PlückeRF: A Line-based 3D Representation for Few-view Reconstruction

Supplementary Material

In this appendix we present additional qualitative results, training details for compared methods, and details about the training and inference of our trained models.

A. Additional Results

We present additional qualitative static comparisons on ShapeNet-SRN 'Chairs' and 'Cars' including more views than shown in the main results. We compare our method to OpenLRM [7] which has been trained on a single object category, pixelNeRF [41], Splatter Image [31], our baseline without bias, and our main method in Fig. 6 and Fig. 7.

B. Shapenet SRN Dataset Details

The Shapenet SRN 'Cars' dataset contains 3514 instances, with a predefined train/val/test split of 2458/352/704 instances respectively. The 'Chairs' dataset contains 6519 instances with a 4612/662/1317 train/val/test split. Each training instance contains 50 posed images taken from random points on a sphere around the object. Each testing and validation instance contains 251 posed images taken on an Archimedean spiral along the sphere. All scenes share camera intrinsics, and images are rendered at a resolution of (128, 128).

C. Model Training and Inference Details

We use the AdamW [18] optimizer with weight decay of 0.05 and β_1 of 0.9, β_2 of 0.95. Our models were trained on 4 GPUs (Nvidia A100-40GB) with a batch size of 32 and with separate models for each object category. Training takes 5 days on 4 A100-40GB GPUs for 500k training steps. Inference takes roughly 15 seconds to render all 251 novel test views for an image pair on a single A100.

Our image encoder is initialized as a pre-trained small DINOv2 vision transformer [23], which has a token size of 384 (`dinov2_vits14_reg`) and creates image patches of size 14×14 pixels. We rescale our input images to 448×448 pixels using bicubic interpolation to get 1024 patches per image.

We use a maximum learning rate of 4×10^{-4} for our 'Chairs' model, and 8×10^{-5} for the 'Cars' model, and a cosine scheduler for the learning rate, with 2500 warm-up iterations. We train our 'Cars' model for 500k iterations, and set the LPIPS loss weight α to 0 at the start of training and increase it to 0.01 at iteration 400k. We train our 'Chairs' model for 800k iterations, and set α to 0.01 at iteration 650k.

D. Splatter Image Training Details

We use the official Splatter Image implementation [31]. Our 2 view Splatter Image model for the Shapenet-SRN 'Chairs' category was trained with their 2 view training configuration recommendations with a L40S GPU for 800k iterations with a batch size of 8, and then fine-tuned with an LPIPS loss for 100k additional training iterations.

E. OpenLRM Training Details

We trained our single category OpenLRM models with the same training parameters as the main experiment where possible. We freeze the image encoder as per the original LRM architecture [8] and use the same image encoder model (`dinov2_vits14_reg`) as our main experiments. We used the same training loss as our Shapenet-SRN 'Cars' model, setting α to 0.01 at training step 400k, and trained each model for a total of 500k steps. From our experiments, setting a greater emphasis on the $\mathcal{L}_{\text{MSE}}$ loss resulted in higher PSNR in the final novel views.

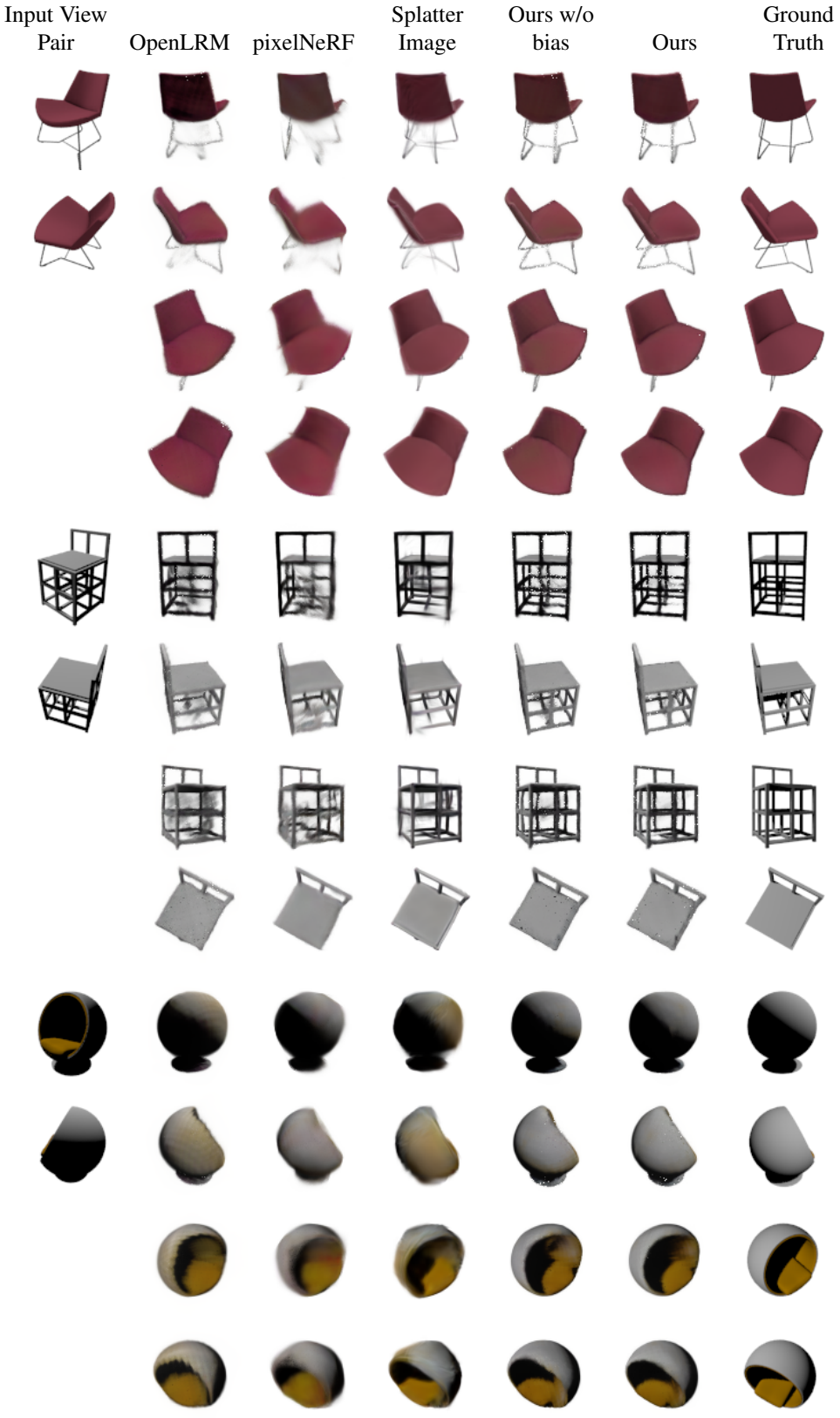

Figure 6. Additional Shapenet-SRN Chairs qualitative results on chairs in the testing split. Given two input views (one in the OpenLRM case), render novel views from around the object. Please zoom in to observe finer details.

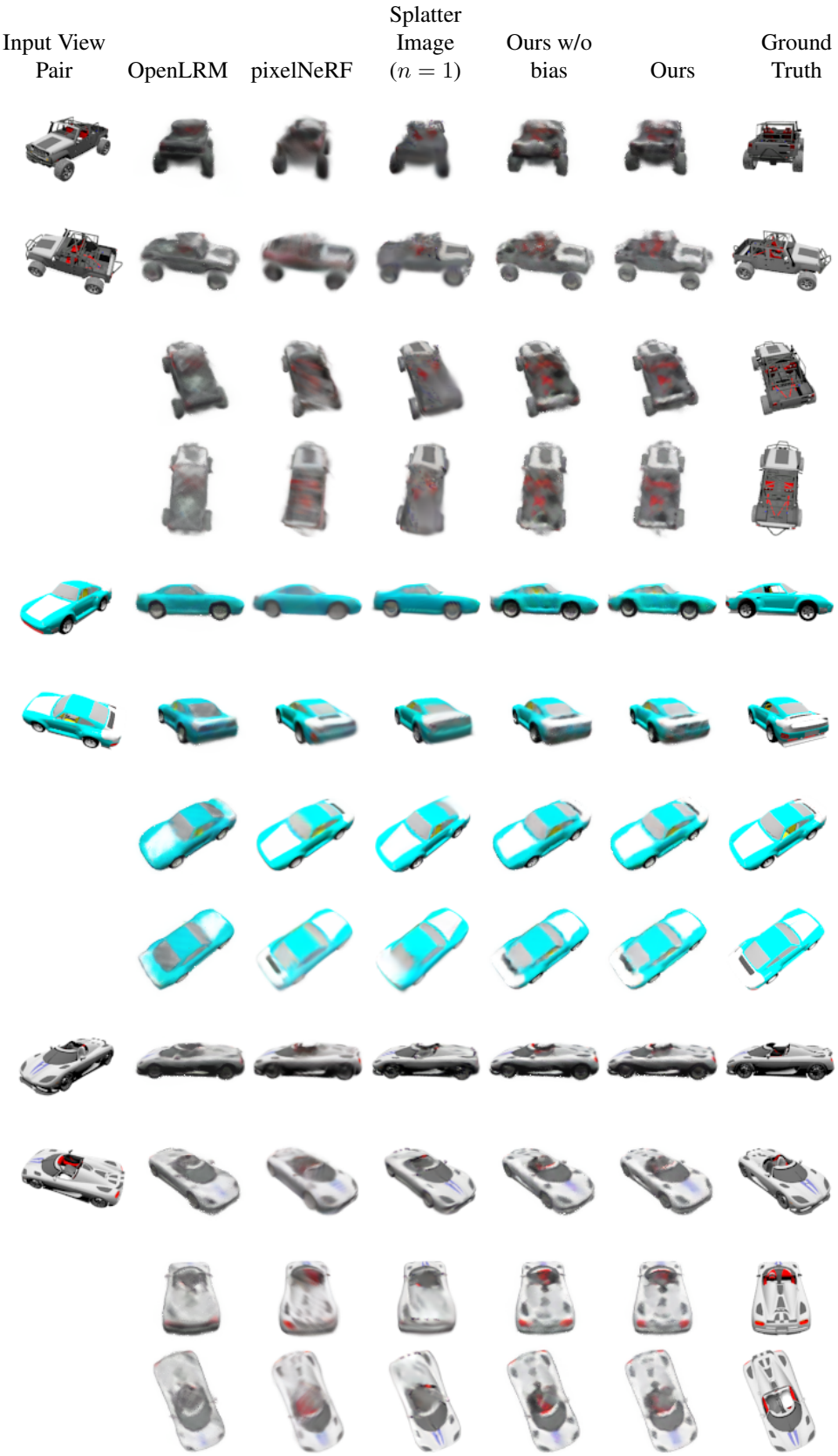

Figure 7. Additional Shapenet-SRN Cars qualitative results on cars in the testing split. Given two input views (one in the OpenLRM and Splatter Image case), render novel views from around the object. Please zoom in to observe finer details.