A. Appendix

A.1. Details of Network Architecture

Table 6: The CNN Module for experiments on the ShapeNetv2 dataset in Sections 4.1 and 4.2

Type	Size/Channels	Activation	Stride
Input: embedding of RGB and viewpoint	-	-	-
L1: Conv 7×7	64	ReLU	2
L2: Conv 3×3	128	ReLU	2
L3: Conv 3×3	256	ReLU	2
L4: Conv 3×3	512	ReLU	2
L5: Conv 4×4	128	ReLU	4
L6: Flatten / Tile	-	-	-
L7: Concat (L6, L4)	-	-	-
L8: Dilated Conv 3×3	256	ReLU	4
L8: Concat (L8, L3)	-	-	-
L9: Dilated Conv 3×3	128	ReLU	8
L9: Concat (L9, L2)	-	-	-
L10: Dilated Conv 3×3	64	ReLU	16
L10: Concat (L10, L1)	-	-	-
L11: Dilated Conv 3×3	128	ReLU	32

Table 7: The CNN Module for experiments on the Synthetic-NeRF dataset in Section 4.3. The average pooling is added to aggressively downsample the feature maps.

Type	Size/Channels	Activation	Stride
Input: embedding of RGB and viewpoint	-	-	-
L1: Conv 7×7	64	ReLU	2
L2: Conv 3×3	128	ReLU	2
L3: Conv 3×3	256	ReLU	2
L4: Conv 3×3	512	ReLU	2
L4: AveragePooling 5×5	-	-	-
L5: Conv 5×5	128	ReLU	5
L6: Flatten / Tile	-	-	-
L7: Concat (L6, L4)	-	-	-
L8: Deconv 3×3	256	ReLU	2
L8: Concat (L8, L3)	-	-	-
L9: Deonv 3×3	128	ReLU	2
L9: Concat (L9, L2)	-	-	-
L10: Deconv 3×3	64	ReLU	2
L10: Concat (L10, L1)	-	-	-
L11: Deconv 3×3	128	ReLU	2

Table 8: The CNN Module for experiments on the real-world dataset (LLFF) in Section 4.4. The average pooling is added to aggressively downsample the feature maps.

Type	Size/Channels	Activation	Stride
Input: embedding of RGB and viewpoint	-	-	-
L1: Conv 7×7	64	ReLU	2
L2: Conv 3×3	128	ReLU	2
L3: Conv 3×3	256	ReLU	2
L4: Conv 3×3	512	ReLU	2
L4: AveragePooling 8×8	-	-	-
L5: Conv 4×4	128	ReLU	4
L6: Flatten / Tile	-	-	-
L7: Concat (L6, L4)	-	-	-
L8: Deconv 3×3	256	ReLU	2
L8: Concat (L8, L3)	-	-	-
L9: Deconv 3×3	128	ReLU	2
L9: Concat (L9, L2)	-	-	-
L10: Deconv 3×3	64	ReLU	2
L10: Concat (L10, L1)	-	-	-
L11: Deconv 3×3	128	ReLU	2

Table 9: The Attention Module, AttSets [56], for experiments on the ShapeNetv2 Dataset in Sections 4.1 and 4.2. The simple AttSets is computationally efficient and we choose it to train the large-scale ShapeNetv2 dataset.

Type	Size/Channels	Activation
Input: Concat($K \times 128$, embedding of viewpoint)	-	-
L1: fc	256	ReLU
L2: fc	256	ReLU
L3: fc	256	ReLU
L4: fc	512	ReLU
L5: fc	512	ReLU
L6: softmax(L5)	-	-
L7: sum(L6*L5, axis=-2)	-	-
L8: fc	512	ReLU

We use Slot Attention as the pixel feature aggregation module for experiments on the Synthetic-NeRF and the real-world dataset in Sections 4.3 and 4.4. In particular, we use two slots, two iterations, and the hidden size is 128. The final output two slots are flattened and a 256 dimensional vector is obtained.

For details of Slot Attention refer to the paper [24]. Details of the neural rendering layers and the volume rendering can be found in NeRF [29]. We set the positional embedding length $L = 5$ for all inputs to the CNN module, except the rotation, which we convert to quaternion and embed at $L = 4$.

During training, we feed the models between 2 and 6 views of each geometry at each gradient step. We set the learning rate for the ShapeNetv2 models at 1e-4. We set the learning rate for leaves and orchids in the real-world dataset at 7e-5, and for the rest, we use 1e-4. For Synthetic-NeRF dataset, we use a learning rate of 1e-4. We use the Adam optimizer for all models, and train for 200k-300k iterations. At each gradient step, we take 1000 rays for ShapeNetv2 with 32 coarse samples and 64 fine samples, and 800 rays for the real-world and Synthetic-NeRF datasets with 64 coarse samples and 192 fine samples. We train each model on a single Nvidia-V100 GPU with 32GB VRAM.

During testing on the ShapeNetv2 dataset in Section 4.1, we feed the model the 4 closest views by cosine similarity to the desired novel view.

A.2. Details of Experimental Results on the Synthetic-NeRF Dataset in Section 4.3

Table 10: The PSNR, SSIM and LPIPS scores of our GRF simultaneously trained on 4 scenes of the Synthetic-NeRF dataset for multi-scene learning in Section 4.3. The scores of SRNs, NeRF and NSVF trained on single scenes are included for comparison.

	Chair	Mic	Ship	Hotdog
		PSNR↑		
SRNs (Single-scene)	26.96	26.85	20.60	26.81
NeRF (Single-scene)	33.00	32.91	**28.65**	36.18
NSVF (Single-scene)	**33.19**	**34.27**	27.93	**37.14**
GRF (Multi-scene)	32.49	32.02	27.76	34.92
		SSIM↑		
SRNs (Single-scene)	0.910	0.947	0.757	0.923
NeRF (Single-scene)	0.967	0.980	0.856	0.974
NSVF (Single-scene)	0.968	**0.987**	0.854	**0.980**
GRF (Multi-scene)	**0.971**	0.982	**0.866**	0.975
		LPIPS↓		
SRNs (Single-scene)	0.106	0.063	0.299	0.100
NeRF (Single-scene)	0.046	0.028	0.206	0.121
NSVF (Single-scene)	0.043	**0.010**	**0.162**	**0.025**
GRF (Multi-scene)	**0.032**	0.019	0.167	0.040

Table 11: The PSNR, SSIM and LPIPS scores of our GRF and NeRF on four novel scenes of Synthetic-NeRF in Group 1&2 experiments in Section 4.3.

	Drums	Lego	Materials	Ficus	*mean*
		PSNR↑			
GRF (Group 1)	13.23	13.53	12.26	15.47	13.62
NeRF (Group 2, 100 iters)	14.54	14.92	15.42	15.72	15.15
NeRF (Group 2, 1k iters)	18.01	20.04	20.40	20.81	19.81
NeRF (Group 2, 10k iters)	21.57	24.99	23.36	23.47	23.35
GRF (Group 2, 100 iters)	18.70	20.24	18.81	21.03	19.69
GRF (Group 2, 1k iters)	20.49	23.64	21.87	22.02	22.00
GRF (Group 2, 10k iters)	23.11	27.07	25.11	25.11	25.10
		SSIM↑			
GRF (Group 1)	0.762	0.736	0.703	0.849	0.763
NeRF (Group 2, 100 iters)	0.769	0.717	0.716	0.808	0.752
NeRF (Group 2, 1k iters)	0.793	0.775	0.812	0.857	0.809
NeRF (Group 2, 10k iters)	0.865	0.862	0.877	0.896	0.875
GRF (Group 2, 100 iters)	0.822	0.813	0.829	0.878	0.835
GRF (Group 2, 1k iters)	0.856	0.877	0.878	0.894	0.876
GRF (Group 2, 10k iters)	0.901	0.924	0.913	0.923	0.916
		LPIPS↓			
GRF (Group 1)	0.256	0.273	0.301	0.150	0.246
NeRF (Group 2, 100 iters)	0.332	0.395	0.314	0.393	0.359
NeRF (Group 2, 1k iters)	0.254	0.264	0.229	0.164	0.228
NeRF (Group 2, 10k iters)	0.157	0.154	0.128	0.110	0.137
GRF (Group 2, 100 iters)	0.196	0.203	0.159	0.117	0.169
GRF (Group 2, 1k iters)	0.154	0.138	0.123	0.097	0.128
GRF (Group 2, 10k iters)	0.104	0.090	0.090	0.071	0.089

A.3. Details of Experimental Results on the real-world dataset (3DScan) [6] in Section 4.3

We select four 360-degree-scanned chair scenes from the challenging real-world 3DScan dataset [6]. A single model is trained for 100000 iterations on 100 images each of three scenes with the following indices: 00032, 00027, 00279. Then, the model is finetuned with a small number of iterations on the scene 00169 from a sparse set of 50 views. The results below show the generality of the features learned by GRF, and that the model quickly converges to plausible representations of complicated real-world object-based scenes. We can see that it is extremely challenging to obtain high-quality results for complex real-world scenes. We leave it for future work to further improve the generalization capability of GRF.

	PSNR↑	SSIM↑
GRF (1k iters)	18.80	0.640
GRF (10k iters)	20.19	0.662

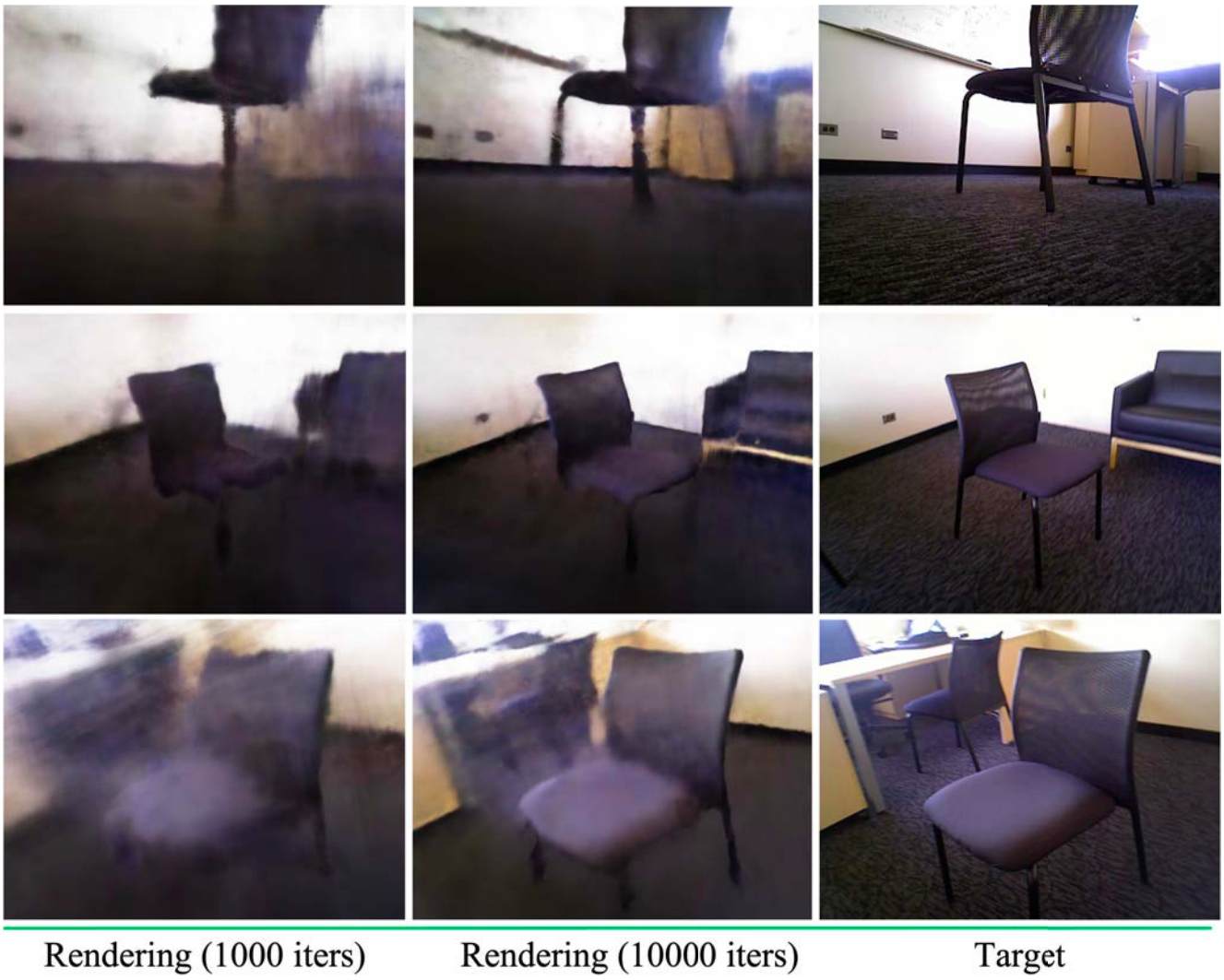

Figure 11: Quantitative and Qualitative results of our GRF for novel view synthesis on a real-world chair after finetuning.

A.4. Details of experimental results on the real-world dataset (LLFF) in Section 4.4.

Table 12: Comparison of the PSNR, SSIM and LPIPS scores of our GRF, SRNs [45], LLFF [28] and NeRF [29] in the real-world dataset for single-scene learning in Section 4.4.

	Room	Fern	Leaves	Fortress	Orchids	Flower	T-Rex	Horns	*Mean*
					PSNR↑				
SRNs	27.29	21.37	18.24	26.63	17.37	24.63	22.87	24.33	22.84
LLFF	28.42	22.85	19.52	29.40	18.52	25.46	24.15	24.70	24.13
NeRF	**32.70**	25.17	20.92	31.16	20.36	27.40	26.80	27.45	26.50
GRF(Ours)	31.74	**25.72**	**21.16**	**31.28**	**20.88**	**27.83**	**27.01**	**27.50**	**26.64**
					SSIM↑				
SRNs	0.883	0.611	0.520	0.641	0.449	0.738	0.761	0.742	0.668
LLFF	0.932	0.753	0.697	0.872	0.588	0.844	0.857	0.840	0.798
NeRF	0.948	0.792	0.690	0.881	0.641	0.827	0.880	0.828	0.811
GRF(Ours)	**0.951**	**0.827**	**0.727**	**0.898**	**0.667**	**0.852**	**0.901**	**0.873**	**0.837**
					LPIPS↓				
SRNs	0.240	0.459	0.440	0.453	0.467	0.288	0.298	0.376	0.378
LLFF	0.155	0.247	**0.216**	0.173	0.313	**0.174**	0.222	0.193	0.212
NeRF	0.178	0.280	0.316	0.171	0.321	0.219	0.249	0.268	0.250
GRF(Ours)	**0.104**	**0.191**	0.238	**0.127**	**0.275**	0.176	**0.146**	**0.169**	**0.178**

Table 13: Comparison of the PSNR (in dB), SSIM and LPIPS [58] scores of our GRF, SRNs [45], NV [25], NeRF [29] and NSVF [21] in the Synthetic-NeRF dataset for single-scene learning.

	Chair	Drums	Lego	Mic	Materials	Ship	Hotdog	Ficus	*Mean*
					PSNR↑				
SRNs	26.96	17.18	20.85	26.85	18.09	20.60	26.81	20.73	22.26
NV	28.33	22.58	26.08	27.78	24.22	23.93	30.71	24.79	26.05
NeRF	33.00	25.01	32.54	32.91	29.62	28.65	36.18	30.13	31.01
NSVF	33.19	25.18	32.29	**34.27**	**32.68**	27.93	37.14	**31.23**	31.74
GRF(Ours)	**34.51**	**25.83**	**32.92**	33.94	30.91	**30.12**	**37.47**	30.75	**32.06**
					SSIM↑				
SRNs	0.910	0.766	0.809	0.947	0.808	0.757	0.923	0.849	0.846
NV	0.916	0.873	0.880	0.946	0.888	0.784	0.944	0.910	0.893
NeRF	0.967	0.925	0.961	0.980	0.949	0.856	0.974	0.964	0.947
NSVF	0.968	0.931	0.960	**0.987**	**0.973**	0.854	0.980	**0.973**	0.953
GRF(Ours)	**0.981**	**0.937**	**0.967**	**0.987**	0.963	**0.891**	**0.983**	0.969	**0.960**
					LPIPS↓				
SRNs	0.106	0.267	0.200	0.063	0.174	0.299	0.100	0.149	0.170
NV	0.109	0.214	0.175	0.107	0.130	0.276	0.109	0.162	0.160
NeRF	0.046	0.091	0.050	0.028	0.063	0.206	0.121	0.044	0.081
NSVF	0.043	0.069	**0.029**	**0.010**	**0.021**	0.162	**0.025**	**0.017**	**0.047**
GRF(Ours)	**0.021**	**0.068**	0.042	0.013	0.041	**0.141**	0.028	0.032	0.048

In order to push the boundaries of single-scene learning, we also conduct experiments on the Synthetic-NeRF dataset in addition to the experiments on real-world scenes in Section 4.4. The detailed results are shown in Table 13. Our GRF outperforms the state-of-the-art NSVF approach on both PSNR and SSIM.

A.5. Analysis of Attention Mechanism

The attention mechanism in our GRF aims to automatically select the correct pixel patch from multiple pixel patches where the light rays intersect at the same query 3D point in space.

In order to investigate how the attention mechanism learns to select the useful information, we retrieve the maximal attention score from the observed multiple pixel patches for analysis. Intuitively, the higher the attention score is assigned to a particular pixel patch, the more important that patch for inferring the novel pixel RGB. In particular, we conduct the following experiment using our GRF model trained on ShapeNetv2 Cars. In this case, the AttSets attention module is used (details are in Table 9). Given a query light ray, multiple 3D points are sampled to query the network.

- We firstly try to find the 3D point which is near the surface according to the predicted volume density for points along the ray through a given pixel, if they exist. Otherwise, we ignore such pixels, making them white.
- Then we compute the M feature vectors from the input M views for these surface points.
- Thirdly, the attention masks for those M feature vectors are computed. We identify the view whose sum of the attention mask along the feature axis is greatest as the main contributor for inferring the novel pixel RGB.
- After querying light rays for each pixel, we obtain a rendered RGB image. At the same time, for each pixel of that image, we select the most important view from the M input views for the surface-intersection point along the ray from the viewpoint through that pixel, according to the maximal attention score. Eventually, we obtain a Max Attention Map corresponding to the rendered RGB image.

Figure 12 shows the qualitative results of the above experiment. In particular, we feed the three images (#1,#2,#3) of an unseen car into our GRF model which is well-trained on car category, and then render a new image (e.g., the 5th image in Figure 12). Note that, we carefully select the input 3 images and the rendered image with very large viewing baselines. In the mean time, we obtain and visualize the Max Attention Map corresponding to the rendered image.

For each pixel of the rendered image, we retrieve the input image pixel that has the highest attention score. Specifically, the rendered pixels with purple color correspond to the input image #1, the green pixels correspond to the input image #2, while the blue pixels correspond to the input image #3.

Analysis. It can be seen that, when inferring a new image, the attention module of our GRF focuses on the most informative pixel patch from the multiple input pixel patches. In addition, it is able to truly deal with the visual occlusion. For example, when inferring the windshield of the car, the attention module focuses on the input image #2 where the windshield is visible, while ignoring the image #1 and #3 where the windshield is self-occluded.

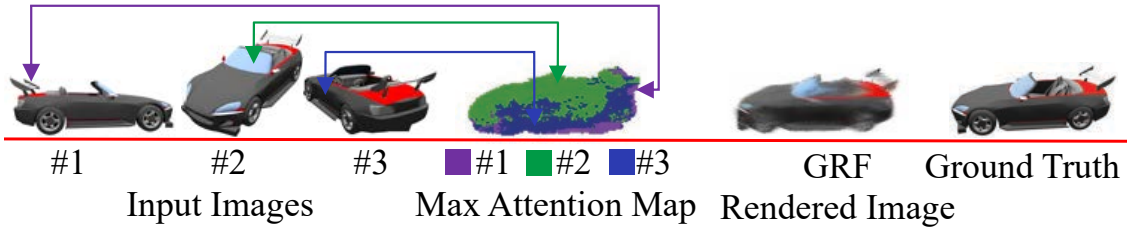

Figure 12: Visualization of Max Attention Map from Multiple Input Images of a Novel Object for Inferring a Novel View.

A.6. Generalization to Visual Occlusions and Variable Input Images

We carefully select the attention module, i.e., either AttSets or SlotAtt, to aggregate the features from an arbitrary number of input views. In order to evaluate how our GRF is able to generalize with a variable number of input views, especially when there is a very sparse number of views with severe visual occlusions, we conduct the following four groups of experiments.

- **1-view Reconstruction**. We feed the a single image of a novel car into our GRF model which is well-trained on car category (trained with 5 images per object), and then render 9 new images from vastly different viewing angles. This is the extreme case where the majority of the object is self-occluded.
- **2-view Reconstruction**. Similarly, we feed only two images of the novel car into the same model and render the same 9 novel views. In this case, more information is given to the network, but there are still many parts occluded.
- **5-view / 10-view Reconstruction**. The same GRF model is fed with 5 and 10 views of the novel object, rendering the same set of new images.

Analysis. Figure 13 shows the qualitative results. It can be seen that: 1) In the extreme case, i.e., 1-view reconstruction, our GRF is still able to recover the general 3D shape of the unseen object, including the visually occluded parts, primarily because our CNN model learns the hierarchical features including the high-level shapes. 2) Given more input views, the originally occluded parts tend to be observed from some viewing angles, and then these parts can be reconstructed better and better. This shows that our GRF is indeed able to effectively identify the corresponding useful pixel features for more accurately recovering shape and appearance.

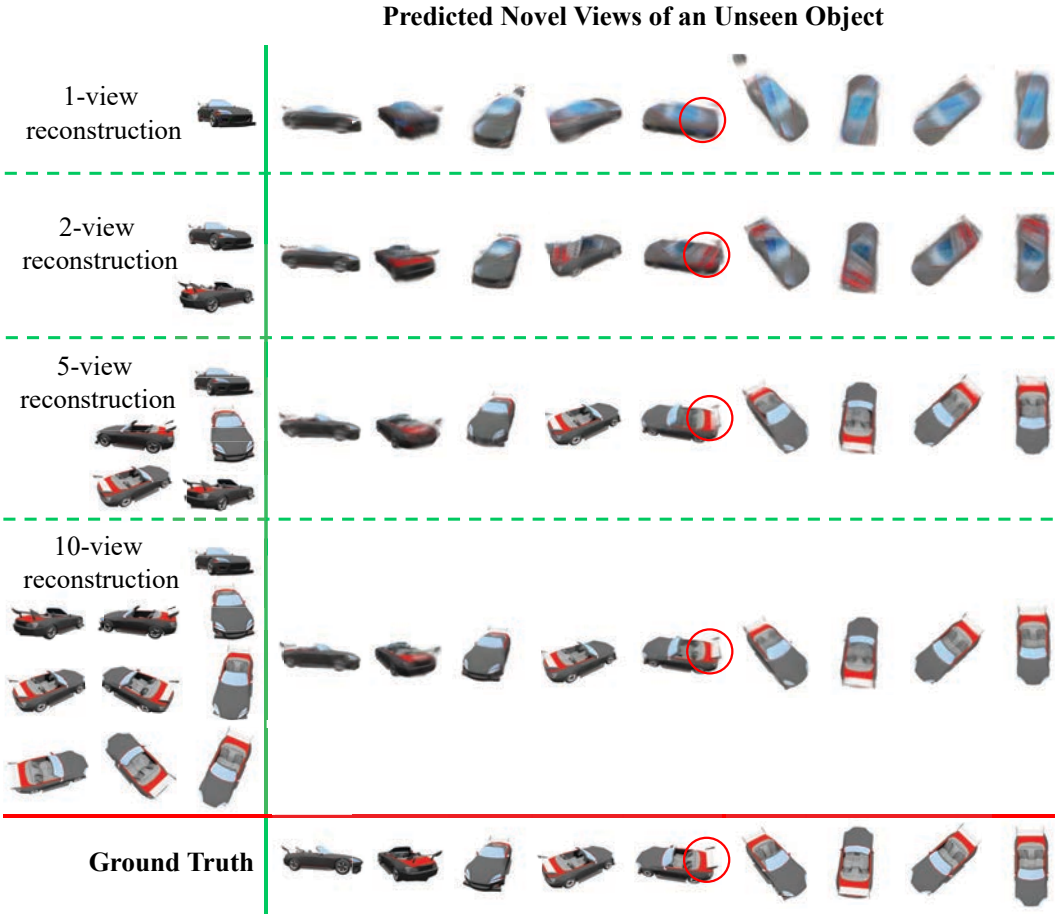

Figure 13: Qualitative results of our GRF when being fed with a variable number of views of a novel object. The red circle highlights that the tail of the car is able to be recovered given more visual cues from more input images.

A.7. More Qualitative Results of real-world scenes in Section 4.4

Figure 14: Qualitative results of our GRF for novel view depth and RGB estimation on the real-world dataset in Section 4.4.

Figure 15: Qualitative results of our GRF for novel view depth and RGB estimation on the real-world dataset in Section 4.4.