

NeRF from Few Images

pixelNeRF, DS-NeRF, (and RegNeRF)

Presenter: Minho Park

pixelNeRF: Neural Radiance Fields from One or Few Images

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa

UC Berkeley

CVPR 2021

Presenter: Minho Park

NeRF

- Objective

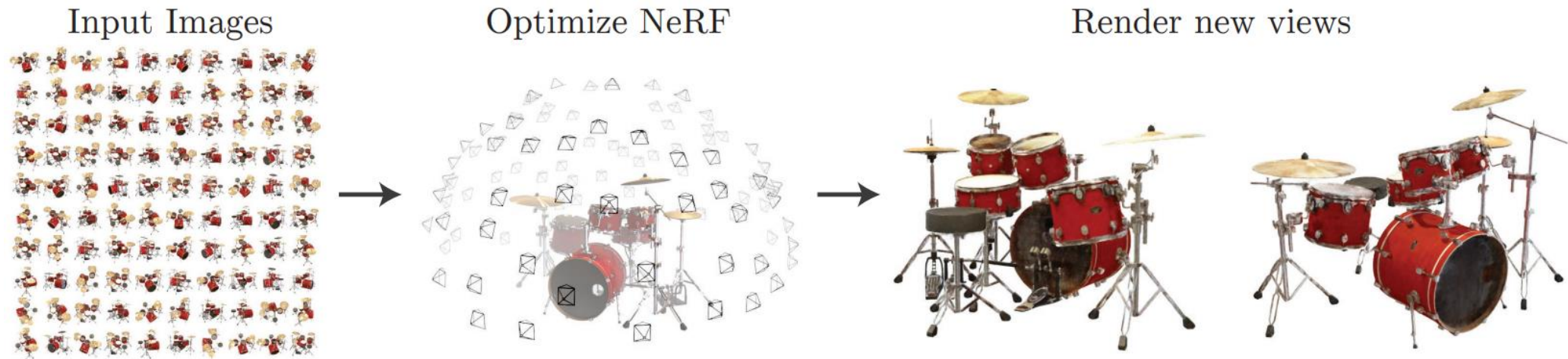


Fig. 1: We present a method that optimizes a continuous 5D neural radiance field representation (volume density and view-dependent color at any continuous location) of a scene from a set of input images.

NeRF

- Implicit Neural Representation: One network for one scene.

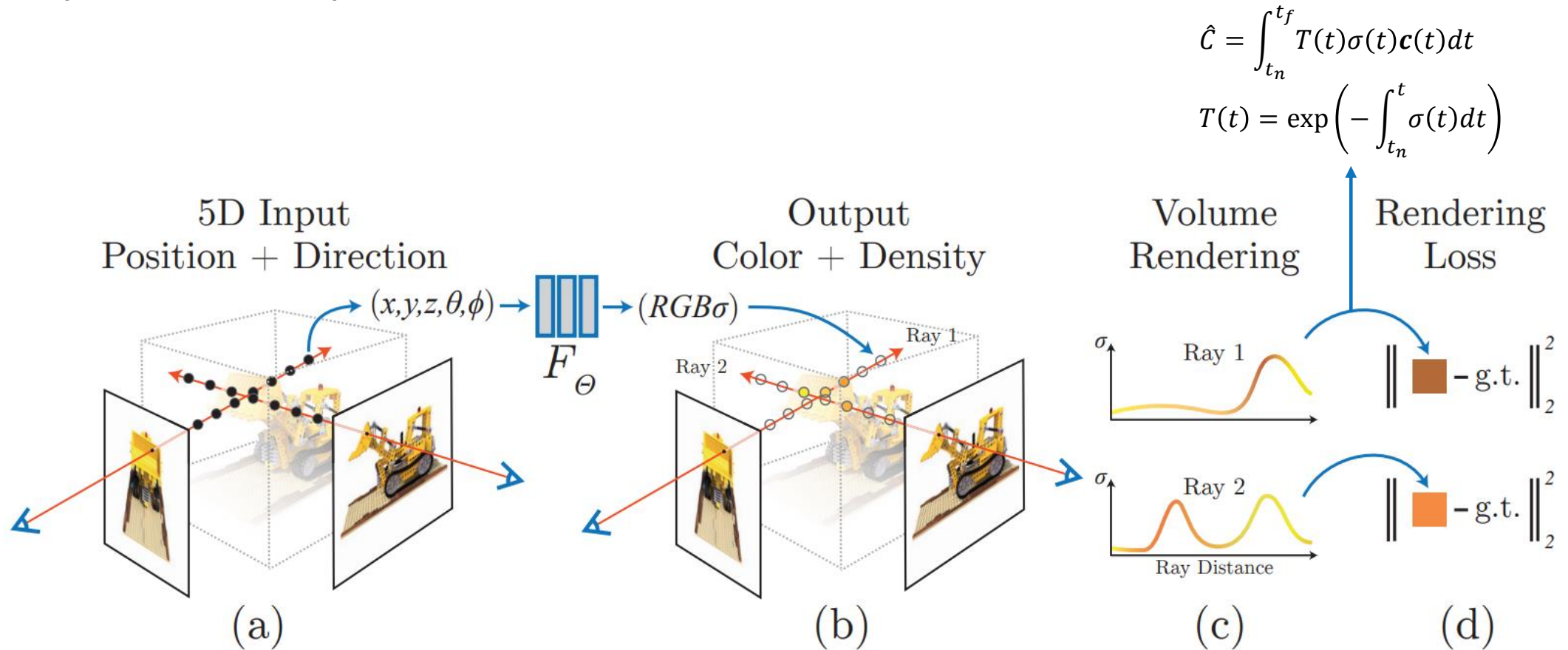


Fig. 2: An overview of our neural radiance field scene representation and differentiable rendering procedure.

NeRF

- Positional Encoding

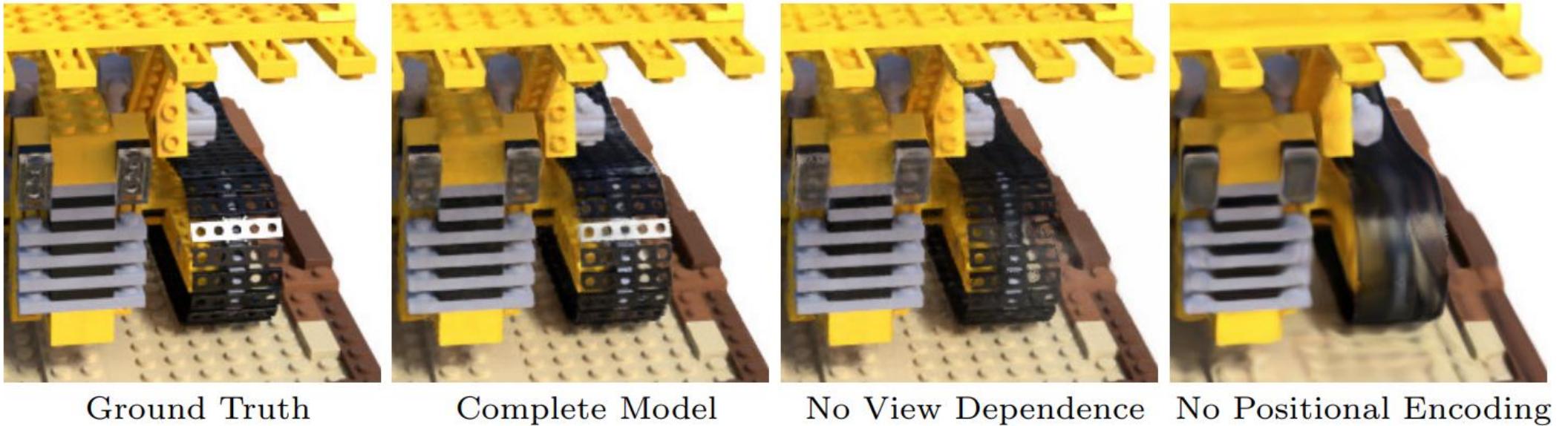


Fig. 4: Here we visualize how our full model benefits from representing view dependent emitted radiance and from passing our input coordinates through a high-frequency positional encoding.

Limitations of NeRF

- It is an optimized-based approach.
 - It is time-consuming.
- Scene must be optimized individually, with no knowledge shared between scenes.
 - It has limited performance with single or extremely sparse views.

⇒ **We want to inject a prior into the network.**

Single-Image pixelNeRF

- **Image-conditioned NeRF:** To share knowledge between scenes, they propose an architecture to **condition NeRF on spatial image features**.
- The model is comprised of two components: a fully-convolutional image encoder E , and a NeRF network f .

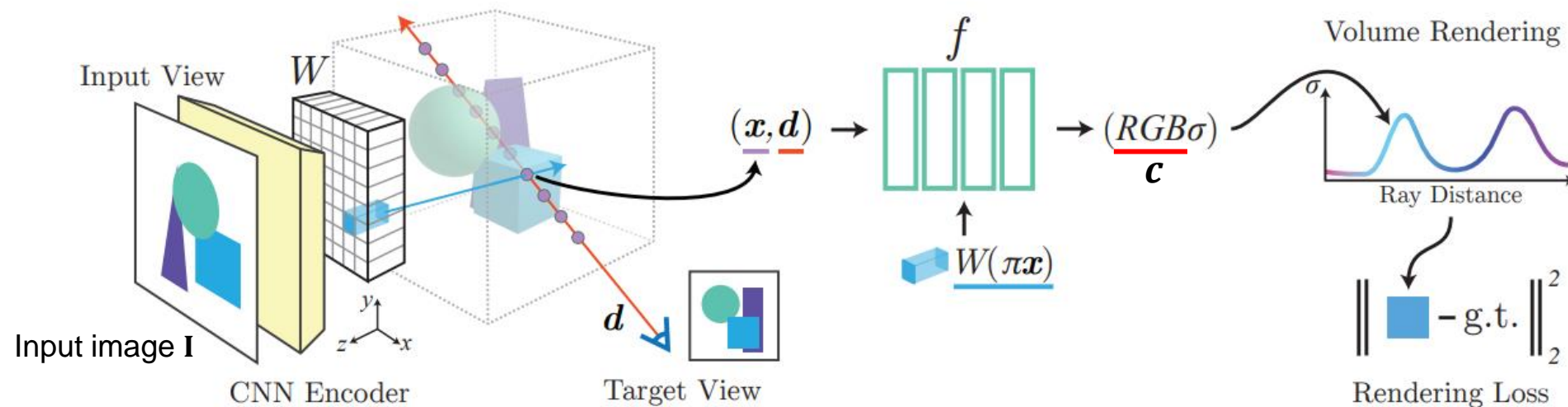
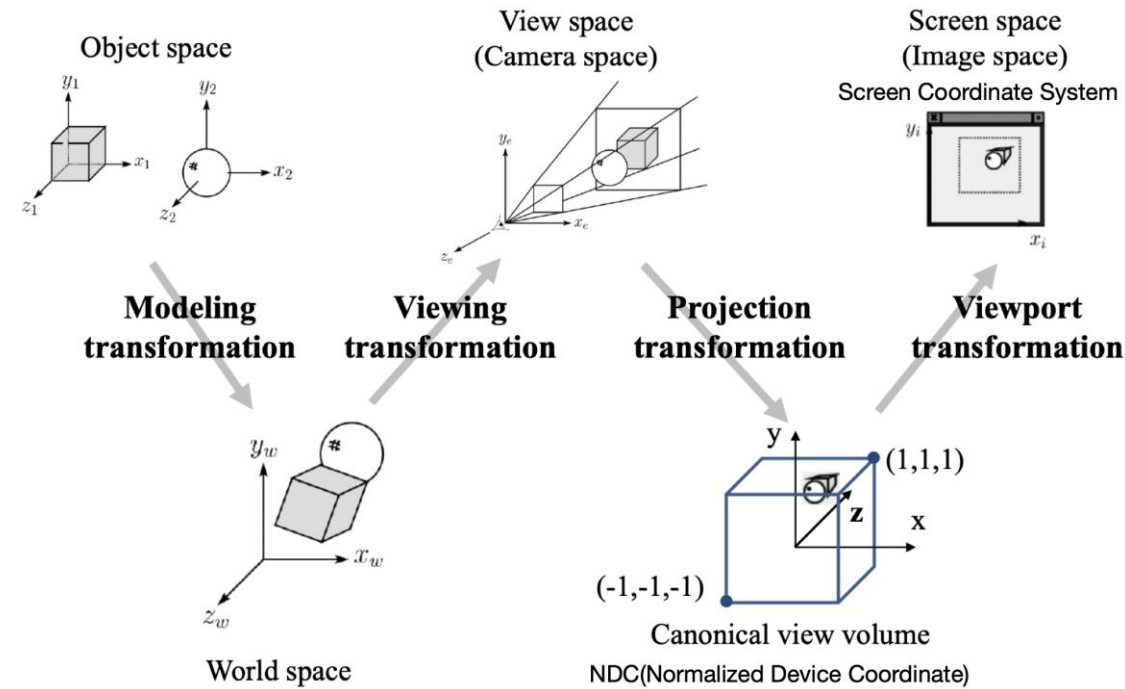


Figure 2: Proposed architecture in the single-view case.

Single-Image pixelNeRF

- Coordinate system: view space (\leftrightarrow canonical space)
 - View space: viewer-centered coordinate system
 - Canonical space: object-centered coordinate system



Vertex Processing (Transformation Pipeline)

Single-Image pixelNeRF

- Extract a feature volume $\mathbf{W} = E(\mathbf{I})$.
- $f(\gamma(\mathbf{x}), \mathbf{d}; \mathbf{W}(\pi(\mathbf{x}))) = (\sigma, \mathbf{c})$

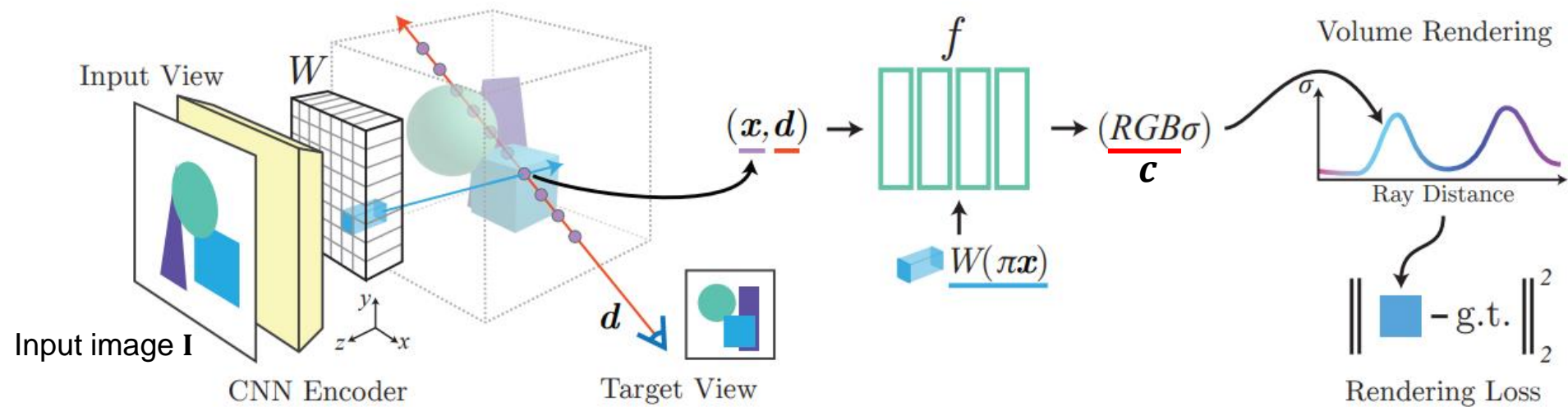


Figure 2: Proposed architecture in the single-view case.

Single-Image pixelNeRF

- If the query view direction is similar to the input view orientation, the model can rely more directly on the input.
- If it is dissimilar, the model must leverage the learned prior from CNN Encoder E .

Incorporating Multiple Views

- Formulation
 - $\mathbf{I}^{(i)}$: i -th input
 - $\mathbf{P}^{(i)} = [\mathbf{R}^{(i)} \quad \mathbf{t}^{(i)}]$: Its associated camera transform from the world space to its view space.
 - $\mathbf{x}^{(i)} = \mathbf{P}^{(i)}\mathbf{x}$, $\mathbf{d}^{(i)} = \mathbf{R}^{(i)}\mathbf{d}$: Transform query point to view space.
- $(\sigma, \mathbf{c}) = f_2 \left(\psi(\mathbf{V}^{(1)}, \dots, \mathbf{V}^{(n)}) \right)$
 - ψ : Average pooling operator
 - $\mathbf{V}^{(i)} = f_1 \left(\gamma(\mathbf{x}^{(i)}), \mathbf{d}^{(i)}; \mathbf{W}^{(i)} \left(\pi(\mathbf{x}^{(i)}) \right) \right)$
 - In the single-view special case, $f = f_1 \circ f_2$

Category-specific 1- and 2-view Reconstruction

		1-view		2-view	
		PSNR	SSIM	PSNR	SSIM
Chairs	GRF [44]	21.25	0.86	22.65	0.88
	TCO [41] *	21.27	0.88	21.33	0.88
	dGQN [9]	21.59	0.87	22.36	0.89
	ENR [8] *	22.83	-	-	-
	SRN [40]	22.89	0.89	24.48	0.92
	Ours *	23.72	0.91	26.20	0.94
Cars	SRN [40]	22.25	0.89	24.84	0.92
	ENR [8] *	22.26	-	-	-
	Ours *	23.17	0.90	25.66	0.94

Table 2: **Category-specific 1- and 2-view reconstruction.** Methods marked * do not require canonical poses at test time. In all cases, a single model is trained for each category and used for both 1- and 2-view evaluation. Note ENR is a 1-view only model.



Figure 3: **Category-specific single-view reconstruction benchmark.** We train a separate model for cars and chairs and compare to SRN. The corresponding numbers may be found in Table 2.

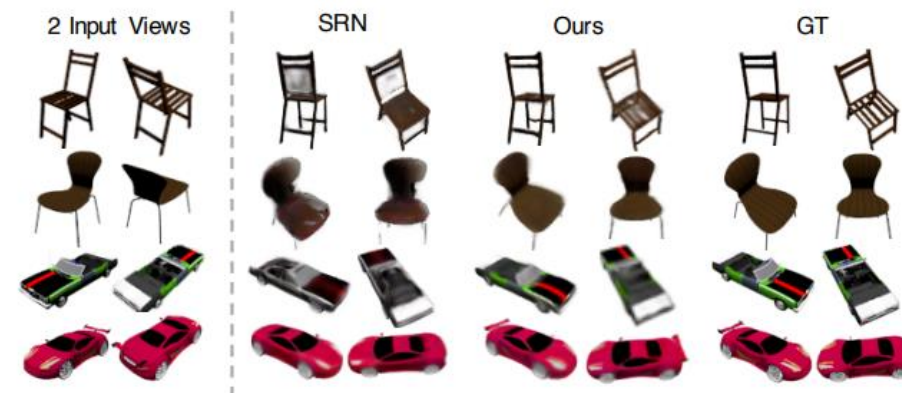
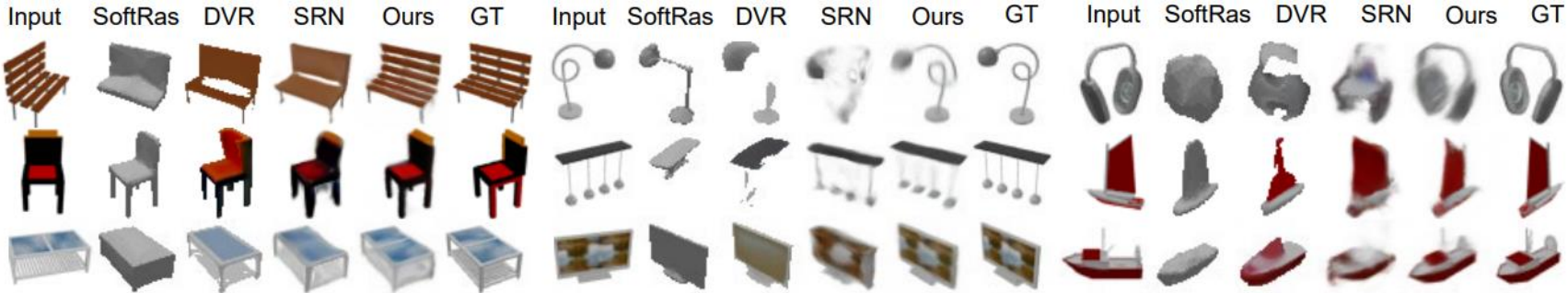


Figure 4: **Category-specific 2-view reconstruction benchmark.** We provide two views (left) to each model, and show two novel view renderings in each case (right). Please also refer to Table 2.

Category-agnostic Single-view Reconstruction.



		plane	bench	cbnt.	car	chair	disp.	lamp	spkr.	rifle	sofa	table	phone	boat	mean
↑ PSNR	DVR	25.29	22.64	24.47	23.95	19.91	20.86	23.27	20.78	23.44	23.35	21.53	24.18	25.09	22.70
	SRN	26.62	22.20	23.42	24.40	21.85	19.07	22.17	21.04	24.95	23.65	22.45	20.87	25.86	23.28
	Ours	29.76	26.35	27.72	27.58	23.84	24.22	28.58	24.44	30.60	26.94	25.59	27.13	29.18	26.80
↑ SSIM	DVR	0.905	0.866	0.877	0.909	0.787	0.814	0.849	0.798	0.916	0.868	0.840	0.892	0.902	0.860
	SRN	0.901	0.837	0.831	0.897	0.814	0.744	0.801	0.779	0.913	0.851	0.828	0.811	0.898	0.849
	Ours	0.947	0.911	0.910	0.942	0.858	0.867	0.913	0.855	0.968	0.908	0.898	0.922	0.939	0.910
↓ LPIPS	DVR	0.095	0.129	0.125	0.098	0.173	0.150	0.172	0.170	0.094	0.119	0.139	0.110	0.116	0.130
	SRN	0.111	0.150	0.147	0.115	0.152	0.197	0.210	0.178	0.111	0.129	0.135	0.165	0.134	0.139
	Ours	0.084	0.116	0.105	0.095	0.146	0.129	0.114	0.141	0.066	0.116	0.098	0.097	0.111	0.108

Table 4: **Category-agnostic single-view reconstruction.** Quantitative results for category-agnostic view-synthesis are presented, with a detailed breakdown by category. Our method outperforms the state-of-the-art by significant margins in all categories.

Pushing the Boundaries of ShapeNet

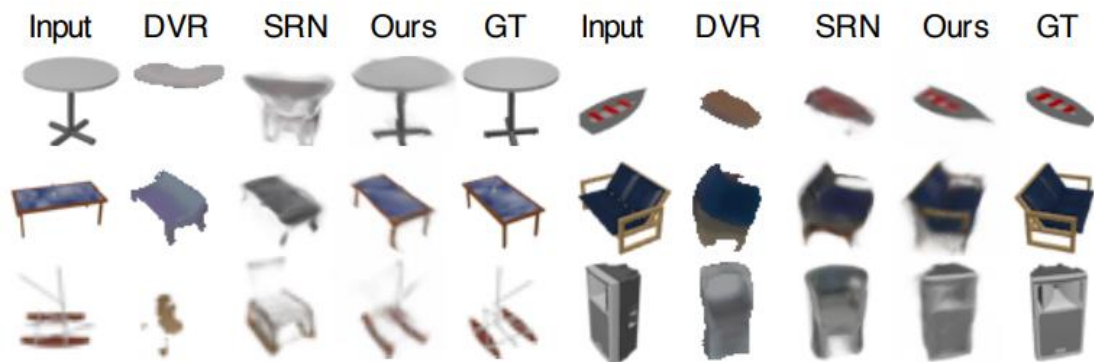


Figure 6: **Generalization to unseen categories.** We evaluate a model trained on planes, cars, and chairs on 10 unseen ShapeNet categories. We find that the model is able to synthesize reasonable views even in this difficult case.

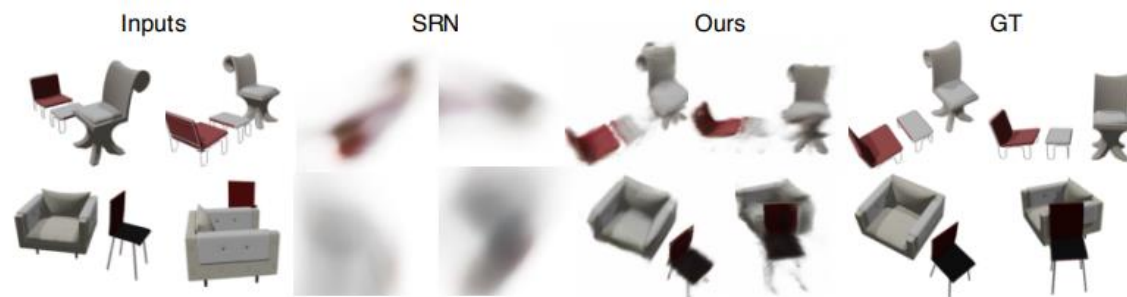


Figure 7: **360° view prediction with multiple objects.** We show qualitative results of our method compared with SRN on scenes composed of multiple ShapeNet chairs. We are easily able to handle this setting, because our prediction is done in view space; in contrast, SRN predicts in canonical space, and struggles with scenes that cannot be aligned in such a way.

Experiments

- Scene Prior on Real Image
 - DTU MVS dataset

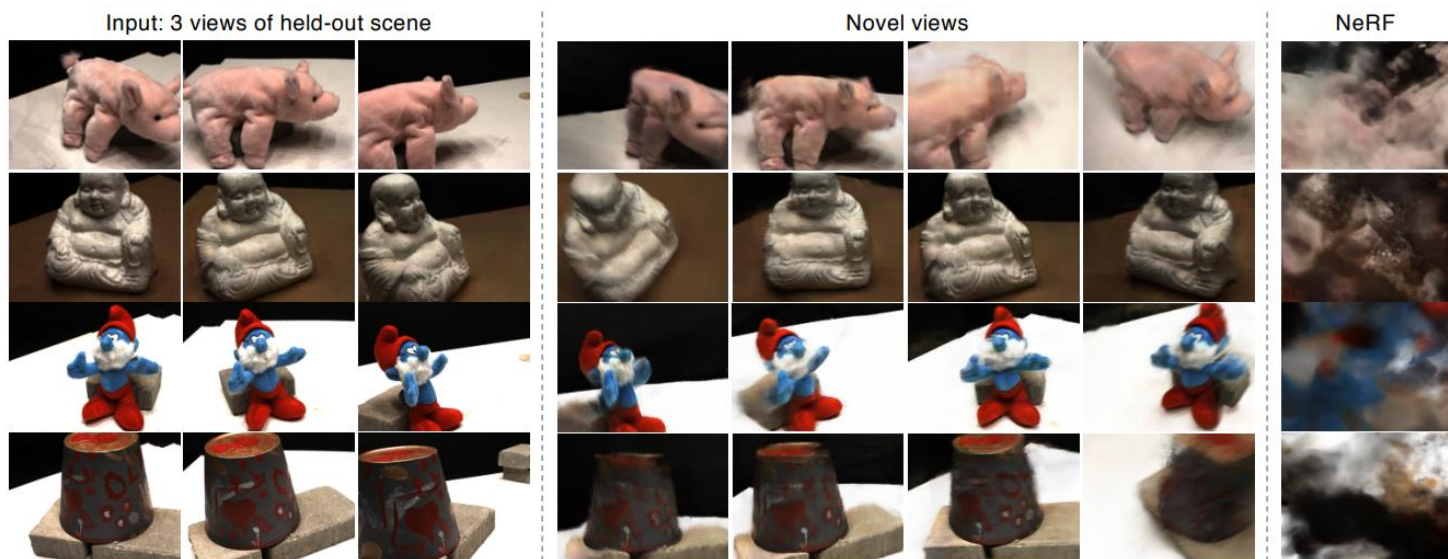


Figure 9: **Wide baseline novel-view synthesis on a real image dataset.** We train our model to distinct scenes in the DTU MVS dataset [14]. Perhaps surprisingly, even in this case, our model is able to infer novel views with reasonable quality for held-out scenes without further test-time optimization, all from only three views. Note the train/test sets share no overlapping scenes.

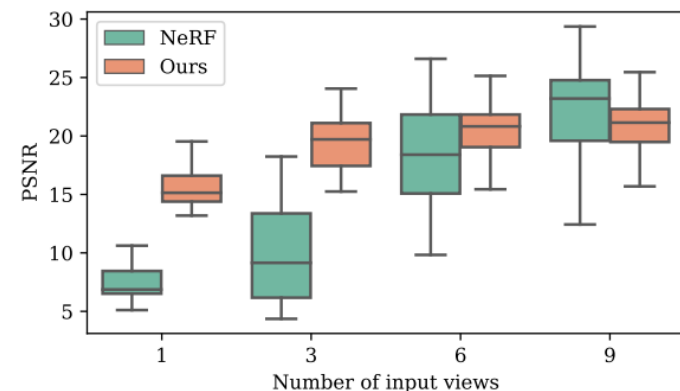


Figure 10: **PSNR of few-shot feed-forward DTU reconstruction.** We show the quantiles of PSNR on DTU for our method and NeRF, given 1, 3, 6, or 9 input views. Separate NeRFs are trained per scene and number of input views, while our method requires only a single model trained with 3 encoded views.

Additional Qualitative Results

- <https://alexYu.net/pixelnerf/>

Depth-supervised NeRF: Fewer Views and Faster Training for Free

Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan

CMU, Google, and Argo AI

Arxiv

Presenter: Minho Park

Task

- One common failure mode of Neural Radiance Field (NeRF) models is fitting incorrect geometries when given an insufficient number of input views.
- Without any priors, the differential density is uniformly allocated across the volume.
- However, this is also a poor initialization as in typical scenes, the majority of space is either occupied or empty.

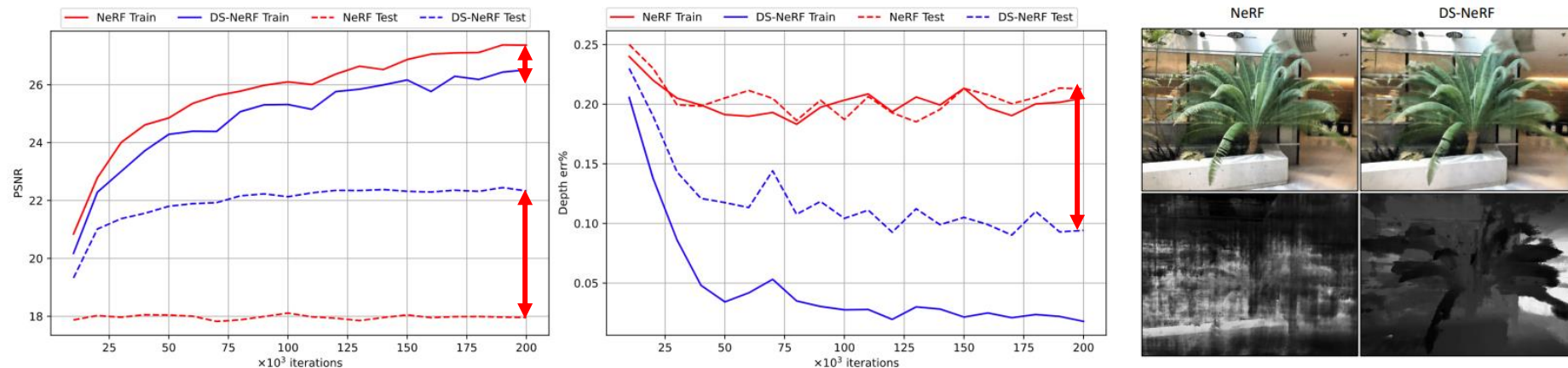


Figure 2: **NeRF can overfit** to a small number of training views (here, two), as shown by the PSNR gap between train and test view renderings (left). This overfitting is due to wildy inaccurate geometries being learned, as evidenced by the large error of the rendered depth maps, even for training views (middle). We visualize both the rendered image and the rendered depth map for a training view on the right. DS-NeRF exploits (sparse) depth supervision to learn far more accurate scene geometry, resulting in better generalization to novel views.

Motivation

- Need more views or some regularizer.
- Current NeRF pipelines require images with known camera poses.
 - Typically estimated by running structure-from-motion (SfM).
- Takes advantage of readily-available depth supervision.
- The sparse depth supervision can be used to regularize the learned geometry.

Depth-supervised NeRF (DS-NeRF)

- Method overview

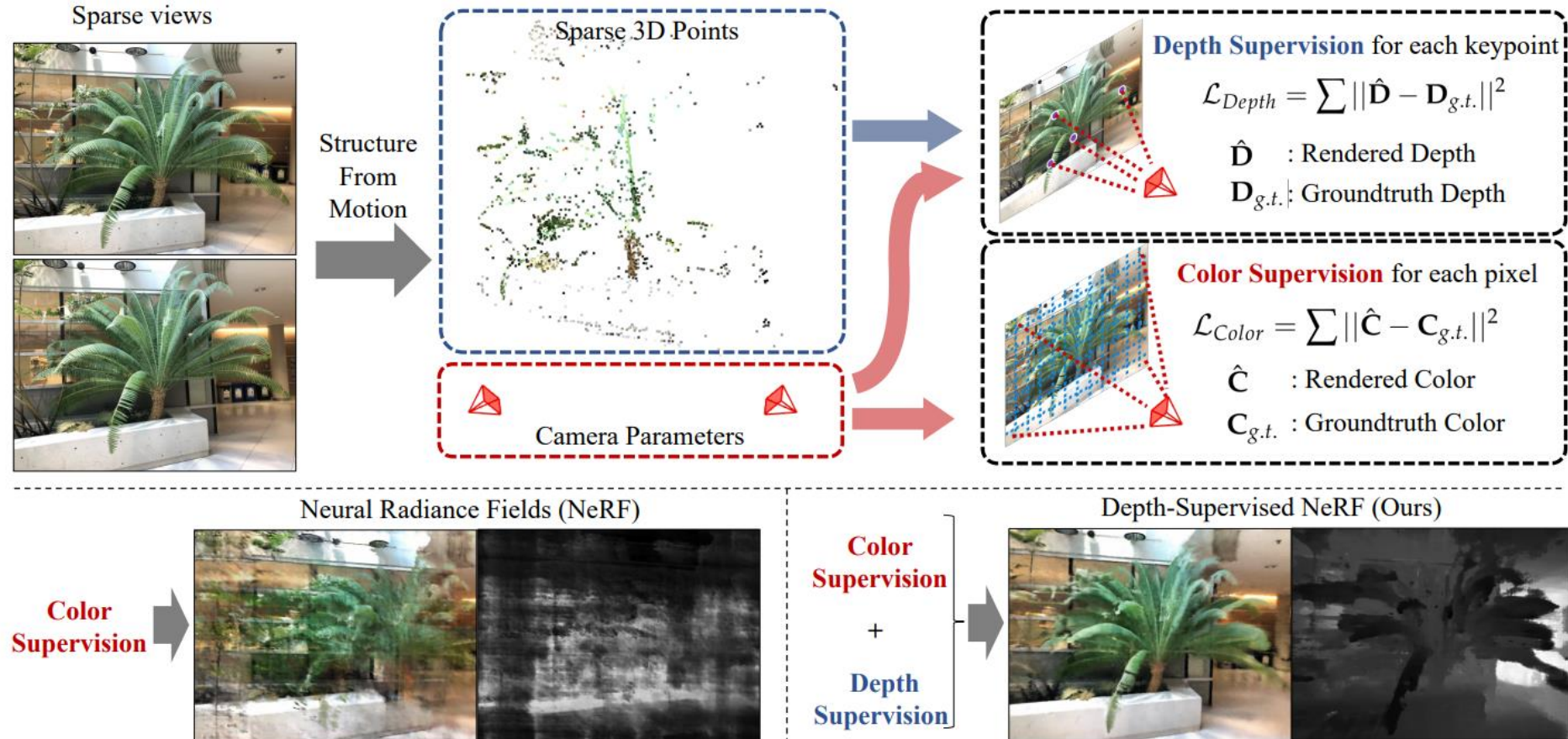


Figure 1: Because NeRFs can be difficult to train given insufficient color supervision (i.e., too few input images), we make use of additional supervision from depth.

Deriving Depth

- In order to train a NeRF, camera poses are estimated using Structure-from-Motion (SfM) frameworks. e.g., COLMAP
- It simultaneously estimate the extrinsic camera pose \mathbf{P} and intrinsics \mathbf{K} from a collection of imagery. i.e., 3D keypoints $\{\mathbf{X}: \mathbf{x}_1, \mathbf{x}_2, \dots \in \mathbb{R}^3\}$.

Depth Supervision Loss

- Given a ray parameterized as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$
- Volumetric rendering: $\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\mathbf{c}(t)dt$ where $T(t) = \exp\left(-\int_{t_n}^t \sigma(t)dt\right)$.
- Depth $\hat{\mathbf{D}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)dt$
- Then loss is

$$\mathcal{L}_{\text{Depth}} = \sum_{x_i \in X_j} w_i |\hat{\mathbf{D}}(\mathbf{r}_{ij}) - (P_j x_i) \cdot [0,0,1]|^2$$

- where i -th keypoint and j -th camera.

Confidence-weighted Keypoints

- SfM might produce spurious correspondences and/or unreliable keypoints.
- They weight each keypoint by its reprojection error estimated by SfM.
- Reprojection error: $e_{ij} = \text{distance}(\mathbf{K}_j \mathbf{P}_j \mathbf{x}_i, \text{detected keypoints in 2D})$
- $e_i = \sum_j e_{ij}$ for each keypoint.

$$w_i = \exp\left(-\left(\frac{e_i}{\bar{e}}\right)^2\right)$$

- Finally training loss is $\mathcal{L} = \mathcal{L}_{\text{Color}} + \lambda_D \mathcal{L}_{\text{Depth}}$.

References

- Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *European conference on computer vision*. Springer, Cham, 2020.
- Gofo, "Rendering Pipeline, Transformation Pipeline" [KR], <https://gofo-coding.tistory.com/entry/Rendering-Pipeline-Modeling-Viewing-Projection-Viewport>
- Kyle Simek, "Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix", <https://ksimek.github.io/2013/08/13/intrinsic/>