Supplementary Material

We detail the multi-scale point generation and coordinate system of the global triplane. Furthermore, in the provided https://pointnerfpp.github.io, we provide more rendering results.

A Multi-scale Point Generation via Grid-subsampling

We build multi-scale points from an input point cloud using grid subsampling, which is more robust to varying density as shown in KPConv [49]. Specifically, a new support point at each scale is the barycenter of the original input points contained in a grid cell. Thereby, we control the scale and density at each level via the grid size of cells.

We set the grid size at the level s as $\omega * \gamma^{s-1}$ where ω is the initial grid size at the first level and γ is the stride size. We use the larger grid size for severely incomplete point clouds and a small grid size for the complete point cloud. Specifically, for `KITTI-360` [28], we set ω as 8cm and γ as 2.92. As a result, the grid size at the coarsest point level (i.e., s=4) is 2 meters. For `ScanNet` [10], we set ω as 0.008 and γ as 2.0. For `Nerf Synthetic` [30] where point clouds are relatively complete, we set ω as 0.004 and γ as 1.6.

B The Coordinate System of Global Triplane

We align the world coordinate system and the normalized coordinate system of global triplane. We use principal component analysis (PCA) to calculate the reference coordinate frame of input point cloud. The resulting reference frame consists of rotation, translation and scale, thereby defining the alignment matrix transforming world coordinates to coordinate system of global triplane. In `ScanNet` [10] and `Nerf Synthetic` [30], where points distribute uniformly along three axes and center at the origin, we simply normalize world coordinates using the scale part of the reference frame. For `KITTI 360` [28], we use full reference frame instead – i.e., we rotate, translate and scale the world coordinates – because, in this dataset, the car moves along one major direction, leading to the points heavily unbalanced along three axes. With this PCA-based canonicalization, we compactly compress all possible query points into the triplane's normalized frame, allowing for fully utilizing the capacity of the global triplane.

C More rendering results

We furthermore provide more rendering results – rendering more frames and generating videos from them. For more details, please refer to the website linked above.

Table 8: SSIM↑ on NeRF Synthetic [30]

SSIM	Uses points	**Avg.**	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
Gauss. Splat. [24]	✗	0.972	0.990	0.955	0.991	0.986	0.986	0.967	0.992	0.904
Gauss. Splat. [24]	✓	0.972	0.990	0.955	0.991	0.987	0.988	0.968	0.994	0.905
PointNeRF [55]	✓	0.967	0.987	0.987	0.987	0.976	0.977	0.940	0.991	0.895
Ours	✓	0.969	0.990	0.943	0.989	0.985	0.988	0.967	0.993	0.899

Table 9: LPIPS↓ on NeRF Synthetic [30]

SSIM	Uses points	**Avg.**	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
Gauss. Splat. [24]	✗	0.023	0.007	0.038	0.008	0.012	0.011	0.017	0.005	0.087
Gauss. Splat. [24]	✓	0.019	0.007	0.037	0.008	0.009	0.008	0.016	0.004	0.065
PointNeRF [55]	✓	0.040	0.013	0.073	0.012	0.032	0.016	0.076	0.011	0.087
Ours	✓	0.024	0.008	0.049	0.012	0.015	0.007	0.028	0.007	0.064

D SSIM and LPIPS in NeRF Synthetic

In addition to the PSNR reported in Tab. 3 on NeRF Synthetic dataset, we further provide SSIM in Tab. 8 and LPIPS in Tab. 9. Note that this dataset is highly saturated. Our method, on par with Gaussian Splatting (rerun), provides state-of-the-art results.

E Public Result in KITTI 360

We test our method in the novel view synthesis benchmark with a 50% dropping rate from KITTI360 [28]. The result is now public and de-anonymized – please refer to this link.

F The Impact of Sparse Point Cloud

In addition to Tab. 5, we further show the impact of sparse point clouds on baselines including PointNeRF [55] and Gaussian Splatting [24]. Our method outperforms two baselines, except with 1% sampling ratio, where we are on par with Gaussian Splatting. Note that Gaussian Splatting, while starting from these points, ends up using many more points via their densification scheme.

Table 10: The impact of sparse point clouds – PSNR Our method consistently outperforms baselines with sparse point clouds of different sampling ratios.

Sampling ratio (%)	Gauss. Splat.	PointNeRF	Ours
1	**18.87**	14.22	<u>18.71</u>
10	<u>18.96</u>	16.04	**19.35**
100	<u>18.59</u>	17.63	**20.05**