sintel* dataset by Butler *et al*. [4]. In our work, we present a novel high-resolution, dynamic dataset for NVS with rich modalities (RGB, depth, normals, segmentation), and overcomplete training and testing camera coverage that addresses open challenges of dynamic 3D reconstruction. The dataset is arranged for different dynamic reconstruction set-ups, namely dense input, sparse input, and monocular input (with varying configurations of the camera trajectories, *e.g*. fast/slow, spline/random). We show a visualization of the modalities in Figure 1, and an overview of the dataset composition in Figure 2. This dataset is generated from the high-quality animated movie Charge including rich scene compositions and a variety of dynamic content. The synthetic nature of the dataset allows us to introduce a large-scale dataset not viable to be collected in the real world due to capture cost. We summarise our contributions as follows:

1. We introduce a new synthetic dataset - Charge - for dynamic novel view synthesis. It is characterised by high visual quality, it outscales all other available datasets in the number of provided images, and it provides rich annotations offering a variety of additional modalities.
2. We unify all relevant 4D reconstruction setups - Dense, Sparse, Mono, and provide benchmarking for all.
3. We perform an extensive evaluation of state-of-the-art reconstruction methods and analyse the results emphasising the importance of the benchmark we propose.

2. Related Work

Advances in novel view synthesis include the introduction of two seminal reconstruction paradigms, namely Neural Radiance Fields [17] and 3D Gaussian Splatting [9]. These developments in static scene reconstruction were quickly

Figure 2. An overview of the Charge dataset. The left side presents a selection of frames from all the animations. On the right side, a 3 setups included in the dataset are presented: Dense, Sparse, and Mono. Cameras allocation and sample movement path for monocular trajectory are overlayed on the section of the point cloud corresponding to scene 040_0040.

followed by several works on dynamic content.

Dynamic Reconstruction: NeRF-based methods for video reconstruction include D-NeRF [21], StreamRF [10], HexPlane [5], K-Planes [6], Tensor4D [24], MixVoxels [26]. Representative monocular approaches include NSFF [13], Nerfies [19], HyperNeRF [20], or DyCheck [7]. Similarly, Gaussian Splatting developments quickly sparked research on dynamic novel view synthesis. Multi-view videos were reconstructed by: GaussianFlow [15], 4D Gaussian Splatting [28], Spacetime Gaussians [14], MotionGS [33], SWinGS [25]. The most recent advances in Gaussian Splatting include monocular video reconstructions by works such as Deformable 3D Gaussians [29] or SC-GS [8].

Datasets: Current evaluation of multi-view dynamic novel view synthesis is based on a handful of datasets summarised in Table 1. This includes a selection of real-world captured data. Neural 3D Video [11] introduces a series of indoor cooking scenes captured with a static rig of GoPro cameras. Similarly, the Technicolor dataset [22] provides several scenes captured with a synchronized rig of 4×4 cameras. The shortcoming of the aforementioned two datasets is the reduced amount of dynamic content (*e.g.* cooking action occupies only a small middle region of the videos), and relatively slow motions. Additionally, those datasets evaluate the performance of the proposed approach only on a 1 camera subselected from the whole rig. Google Immersive [3] provides dynamic videos captured with a spherical rig of outward-facing cameras and suffers from heavy fish-eye distortions and very low overlap between views. DiVa-360 [16] captures a set of objects and hand-object interactions with a 360° setup. Monocular video datasets include *e.g.* Nvidia Dynamic Scene [30] which introduces sequences captured by a rig of 12 cameras subsampling alternating views to create a monocular trajectory (however aggressively temporally subsampled). Nerfies [19] and HyperNeRF [20] introduce a monocular video by capturing action with two cameras and alternating their frames in the input and test sequences. These datasets suffer from highly unrealistic motion due to teleporting cameras creating in this way pseudo-multi-view capture. This is characterised

as a high effective multi-view factor in DyCheck [7], which introduces a new monocular capture with 2 static cameras for evaluation. However, it suffers from an imperfect set of camera poses for the training trajectory. A better set of camera poses can be seen in DTU [1] dataset which was captured with a camera mounted on a robotic arm. It is broadly used in novel view synthesis evaluation in dense and sparse setups. However, it provides only static scenes. Similarly, Sparse Neural Rendering Dataset [18] provides a DTU-like setup but in a synthetic version, and thus is also static and present a single object per scene (no complex scenes). D-NeRF [21] is a synthetic dataset of dynamic content. It captures lower-quality subject assets on a white background from a set of cameras placed on a dome. Further, it creates a monocular input sequence by sampling consecutive cameras (suffering from teleporting-related issues). In contrast, Charge provides high-quality renderings of rich sequences with carefully designed setups for dense and sparse multi-view, and monocular captures.

3. The Charge Dataset

In this work we introduce a new dataset for dynamic novel view synthesis. In pursue of high-quality data, we explored a scene of animated movies created in the rendering software Blender [2]. Specifically, we focus on the latest release of Charge, a short movie set in a dystopian world, created to showcase Blender's photorealistic rendering capability.

3.1. Dataset Creation

We utilise production shots (short scenes provided with 3D assets) from the Charge movie to create our dataset. For each such scene a composition with animation, lighting, and a library of items is available. We processed all 8 available scenes for Charge of which examples can be seen in Figure 2 (left). For every scene, we manually picked places for cameras for the best capture and visibility of the scene after spanning the whole rig. Additionally, we removed the original rendering pipeline due to it containing postprocessing effects that may degrade the 3D consistency of the capture. Instead, we replace the rendering pipeline with direct rendering from the camera, and we add rendering of additional modalities (such as depth, or normals). Further, we notice that this action-heavy movie was developed in 24FPS, therefore, we subsample the animation 4 times.

3.2. Camera Setups

We believe that our Charge dataset provides a unique benchmarking quality of having various camera setups corresponding to commonly tackled novel view synthesis problems, namely: Dense, Sparse, and Mono. Commonly used camera setups include the use of coplanar capture or dome capture. Inspired by that, we decided to put our cameras on the section of the sphere, such that scenes are mostly forward-facing (similar to most real-world captures), yet offer the benefits of changing perspective. To this end, we manually adjust the position of the rig and the radius of the aforementioned sphere for each scene.

Dense setup consists of 25 training cameras and 16 testing cameras (see Fig. 2 top right). This offers a setup similar to Neural 3D Video [11] or Technicolor [22]. Those works, limited by the constraints of the real world (high capture cost), offer only one camera for testing. We provide a rig of 16 testing cameras allowing for more thorough evaluation.

Sparse setup follows a common practise in literature (e.g. pixelNeRF [31]) and we provide training scenarios including 3, 6, or 9 training cameras, accompanied by 10 testing views (see Fig.2 middle right). Note that the sparse setup covers the scenes from a different perspective instead of simply subsampling the training cameras. This increases the diversity of our dataset and opens up new areas of research and exploration when performing distant novel view synthesis, *i.e*. with an emphasis on view extrapolation as opposed to interpolation.

Mono setup introduces 4 different trajectories of the monocular camera. We believe that such a setup should contain humanly viable camera motions. As characterised by DyCheck, a monocular capture should not exhibit a high effective multiview factor (*e.g*. teleporting or extremely fast camera). To this end, based on empirical experimentation with the use of a phone camera and a target captured from around $1-2$m, we estimate that the reasonable velocity of a capture device lies around the range of $15-50$cm/s. Therefore, we introduce *Fast* and *Slow* monocular trajectories corresponding roughly to the upper and lower bounds of the measured values. Similarly, we propose that the capture may deliberately try to cover a wider perspective or be more unregularised. Hence, we suggest two following qualities of the camera trajectory: *Spline* - constrained to a spline curve with controlling nodes spanning the majority of corresponding train cameras positions, and *Random Walk* - randomly generated direction for every time step with a smoothing factor. This results in the following training scenarios: *SplineFast*, *SplineSlow*, *RandomWalkFast*, *RandomWalkSlow*. Further, monocular reconstruction encompasses many downstream goals, such as stabilisation, or spatial video (stereo pair synthesis). Thus, we propose to test each of these trajectories with 4 static cameras chosen from the set of Dense testing cameras (allowing for direct comparison), and 4 cameras relative to the training one (different baselines, and a camera orbiting around).

3.3. Data qualities

As visible on the left-hand side of Figure 2, Charge introduces a large diversity of scenes posing challenging and interesting problems for 3D reconstruction methods. Examples include a scene with large open bounds and a smaller

Table 2. Quantitative evaluation results of Charge dataset - Dense and Mono setups.

Method	PNSR	PNSR-D	PNSR-S	SSIM	SSIM-D	LPIPS	LPIPS-D	FOV_O
Dense								
4DGS [28]	28.94	26.84	31.85	0.881	0.848	0.231	0.307	0.70
STG [14]	29.29	28.15	31.33	0.886	0.866	0.193	0.263	0.70
Mono - Spline Fast								
4DGS [28]	24.85	22.69	27.25	0.840	0.794	0.264	0.352	0.42
D-3DGS [29]	24.88	22.91	27.24	0.869	0.807	0.201	0.295	0.42
SC-GS [8]	23.97	23.57	25.89	0.847	0.798	0.217	0.291	0.42
Mono - Spline Slow								
4DGS [28]	24.18	21.95	26.65	0.834	0.787	0.266	0.360	0.41
D-3DGS [29]	24.18	22.53	26.15	0.858	0.794	0.203	0.290	0.41
SC-GS [8]	23.29	22.54	25.15	0.843	0.786	0.225	0.304	0.41
Mono - Random Walk Fast								
4DGS [28]	24.78	22.79	26.81	0.835	0.790	0.267	0.359	0.41
D-3DGS [29]	24.34	22.55	26.24	0.852	0.789	0.217	0.309	0.41
SC-GS [8]	22.90	23.04	24.22	0.822	0.775	0.243	0.314	0.41
Mono - Random Walk Slow								
4DGS [28]	23.38	21.68	25.64	0.818	0.771	0.277	0.370	0.38
D-3DGS [29]	22.85	21.61	24.76	0.829	0.762	0.227	0.318	0.38
SC-GS [8]	21.86	21.73	23.51	0.811	0.762	0.244	0.318	0.38

dynamic subject (020_0020), with a large amount of movement (040_0040), with a large area of dynamic content (070_0123), or highly non-rigid elements (050_0130).

With the development of 3DGS, the high-resolution evaluation became viable and is arguably preferable. Thus, we render our dataset in a resolution of **2048×858**. Further, we provide a high-speed capture in **96FPS**. Additionally, with the use of synthetic rendering for data collection, we are able to provide highly precise camera poses in contrast to real-world captures that have to rely on estimations. Finally, along with *RGB images*, we supply **metric depth, optical flow, normals, UV maps, object segmentation, and dynamic content masks**.

In total, Charge offers **8** diverse scenes, each rendered with **25+16** dense setup cameras, **9+10** sparse setup cameras, and **4+16** monocular setup cameras. This accounts for a total of **185 600** frames available, providing a larger scale of data than real-world datasets (see Tab. 1).

4. Benchmark

We evaluate the Charge dataset with a selection of methods representing the state-of-the-art in multi-view and monocular Gaussian Splatting reconstruction. We report a large selection of metrics, introducing a measure of difficulty for each task, and analyse the results focusing on challenges available in our dataset that emphasise problems that are difficult for current methods to deal with.

4.1. Metrics

In our evaluation, we report commonly used metrics, such as PSNR, SSIM [27], and LPIPS [32]. Further, for PSNR, we include the metric calculated within the mask of dynamic and static content (PSNR-D and PSNR-S). Similarly, for SSIM and LPIPS we report a version of these metrics calculated for dynamic content based on a tight bounding box of the dynamic mask (SSIM-D, LPIPS-D).

Additionally, we propose to quantify the difficulty of each task. To this end, we suggest calculating a parameter indicating an overlap between the field of views of the testing camera with respect to training cameras. To this end, for each testing view, we reproject the image plane (four corners of the image) into all training views obtaining a coverage mask - m_i (effectively saying which area of the testing view is visible from the given training view). Further, we sum up all such created masks and normalise them by the number of training views $\#Tr$ obtaining a weighted covisibility proxy (where pixels seen by more training views have a value closer to 1). Therefore, for a given testing view, field

Table 3. Quantitative evaluation results of Charge dataset - Sparse setup.

Method	Views	PNSR	PNSR-D	PNSR-S	SSIM	SSIM-D	LPIPS	LPIPS-D	FOV_O
Sparse									
4DGS [28]	**3**	19.71	20.71	20.28	0.776	0.740	0.358	0.416	0.54
	6	23.93	24.89	24.29	0.840	0.810	0.277	0.338	0.62
	9	26.67	26.51	27.45	0.874	0.840	0.226	0.301	0.64
STG [14]	**3**	18.80	20.15	19.04	0.753	0.721	0.371	0.418	0.54
	6	22.39	23.61	22.57	0.820	0.790	0.295	0.361	0.62
	9	24.52	24.96	25.03	0.850	0.815	0.260	0.347	0.64

of view overlap is expressed as:

$$FOV_O = \frac{\sum_{\#Tr} m_i}{\#Tr \cdot \#pixels} \tag{1}$$

Finally, we average the value for all the test views in the given task to obtain an indication of task difficulty. Note, that for the reprojection, one needs to use depth. In our experiments, we use median depth for the given image. We find that using real depth values for the projected point, using median depth, or using a sensibly chosen handcrafted value does not result in significant value deviations for FOV_O. A more detailed analysis, along with the breakdown of the metrics, including separation per scene is available in the Supplementary Material.

4.2. Results

We provide evaluation for all 3 setups in our dataset, namely Dense, Sparse, and Mono. Therefore, we use 4D Gaussian Splatting [28] to train all sequences in all setups, as it is a method suitable for both multi- and single-view reconstruction. Further, we use Spacetime Gaussians [14] to evaluate Dense and Sparse scenarios. Finally, in a monocular setting, we add 2 dedicated methods, Deformable 3D Gaussians [29] and SC-GS [8]. All methods were trained on full-length sequences and evaluated in full resolution of data in Charge (2048×858).

4.3. Dense

The results of the experiments on the Dense setup of the Charge dataset are presented in Table 2 (top rows). Additionally, selected qualitative samples can be seen in Figure 3. We can observe PSNR results of 28.94, and 29.29 for 4DGS and STG respectively. This indicates that Charge poses a good challenge for current state-of-the-art methods. In Fig. 3 (left side) we can see that both methods struggle with reconstructing a splash of paint which leads us to believe that even in the dense multi-view camera setup, the reconstruction of the dynamics of a highly non-rigid object is yet to be explored in depth. Further, on the example of scene 040_0040 (fig. 3 right), we note that our dataset provides a good way of evaluating the effect of the length of the sequence on the reconstruction (with sequences ranging from 93 to 653 frames). The given example shows that 4DGS struggles with reconstructing a long-range sequence with a large amount of movement (note the character coming into a frame). Notably, however, we do not notice a decrease in the performance with the consecutive frames but rather a consistently lower quality of the reconstruction. Finally, unsurprisingly, we can observe the gap between the reconstruction of dynamic and static regions which is reflected in the presented results. This leads us to believe that data like

Figure 3. Example results of rendering in Charge dataset evaluation - Dense setup. The top and bottom constitute a pair of first and last frames from the same scene. Best viewed zoomed in.

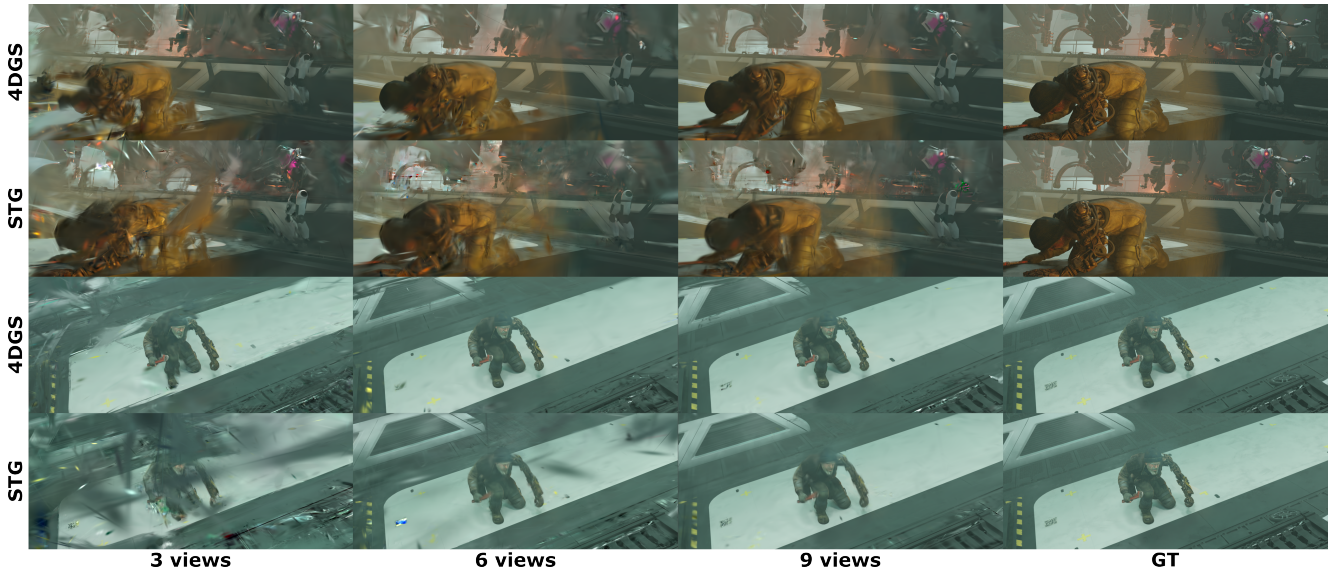

Figure 4. Example results of rendering in Charge dataset evaluation - Sparse setup. Top rows - scene 050_0160, bottom - 060_0100. Best viewed zoomed in.

Figure 5. Example results of rendering in Charge dataset evaluation - Mono setup, selected trajectory types. Test cameras left to right: static camera, rig camera - small baseline, rig camera - orbiting around train, static camera. Best viewed zoomed in.

Charge with a large amount of dynamic content is crucial for an insightful evaluation.

4.4. Sparse

In Table 3 we present the results of the evaluation of 4DGS and STG on the Sparse setup of the Charge dataset. Note that as indicated in Figure 2 (right), the sparse camera setup does not overlap with the dense one. Therefore, the corresponding numerical values should not be directly compared. However, based on the values of the field of view overlap, we notice that sparse rendering is a significantly harder task than dense reconstruction. Additionally, we may note that the difficulty increase from 6 to 3 training views is higher than from 9 to 6 training views. An example selection of scenes trained and rendered in all scenarios (with 3, 6, and 9 input views) is shown in Figure 4. We can observe that in the lower camera regime, 4DGS performs slightly better than STG, indicating that the selection of the best method for the task is configuration-dependent. Further, we can clearly observe, both visually, and in the presented metrics, a drastic difference in performance with respect to the number of training cameras. In Figure 4 we can see that even

in 9-views case, the renderings exhibit a lack of fine details (*e.g*. hands in the top row, face in the bottom row). Further, with 6 training views, we notice an increasing number of background missing or incorrect gaussian. Finally, 3 views present a very hard case for the benchmarked methods leading to very artifact-filled renders. Interestingly, the differences in the reconstruction of dynamic and static content are not as prominent in the Sparse setup as they were in the Dense one (to the extent of dynamic content being reconstructed better than static in the 6-view case). We believe that the decrease in the overlap between cameras leads to higher uncertainties, especially in disoccluded areas, as well as edge areas, seen by a very low number of cameras.

4.5. Mono

We propose 4 configurations for evaluating monocular trajectories and provide evaluation separated as such, namely *Spline Fast*, *Spline Slow*, *Random Walk Fast*, *Random Walk Slow*. The results from all configurations are presented in Table 2 (bottom rows). A selection of qualitative results for the Mono setup is shown in Fig. 5. This setup is evaluated based on 8 cameras - 4 cameras correspond to central testing cameras from the Dense setup, and 4 further cameras are moving together with the training one (3 represent different baselines in the task of creating stereo pairs, and 1 camera is orbiting around the training one).

Unsurprisingly, we can notice a decrease in field of view overlap (increase in difficulty) for this task when compared to the dense setup. We see slight differences between the configurations of the Mono setup, indicating structured (spline covering the area better) and faster (more area covered) motion to be easier to reconstruct.

For all tested methods, we notice a similar performance, with a slightly lower for SC-GS. We believe this may be caused by the fact that SC-GS to provide controllability to the scenes, describes the motion with a relatively low number of parameters. This may cause the performance to suffer when evaluated on our dataset with a large amount of motion. Further, we note that the performance between configurations is similar and preserves relative ordering with respect to difficulty (based on FOV_O). In Fig. 3 we notice that when reconstructing a scene with a large amount of dynamic content close to the camera (*e.g*. 070_0123) the methods tend to struggle with the reconstruction of such regions. Similarly, as seen for 010_0050, we notice how monocular reconstruction is affected by a lack of coverage outside the field of view causing dark areas near the edges.

In Table 4 we provide additional results for the Mono setup separated into static cameras, and cameras moving together with the training camera (Rig cameras). We can observe a consistently higher performance of the rig cameras. This shows that current methods for monocular reconstruction provide much higher quality results in the local neighbourhood of the training view. We believe including both types of target cameras provides a fuller overview of the performance of the method. Qualitative results in Fig. 5 coincide with intuition regarding static and rig cameras. Namely, stereo reconstruction (060_0130) is the most straightforward task, static cameras pose a bigger challenge (070_0123, 020_0020) similar to a camera orbiting around the training one (010_0050) notably due to change in relative angle to the training view.

Table 4. Quantitative evaluation results of static and rig cameras in the Mono setup.

	4DGS		D-3DGS		SC-GS	
	PSNR	PSNR-D	PSNR	PSNR-D	PSNR	PSNR-D
Rig cameras						
SF	26.48	24.24	26.60	24.38	26.07	25.25
SS	25.84	23.53	25.81	24.16	25.16	24.10
WF	26.63	24.58	26.09	24.26	24.99	24.80
WS	24.80	22.96	24.19	22.80	23.28	22.95
Static cameras						
SF	23.23	21.14	23.15	21.45	21.88	21.88
SS	22.53	20.38	22.55	20.90	21.42	20.99
WF	22.93	21.00	22.59	20.85	20.80	21.29
WS	21.95	20.40	21.52	20.42	20.44	20.51

5. Conclusions

In this paper, we introduced Charge - a new dataset and benchmark for dynamic novel view synthesis. Our dataset was synthetically rendered from a professionally created animated movie. This enables us to introduce rich data, with detailed shapes and textures as well as intricate movements and interactions. Charge offers several advantages over real-world datasets. We provide various capture setups for different problems (dense and sparse multi-view, and monocular reconstruction) with more cameras, and with more accurate pose annotation increasing the quality of evaluation. Charge includes rich annotations including various modalities enabling the development of new methods.

We extensively tested the performance of the main dynamic Gaussian Splatting-based methods on the Charge dataset. In our experiments, we provided a thorough analysis of the selected methods in all proposed setups (Dense, Sparse, Mono). Our evaluation emphasises the shortcomings of current approaches by specifically showing areas that need improvement. Finally, the performance of the state-of-the-art methods in the Charge benchmark shows that the data we propose is non-trivial and not less challenging than real-world captures, therefore, worth including in training and evaluation protocols in dynamic novel view synthesis research.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-Scale Data for Multiple-View Stereopsis. *IJCV*, pages 1–16, 2016. 4

[2] Blender Online Community. Blender. accessed on 29th August 2024. 4

[3] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew DuVall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive Light Field Video with a Layered Mesh Representation. *ACM TOG*, 39(4):86:1–86:15, 2020. 2, 3

[4] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2

[5] Ang Cao and Justin CV. HexPlane: A Fast Representation for Dynamic Scenes. In *CVPR*, 2023. 3

[6] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. In *CVPR*, 2023. 3

[7] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular Dynamic View Synthesis: A Reality Check. In *NeurIPS*, 2022. 2, 3, 4

[8] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. SC-GS: Sparse-Controlled Gaussian Splatting for Editable Dynamic Scenes. In *CVPR*, 2023. 3, 5, 6, 4

[9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM TOG*, 42(4), 2023. 2

[10] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming Radiance Fields for 3D Video Synthesis. In *NeurIPS*, 2022. 3

[11] Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3D Video Synthesis from Multi-view Video. In *CVPR*, 2022. 2, 3, 4

[12] Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard A. Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. In *CVPR*, 2022. 2

[13] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *CVPR*, 2021. 3

[14] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime Gaussian Feature Splatting for Real-Time Dynamic View Synthesis. In *CVPR*, 2024. 2, 3, 5, 6

[15] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-Flow: 4D Reconstruction with Dynamic 3D Gaussian Particle. In *CVPR*, 2024. 3

[16] Cheng-You Lu, Peisen Zhou, Angela Xing, Chandradeep Pokhariya, Arnab Dey, Ishaan N Shah, Rugved Mavidipalli, Dylan Hu, Andrew Comport, Kefan Chen, and Srinath Sridhar. Diva-360: The dynamic visual dataset for immersive neural fields. In *CVPR*, 2024. 2, 3

[17] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[18] Michal Nazarczuk, Thomas Tanay, Sibi Catley-Chandar, Richard Shaw, Radu Timofte, and Eduardo Pérez-Pellitero. AIM 2024 Sparse Neural Rendering Challenge: Dataset and Benchmark. In *ECCVW*, 2024. 4

[19] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. *ICCV*, 2021. 2, 3

[20] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: a higher-dimensional representation for topologically varying neural radiance fields. *ACM TOG*, 40(6), 2021. 2, 3

[21] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2021. 2, 3, 4

[22] Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, et al. Dataset and Pipeline for Multi-view Light-Field Video. In *CVPRW*, 2017. 2, 3, 4

[23] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 2

[24] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4D: Efficient Neural 4D Decomposition for High-fidelity Dynamic Reconstruction and Rendering. In *CVPR*, 2023. 3

[25] Richard Shaw, Michal Nazarczuk, Jifei Song, Arthur Moreau, Sibi Catley-Chandar, Helisa Dhamo, and Eduardo Pérez-Pellitero. SWinGS: Sliding Windows for Dynamic 3D Gaussian Splatting. In *ECCV*, 2024. 3

[26] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, and Huaping Liu. Mixed Neural Voxels for Fast Multi-view Video Synthesis. In *ICCV*, 2023. 3

[27] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4), 2004. 5

[28] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. In *CVPR*, 2024. 2, 3, 5, 6, 4

[29] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. In *CVPR*, 2024. 3, 5, 6, 4

[30] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*, 2020. 2, 3

[31] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images. In *CVPR*, 2021. 4

[32] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018. 5

[33] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. MotionGS: Exploring Explicit Motion Guidance for Deformable 3D Gaussian Splatting. In *NeurIPS*, 2024. 2, 3

Charge: A Comprehensive Benchmark and Dataset for Dynamic Novel View Synthesis

Supplementary Material

6. Field of View Overlap

In this section, we extend the analysis of the proposed field of view overlap parameter. In the main manuscript, we propose to calculate the projection of a test image plane into training views and calculate FOV_O based on a weighted average across all corresponding training views. This serves as an estimation of camera coverage between test and training views and can be treated as a proxy for the difficulty of reconstruction.

Notably, the value of such a parameter is dependent on the depth in which we place the projection plane. Here, we consider 3 possible scenarios:

- ground truth depth values for corners of the image plane,
- median value of the true depth in the whole image,
- reasonably chosen custom value (here, we set it to $5m$ for all scenes).

In Table 5, we present the values of the field of view overlap for the presented experiments setups for all the aforementioned ways of calculation. We can observe that in a majority of cases, all values for the given scene are similar for all 3 methods. Additionally, in most cases, the relative ranking is preserved. We note that the main discrepancy can be seen for scene 070_0123 in Dense and Sparse setups. This is caused by the fact that this scene is mainly covered by a closeup of an action scene occupying the centre of the field of view. Arguably, in such a case the median value of depth should be more representative with respect to the visibility of pixels between the views.

In summary, we argue that the proposed field of view overlap (FOV_O) metric is a good indicator of spatial covisibility between test and training views. Additionally, it is not highly sensitive to the used estimation of depth making it available for use with no ground truth depth available.

Table 5. Field of view overlap values calculated per scene for each experiment setup (values for all Mono setups averaged).

	Scene	Real	Median	Custom
Dense	010_0050	0.63	0.67	0.67
	020_0020	0.76	0.80	0.91
	040_0040	0.63	0.85	0.85
	050_0130	0.42	0.43	0.45
	050_0160	0.55	0.59	0.60
	060_0100	0.68	0.73	0.75
	060_0130	0.63	0.68	0.68
	070_0123	0.37	0.84	0.93
Sparse - 3	010_0050	0.50	0.51	0.51
	020_0020	0.62	0.63	0.71
	040_0040	0.55	0.61	0.61
	050_0130	0.34	0.34	0.36
	050_0160	0.53	0.54	0.55
	060_0100	0.61	0.64	0.64
	060_0130	0.51	0.52	0.52
	070_0123	0.32	0.62	0.73
Sparse - 6	010_0050	0.59	0.60	0.60
	020_0020	0.72	0.73	0.81
	040_0040	0.65	0.70	0.70
	050_0130	0.37	0.38	0.40
	050_0160	0.61	0.62	0.63
	060_0100	0.72	0.74	0.74
	060_0130	0.59	0.59	0.60
	070_0123	0.34	0.71	0.84
Sparse - 9	010_0050	0.59	0.60	0.60
	020_0020	0.74	0.76	0.86
	040_0040	0.66	0.73	0.73
	050_0130	0.39	0.40	0.42
	050_0160	0.61	0.64	0.64
	060_0100	0.72	0.76	0.76
	060_0130	0.61	0.62	0.62
	070_0123	0.37	0.74	0.88
Mono	010_0050	0.34	0.36	0.36
	020_0020	0.43	0.45	0.49
	040_0040	0.43	0.48	0.48
	050_0130	0.39	0.39	0.39
	050_0160	0.36	0.38	0.38
	060_0100	0.40	0.42	0.42
	060_0130	0.39	0.41	0.41
	070_0123	0.36	0.48	0.49

7. Detailed Results

In addition, we present detailed results of all experiments. In Table 6 we show per-scene results for Dense setup. Interestingly, the hardest scene for the benchmarked methods was 040_0040 which is the longest scene that also contains a large amount of movement, specifically a character moving towards the camera rig. Similarly, 070_0123 poses a significant challenge, it is a scene with a close-up interaction between a humanoid robot and a human in which the dynamic content covers a large part of the field of view and introduces a lot of occluded parts between training views.

Table 7 shows results for Sparse setup - configurations with 3, 6, 9 input views. Notably, it shows a superior performance of 4DGS over STG for almost all scenes across all configurations which stands in contrast to results in the

Table 6. Quantitative evaluation of Charge dataset - results per scene for Dense setup.

Scene	Method	PNSR	PNSR-D	PNSR-S	SSIM	SSIM-D	LPIPS	LPIPS-D	FOV_O
Dense									
010_0050	4DGS [28]	31.62	33.32	31.33	0.946	0.938	0.151	0.203	0.669
	STG [14]	32.40	34.34	32.04	0.950	0.945	0.130	0.161	0.669
020_0020	4DGS [28]	27.14	28.74	27.13	0.708	0.633	0.412	0.580	0.797
	STG [14]	26.34	31.54	26.31	0.708	0.689	0.282	0.489	0.797
040_0040	4DGS [28]	23.16	22.63	24.07	0.775	0.737	0.379	0.436	0.849
	STG [14]	24.45	25.20	24.47	0.813	0.794	0.264	0.284	0.849
050_0130	4DGS [28]	31.35	24.04	41.68	0.949	0.884	0.140	0.300	0.433
	STG [14]	32.92	25.96	40.78	0.954	0.901	0.129	0.265	0.433
050_0160	4DGS [28]	33.63	32.50	33.96	0.936	0.922	0.154	0.173	0.592
	STG [14]	32.19	31.24	32.41	0.934	0.919	0.138	0.171	0.592
060_0100	4DGS [28]	33.43	28.87	34.86	0.944	0.915	0.145	0.250	0.732
	STG [14]	34.17	31.15	34.92	0.948	0.926	0.118	0.192	0.732
060_0130	4DGS [28]	25.56	20.50	31.56	0.891	0.865	0.266	0.302	0.680
	STG [14]	27.91	23.13	32.19	0.907	0.888	0.203	0.244	0.680
070_0123	4DGS [28]	25.62	24.14	30.18	0.900	0.890	0.202	0.215	0.840
	STG [14]	23.93	22.60	27.54	0.877	0.866	0.281	0.301	0.840

Dense setup.

Finally, Tables 8 and 9 present results for Mono setup in Spline Fast and Slow, and Random Walk Fast and Slow configurations respectively. Not surprisingly, the *Slow* setups cover smaller paths and provide a harder challenge for the benchmarked methods.

8. Rendering setup

Charge was rendered in Blender 4.0.2 with the use of Cycles as the rendering engine. The most important settings include a resolution of 2048×858, 256 samples for path tracing, a camera with a focal length of 25mm or 20mm (depending on the scene), and a sensor size of 25mm (original Charge camera settings).

Setup: The scenes were rendered on servers each equipped with 4 Nvidia V100 GPUs and a 16-cores (32 threads) of Intel Xeon 8160 CPU. The equivalent rendering time of the whole dataset with one unit of this setup is equal to roughly 1 year.

License: Blender Open Movies is a subscription library offering movie production assets under a Creative Commons Attribution 4.0 license. This allows for modification and redistribution under the attribution condition. Charge creators can be found at `https://studio.blender.org/projects/charge/pages/credits/` and are also linked at the dataset download website.

Table 7. Quantitative evaluation of Charge dataset - results per scene for Sparse setup.

	Method	Views	PNSR	PNSR-D	PNSR-S	SSIM	SSIM-D	LPIPS	LPIPS-D	FOV_O
Sparse										
010_0050	4DGS [28]	**3**	19.29	27.11	18.30	0.847	0.869	0.295	0.307	0.514
		6	22.70	31.28	21.72	0.891	0.913	0.222	0.237	0.595
		9	28.54	31.81	27.97	0.927	0.924	0.201	0.264	0.601
	STG [14]	**3**	18.93	27.19	17.91	0.845	0.863	0.302	0.311	0.514
		6	21.73	31.17	20.66	0.884	0.906	0.251	0.270	0.595
		9	23.48	32.04	22.47	0.901	0.915	0.225	0.259	0.601
020_0020	4DGS [28]	**3**	23.50	26.52	23.47	0.590	0.557	0.421	0.559	0.634
		6	25.45	27.62	25.43	0.663	0.596	0.410	0.591	0.734
		9	26.39	28.98	26.37	0.699	0.623	0.389	0.583	0.762
	STG [14]	**3**	20.14	24.72	20.11	0.461	0.474	0.393	0.433	0.634
		6	22.84	27.11	22.81	0.583	0.561	0.354	0.438	0.734
		9	24.80	28.30	24.78	0.658	0.609	0.354	0.525	0.762
040_0040	4DGS [28]	**3**	16.95	15.81	17.70	0.680	0.632	0.452	0.494	0.610
		6	21.44	21.79	21.36	0.786	0.757	0.318	0.355	0.704
		9	23.60	24.29	23.59	0.828	0.799	0.253	0.301	0.731
	STG [14]	**3**	17.94	18.86	17.63	0.731	0.703	0.402	0.425	0.610
		6	21.08	23.25	20.35	0.800	0.774	0.313	0.350	0.704
		9	22.35	24.47	21.71	0.819	0.794	0.283	0.333	0.731
050_0130	4DGS [28]	**3**	24.40	18.83	27.93	0.903	0.811	0.192	0.349	0.340
		6	28.99	24.39	31.38	0.938	0.881	0.140	0.248	0.375
		9	30.07	25.36	32.74	0.944	0.888	0.133	0.244	0.398
	STG [14]	**3**	24.00	19.08	26.72	0.896	0.809	0.203	0.362	0.340
		6	27.38	22.23	30.21	0.925	0.855	0.175	0.321	0.375
		9	29.49	23.14	34.20	0.936	0.871	0.159	0.300	0.398
050_0160	4DGS [28]	**3**	18.55	24.89	18.32	0.812	0.806	0.300	0.303	0.543
		6	23.95	29.34	23.75	0.877	0.870	0.190	0.202	0.624
		9	26.56	29.93	26.43	0.913	0.891	0.139	0.186	0.636
	STG [14]	**3**	16.82	20.01	16.68	0.793	0.739	0.356	0.408	0.543
		6	22.91	24.40	22.89	0.884	0.832	0.198	0.290	0.624
		9	26.22	25.73	26.36	0.913	0.854	0.159	0.275	0.636
060_0100	4DGS [28]	**3**	18.56	19.38	18.63	0.812	0.774	0.355	0.394	0.635
		6	26.08	24.52	26.61	0.885	0.844	0.268	0.343	0.737
		9	29.01	26.47	29.92	0.911	0.874	0.199	0.277	0.756
	STG [14]	**3**	18.06	18.81	18.03	0.781	0.745	0.403	0.428	0.635
		6	22.80	22.68	22.86	0.849	0.820	0.286	0.351	0.737
		9	25.48	24.52	25.71	0.877	0.846	0.239	0.319	0.756
060_0130	4DGS [28]	**3**	17.70	14.21	19.10	0.770	0.691	0.442	0.502	0.520
		6	20.76	17.27	22.80	0.822	0.767	0.377	0.425	0.594
		9	23.06	18.05	27.07	0.871	0.823	0.287	0.345	0.622
	STG [14]	**3**	16.32	14.25	16.96	0.728	0.661	0.488	0.543	0.520
		6	18.94	16.06	20.00	0.787	0.729	0.417	0.491	0.594
		9	20.55	16.94	22.04	0.820	0.763	0.364	0.461	0.622
070_0123	4DGS [28]	**3**	18.74	18.90	18.80	0.794	0.783	0.406	0.418	0.621
		6	22.09	22.95	21.25	0.857	0.847	0.295	0.304	0.705
		9	26.12	27.15	25.51	0.901	0.895	0.204	0.208	0.744
	STG [14]	**3**	18.22	18.29	18.27	0.786	0.776	0.418	0.429	0.621
		6	21.43	21.97	20.78	0.847	0.839	0.363	0.373	0.705
		9	23.79	24.55	22.99	0.877	0.870	0.293	0.303	0.744

Table 8. Quantitative evaluation of Charge dataset - results per scene for Mono - Spline Fast and Mono - Spline Slow.

Scene	Method	PNSR	PNSR-D	PNSR-S	SSIM	SSIM-D	LPIPS	LPIPS-D	FOV_O
Mono - Spline Fast									
010_0050	4DGS [28]	28.88	29.76	28.77	0.926	0.916	0.158	0.218	0.318
	D-3DGS [29]	28.36	30.98	28.03	0.932	0.924	0.110	0.141	0.318
	SC-GS [8]	22.64	32.15	21.59	0.902	0.912	0.153	0.156	0.318
020_0020	4DGS [28]	26.32	28.15	26.31	0.667	0.606	0.421	0.598	0.413
	D-3DGS [29]	27.84	27.67	27.89	0.816	0.633	0.167	0.400	0.413
	SC-GS [8]	26.41	26.93	26.43	0.727	0.586	0.217	0.468	0.413
040_0040	4DGS [28]	18.68	17.58	19.81	0.759	0.723	0.365	0.411	0.467
	D-3DGS [29]	20.25	20.19	20.52	0.792	0.766	0.358	0.383	0.467
	SC-GS [8]	21.18	21.56	21.23	0.800	0.780	0.333	0.347	0.467
050_0130	4DGS [28]	26.19	19.58	33.21	0.910	0.799	0.177	0.373	0.402
	D-3DGS [29]	24.94	18.34	31.79	0.909	0.792	0.189	0.426	0.402
	SC-GS [8]	25.48	18.92	31.95	0.906	0.784	0.156	0.336	0.402
050_0160	4DGS [28]	25.52	23.39	25.85	0.866	0.820	0.201	0.252	0.335
	D-3DGS [29]	25.58	22.55	26.19	0.882	0.810	0.190	0.282	0.335
	SC-GS [8]	27.21	26.99	27.37	0.893	0.864	0.177	0.205	0.335
060_0100	4DGS [28]	29.57	25.46	30.99	0.924	0.880	0.157	0.268	0.426
	D-3DGS [29]	25.88	23.92	26.95	0.893	0.856	0.193	0.260	0.426
	SC-GS [8]	25.70	24.33	26.33	0.891	0.847	0.194	0.262	0.426
060_0130	4DGS [28]	21.65	17.21	25.04	0.835	0.801	0.383	0.427	0.354
	D-3DGS [29]	21.78	16.81	26.47	0.865	0.830	0.236	0.290	0.354
	SC-GS [8]	21.42	17.45	24.53	0.820	0.799	0.289	0.325	0.354
070_0123	4DGS [28]	22.01	20.42	28.05	0.831	0.808	0.251	0.270	0.476
	D-3DGS [29]	24.40	22.84	30.04	0.863	0.846	0.169	0.173	0.476
	SC-GS [8]	21.75	20.20	27.72	0.832	0.810	0.216	0.229	0.476
Mono - Spline Slow									
010_0050	4DGS [28]	23.00	26.09	22.40	0.882	0.884	0.220	0.271	0.351
	D-3DGS [29]	22.87	28.15	22.02	0.888	0.889	0.152	0.176	0.351
	SC-GS [8]	22.03	29.94	21.00	0.895	0.899	0.169	0.169	0.351
020_0020	4DGS [28]	25.69	26.10	25.71	0.638	0.574	0.428	0.615	0.428
	D-3DGS [29]	27.20	25.24	27.27	0.807	0.586	0.150	0.383	0.428
	SC-GS [8]	26.10	27.14	26.10	0.757	0.595	0.189	0.430	0.428
040_0040	4DGS [28]	18.91	17.74	20.11	0.762	0.722	0.342	0.395	0.468
	D-3DGS [29]	19.76	19.17	20.43	0.783	0.755	0.352	0.386	0.468
	SC-GS [8]	19.89	19.14	20.69	0.785	0.755	0.342	0.375	0.468
050_0130	4DGS [28]	26.83	19.95	34.24	0.918	0.808	0.171	0.385	0.357
	D-3DGS [29]	25.25	18.76	31.52	0.908	0.786	0.177	0.397	0.357
	SC-GS [8]	25.04	18.29	31.85	0.908	0.776	0.164	0.375	0.357
050_0160	4DGS [28]	28.98	25.49	29.71	0.912	0.856	0.170	0.223	0.353
	D-3DGS [29]	25.88	24.29	26.32	0.865	0.819	0.174	0.230	0.353
	SC-GS [8]	24.64	24.92	24.75	0.865	0.832	0.199	0.226	0.353
060_0100	4DGS [28]	28.55	24.92	29.89	0.921	0.878	0.180	0.295	0.398
	D-3DGS [29]	27.23	25.46	27.90	0.898	0.857	0.176	0.234	0.398
	SC-GS [8]	26.04	23.53	26.76	0.886	0.831	0.192	0.262	0.398
060_0130	4DGS [28]	21.07	16.46	24.66	0.833	0.791	0.333	0.390	0.377
	D-3DGS [29]	21.90	17.00	26.26	0.866	0.828	0.248	0.304	0.377
	SC-GS [8]	21.02	17.24	23.70	0.821	0.795	0.313	0.342	0.377
070_0123	4DGS [28]	20.44	18.88	26.51	0.809	0.783	0.287	0.311	0.475
	D-3DGS [29]	23.39	22.14	27.44	0.852	0.834	0.199	0.208	0.475
	SC-GS [8]	21.55	20.15	26.33	0.824	0.803	0.236	0.251	0.475

Table 9. Quantitative evaluation of Charge dataset - results per scene for Mono - Random Walk Fast and Mono - Random Walk Slow.

Scene	Method	PNSR	PNSR-D	PNSR-S	SSIM	SSIM-D	LPIPS	LPIPS-D	FOV_O
Mono - Walk Fast									
010_0050	4DGS [28]	26.84	28.39	26.52	0.922	0.914	0.168	0.228	0.370
	D-3DGS [29]	28.74	29.68	28.83	0.935	0.919	0.114	0.151	0.370
	SC-GS [8]	22.44	31.79	21.32	0.899	0.914	0.147	0.145	0.370
020_0020	4DGS [28]	26.46	28.04	26.45	0.671	0.612	0.399	0.597	0.482
	D-3DGS [29]	27.35	27.74	27.36	0.807	0.632	0.149	0.383	0.482
	SC-GS [8]	26.00	27.63	25.98	0.742	0.599	0.183	0.389	0.482
040_0040	4DGS [28]	20.77	19.64	22.06	0.772	0.734	0.335	0.382	0.486
	D-3DGS [29]	21.29	21.20	21.72	0.801	0.778	0.332	0.363	0.486
	SC-GS [8]	20.84	21.02	21.04	0.798	0.774	0.340	0.366	0.486
050_0130	4DGS [28]	26.03	19.57	32.30	0.904	0.791	0.193	0.393	0.393
	D-3DGS [29]	24.48	18.03	30.81	0.896	0.765	0.200	0.430	0.393
	SC-GS [8]	23.98	18.23	28.90	0.852	0.740	0.211	0.420	0.393
050_0160	4DGS [28]	27.62	25.05	28.24	0.884	0.837	0.164	0.225	0.388
	D-3DGS [29]	24.28	22.97	24.52	0.879	0.808	0.212	0.276	0.388
	SC-GS [8]	23.57	24.94	23.57	0.845	0.814	0.219	0.231	0.388
060_0100	4DGS [28]	29.52	26.34	30.60	0.924	0.891	0.162	0.261	0.427
	D-3DGS [29]	29.01	27.07	29.66	0.916	0.884	0.137	0.186	0.427
	SC-GS [8]	26.66	26.31	26.95	0.897	0.865	0.188	0.239	0.427
060_0130	4DGS [28]	20.59	16.47	23.55	0.812	0.774	0.360	0.407	0.449
	D-3DGS [29]	20.72	16.40	24.35	0.830	0.797	0.256	0.319	0.449
	SC-GS [8]	20.19	16.35	22.83	0.792	0.760	0.325	0.370	0.449
070_0123	4DGS [28]	20.41	18.82	24.78	0.789	0.770	0.351	0.377	0.472
	D-3DGS [29]	18.83	17.35	22.66	0.751	0.733	0.333	0.361	0.472
	SC-GS [8]	19.48	18.09	23.16	0.753	0.732	0.332	0.353	0.472
Mono - Walk Slow									
010_0050	4DGS [28]	25.50	28.16	25.00	0.912	0.906	0.179	0.235	0.405
	D-3DGS [29]	22.92	29.43	21.97	0.882	0.891	0.146	0.163	0.405
	SC-GS [8]	21.80	30.54	20.69	0.890	0.896	0.163	0.163	0.405
020_0020	4DGS [28]	25.61	27.51	25.60	0.650	0.600	0.406	0.592	0.478
	D-3DGS [29]	25.78	24.92	25.80	0.766	0.519	0.150	0.375	0.478
	SC-GS [8]	24.83	26.72	24.82	0.663	0.559	0.179	0.351	0.478
040_0040	4DGS [28]	18.96	17.63	20.35	0.754	0.703	0.341	0.407	0.480
	D-3DGS [29]	19.71	18.98	20.40	0.785	0.747	0.350	0.393	0.480
	SC-GS [8]	18.98	19.50	18.97	0.788	0.759	0.348	0.375	0.480
050_0130	4DGS [28]	25.51	19.02	32.12	0.905	0.787	0.181	0.394	0.407
	D-3DGS [29]	23.92	17.43	30.22	0.896	0.763	0.198	0.439	0.407
	SC-GS [8]	23.66	17.26	29.68	0.885	0.744	0.204	0.445	0.407
050_0160	4DGS [28]	25.52	23.86	25.82	0.860	0.821	0.192	0.247	0.434
	D-3DGS [29]	23.40	21.41	23.87	0.820	0.777	0.211	0.294	0.434
	SC-GS [8]	22.20	23.34	22.17	0.831	0.803	0.238	0.273	0.434
060_0100	4DGS [28]	28.49	25.38	29.74	0.912	0.877	0.181	0.274	0.428
	D-3DGS [29]	26.36	25.44	26.90	0.885	0.841	0.188	0.248	0.428
	SC-GS [8]	24.79	22.91	25.47	0.882	0.832	0.202	0.269	0.428
060_0130	4DGS [28]	19.25	15.28	22.00	0.790	0.752	0.382	0.428	0.447
	D-3DGS [29]	20.90	16.74	24.22	0.826	0.802	0.291	0.332	0.447
	SC-GS [8]	20.77	17.20	23.13	0.808	0.785	0.309	0.338	0.447
070_0123	4DGS [28]	18.18	16.58	24.49	0.757	0.725	0.355	0.387	0.478
	D-3DGS [29]	19.85	18.52	24.69	0.775	0.756	0.279	0.298	0.478
	SC-GS [8]	17.85	16.41	23.16	0.742	0.714	0.308	0.332	0.478