CSC2457 3D & Geometric Deep Learning:

NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections

Ricardo Martin-Brualla*, Noha Radwan*, Mehdi S. M. Sajjadi*,
Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth

Google Research

2021-02-23

Presenter: Gary Leung

Instructor: Animesh Garg

UNIVERSITY OF TORONTO

Problem Setting – Novel View Synthesis

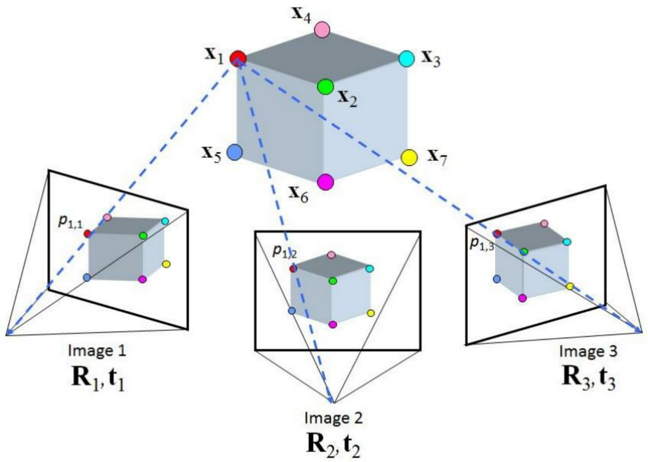
Goal:

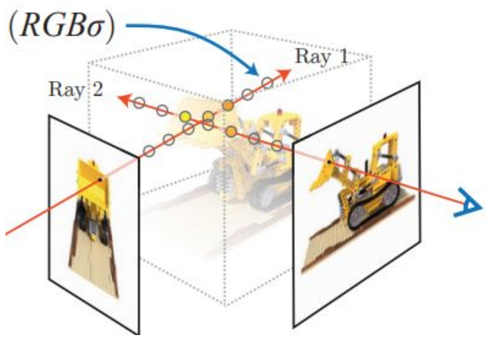
Photo Collection $\{\mathcal{I}_i\}_{i=1}^{N}$

3D Representation

Generated Novel View

Approaches for Novel View Synthesis





Structure from Motion

Bundle Adjustment

Re-render from Inputs

Neural Radiance Fields

Neural Rendering

Classic Approaches

Recent Approaches

Novel View Synthesis for Unconstrained Photo Collections is Hard!

Constrained Collection

Unconstrained Collection

Constrained Collection		Unconstrained Collection
✓	Constant Illumination	✗
✓	Static Objects	✗
✓	Camera Consistency	✗

Contributions – NeRF-W

- Prior Work:
  - **Neural Rerendering in the Wild (NRW):**
    - Approach results in checkerboard and temporal artifacts under camera motion.
  - **Neural Radiance Fields (NeRF):**
    - Strict consistency assumptions result in inaccuracies when applied to photos in the wild
- NeRF-W proposes:
  - An extension to NeRF capable of dealing with photometric and environmental variations.
- Compared to past work, NeRF-W demonstrates:
  - Higher performance on image quality metrics such as PSNR and MS-SSIM.
  - Smoother appearance interpolation and temporal consistency in the presence of appearance variation.
  - Similar performance to NeRF in controlled settings.

Problem Setting - NeRF

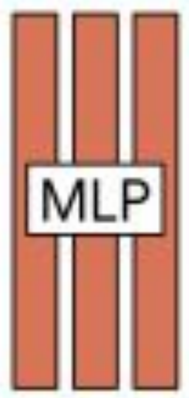
We represent the scene using a learned, continuous volumetric radiance field:

3D Position

$\mathbf{x} = (x, y, z)$

Unit Norm Viewing Direction

$\mathbf{d} = (d_x, d_y, d_z)$



Color

$\mathbf{c} = (r, g, b)$

Volume Density

σ

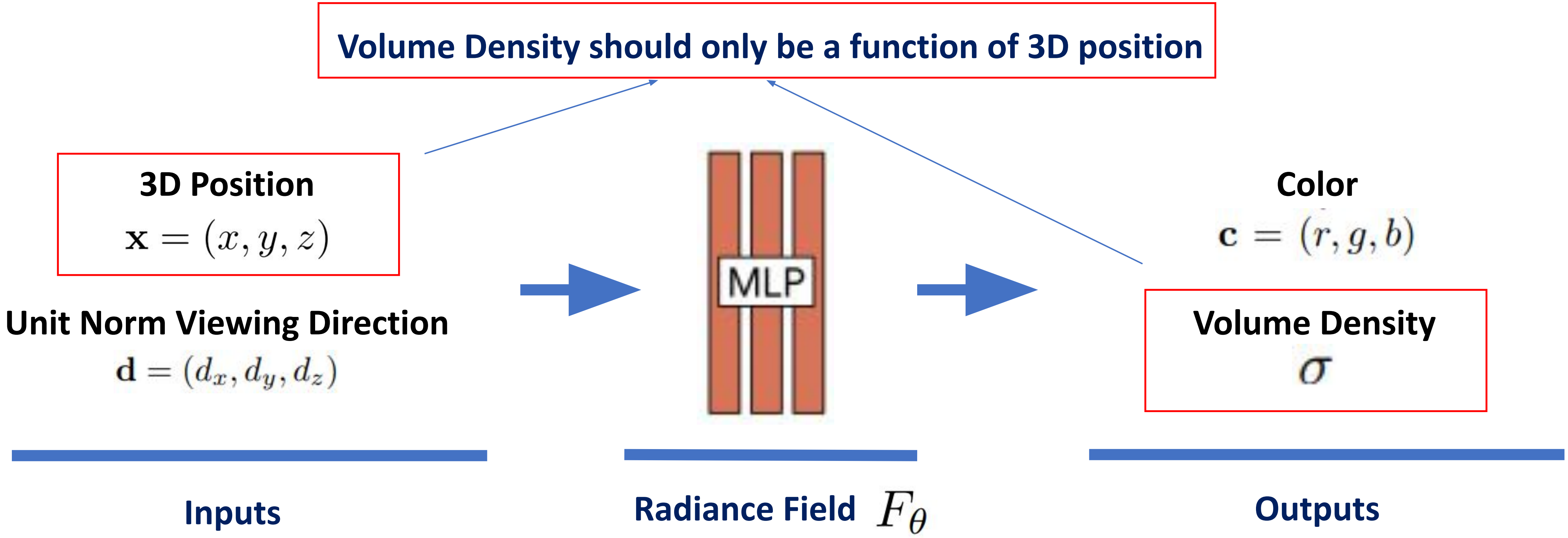
Inputs

Radiance Field F_θ

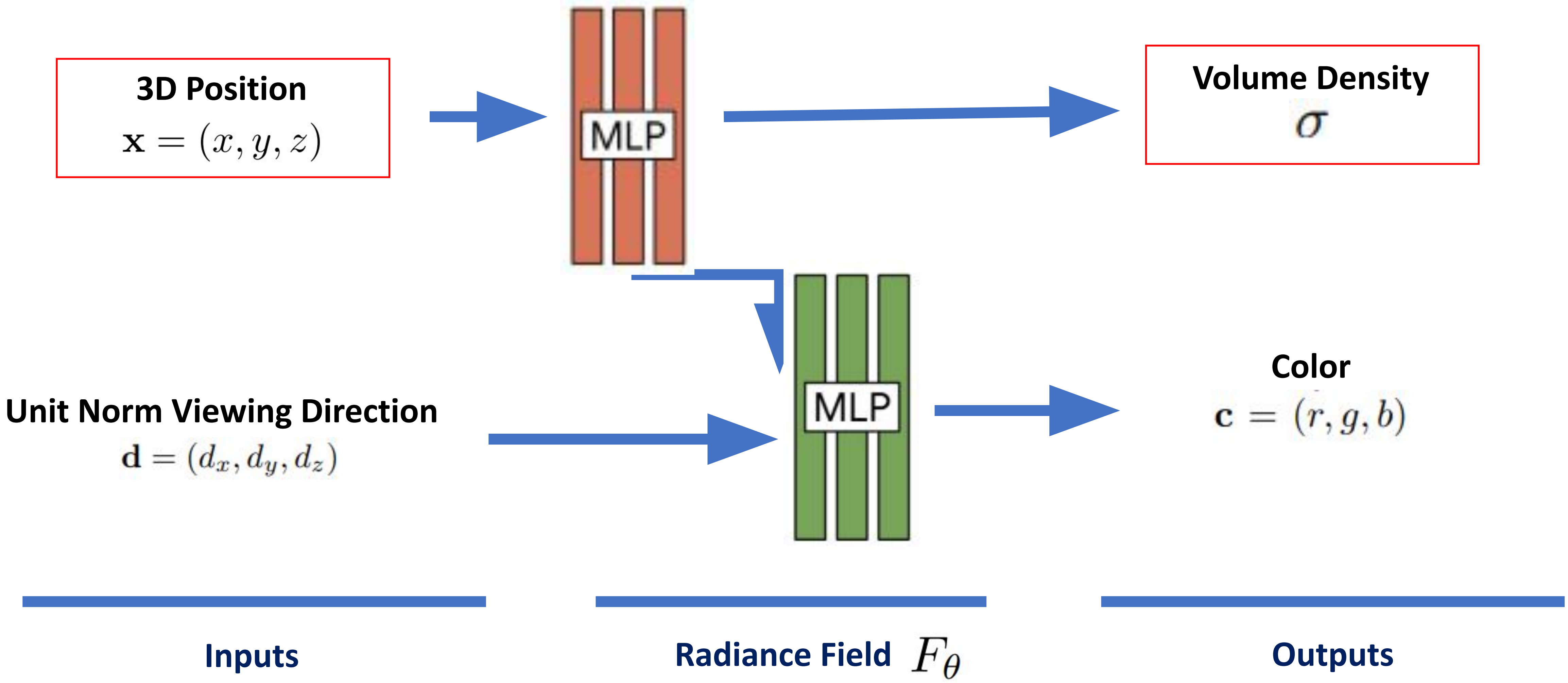
Outputs

NeRF – Multiview Consistency

- One of the constraints NeRF has is multiview consistency



NeRF – Multiview Consistency



NeRF – Color Computation

We essentially have a ray from which we can get color $\mathbf{c}(t)$ and density $\sigma(t)$ at any point within the ray.

Scene Pixel

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$$

Camera o

Volume Density

σ

Probability of ray terminating at time t

NeRF – Color Computation

Given a ray, we can compute the color of a single pixel using volumetric rendering.

$$\hat{\mathbf{C}}(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma)$$

Predicted Color of Pixel

Volumetric rendering of color given density

NeRF – Color Computation

Expected Color

$$\underline{\hat{\mathbf{C}}(\mathbf{r})} = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma)$$

Far Bound of time

Color at ray given viewing direction d

$$= \int_{t_n}^{t_f} T(t)\sigma(\underline{\mathbf{r}(t)})\mathbf{c}(\underline{\mathbf{r}(t), \mathbf{d}})dt\,, \text{ where } \underline{T(t) =} \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

Near Bound of time

Probability of ray terminating at location r(t)

Probability ray travels from tn to t without terminating

Expected color between two time bounds, calculated by the color at a point weighted by the probability of the ray reaching that point

NeRF – Color Computation Estimation

$$\hat{\mathbf{C}}(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma)$$

We estimate this integral using quadrature:

Color at ray given viewing direction d

$$= \sum_{i=1}^{N} T_i \underbrace{(1 - \exp(-\sigma_i \delta_i))}_{} \underbrace{\mathbf{c}_i}_{}\,, \text{ where } \underbrace{T_i}_{} = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$$

Cumulative Distribution Function of ray terminating

Probability ray travels from j=1 to i-1 without terminating

$$\boxed{\delta_i = t_{i+1} - t_i}$$

NeRF – Color Computation Estimation

$$\hat{\mathbf{C}}(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma)$$

Within the NeRF-W paper, this formula is formatted like this:

Color at ray given viewing direction d

$$= \sum_{k=1}^{K} T(t_k)\, \alpha(\sigma(t_k)\delta_k)\, \mathbf{c}(t_k)$$

$$T(t_k) = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$$

Cumulative Distribution Function of ray terminating

Probability ray travels from j=1 to i-1 without terminating

$$\boxed{\delta_i = t_{i+1} - t_i}$$

$$\boxed{\alpha(x) = 1 - \exp(-x)}$$

NeRF - Loss

- We can optimize directly for the reconstruction loss:

$$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2$$

Problem:

Densely estimating the integral from N query points along every camera ray is inefficient!

We don’t want to repeatedly sample free space and occluded regions.

NeRF - Loss

- Solution:
  - We optimize simultaneously for a coarse and a fine network.

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right]$$

  - Coarse Network
    - Trained using a first set of N locations sampled in a stratified manner.
  - Fine Network
    - Trained with additional locations sampled on the ray in a weighted manner using the coarse network's predicted volume density.

NeRF – Problem

- Assumes consistency in that the same 3D position and viewing directions in two images should result in the same color/density.
  - This does not hold in the wild!

Transient objects such as people

Photometric variation such as camera lighting

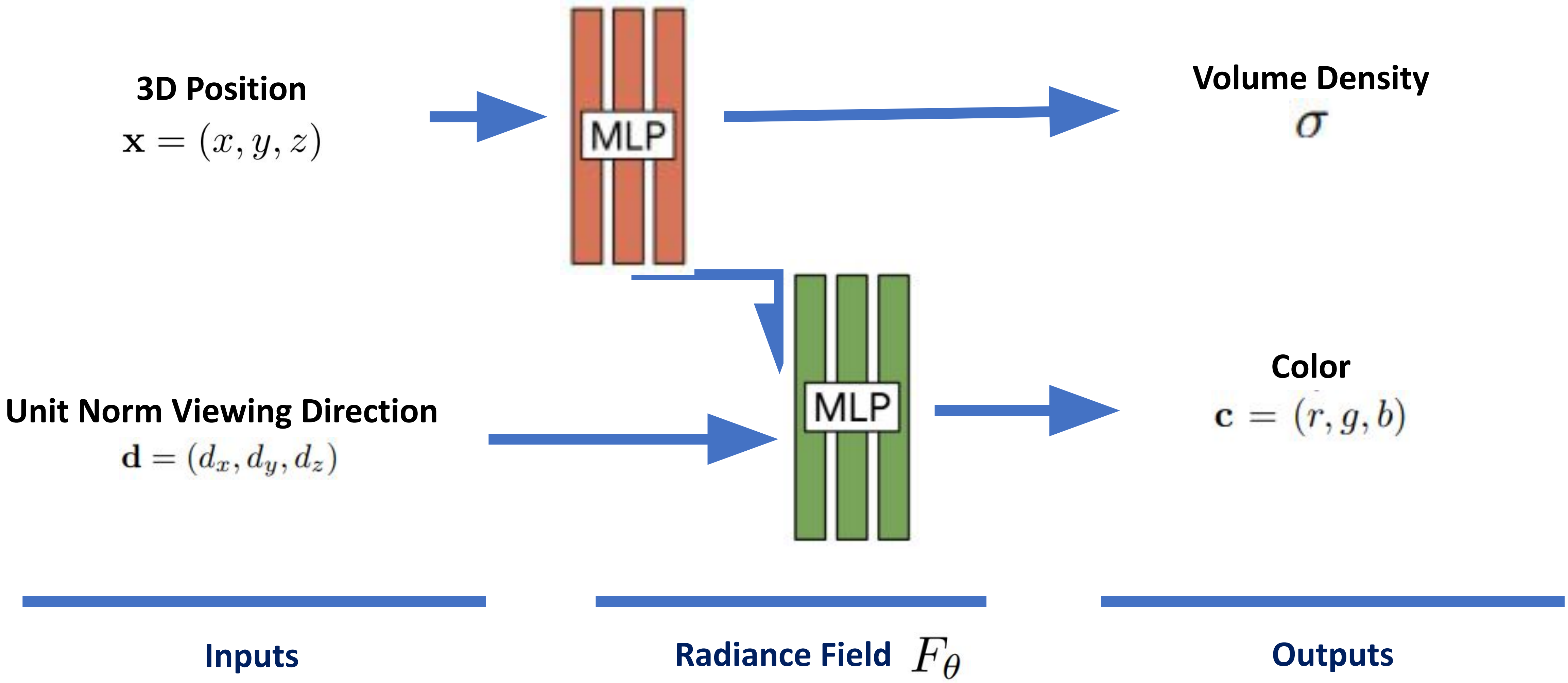
Nerf-W(ild) - Introduction

- Adapts NeRF to variable lightning and photometric changes by introducing a dependence on the images $\{\mathcal{I}_i\}_{i=1}^{N}$.

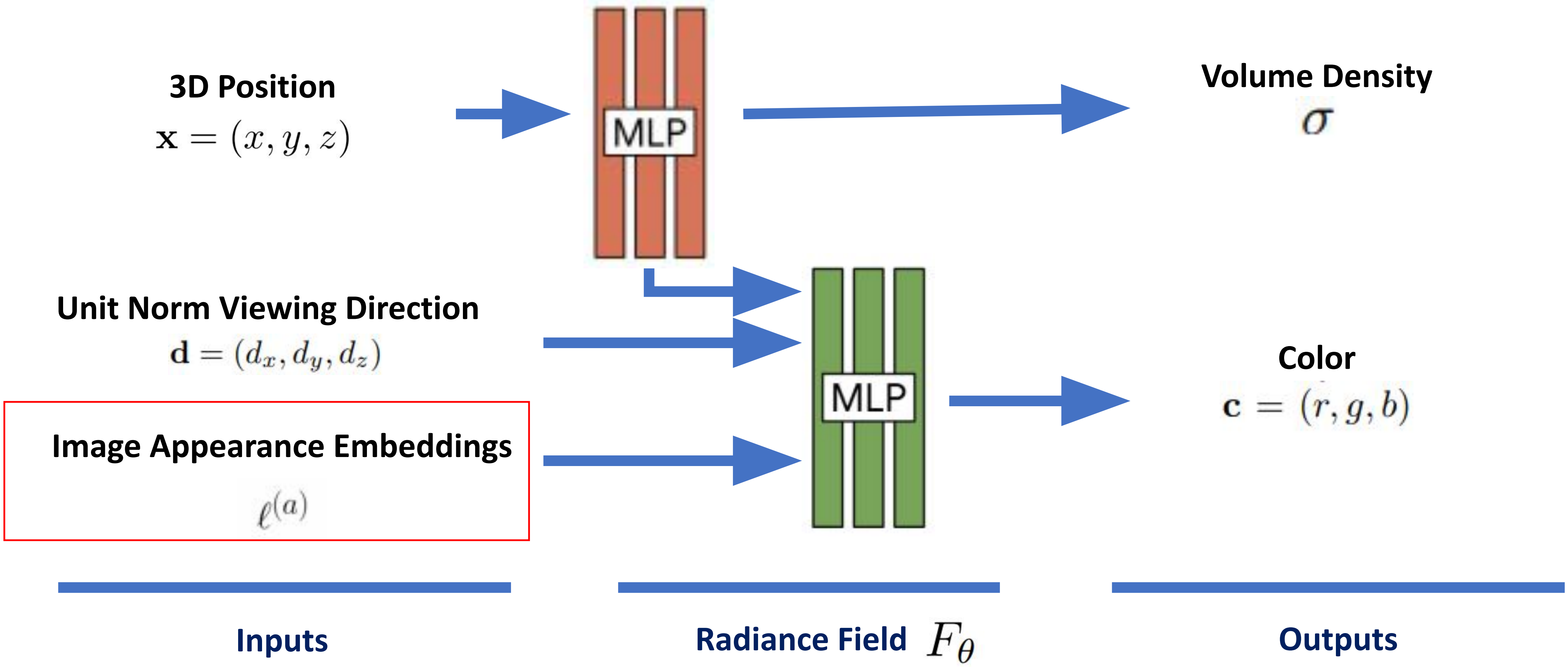
$$\hat{\mathbf{C}}(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma) \longrightarrow \hat{\mathbf{C}}_i(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}_i, \sigma)$$

NeRF **NeRF-W**

NeRF-W – Image Latents

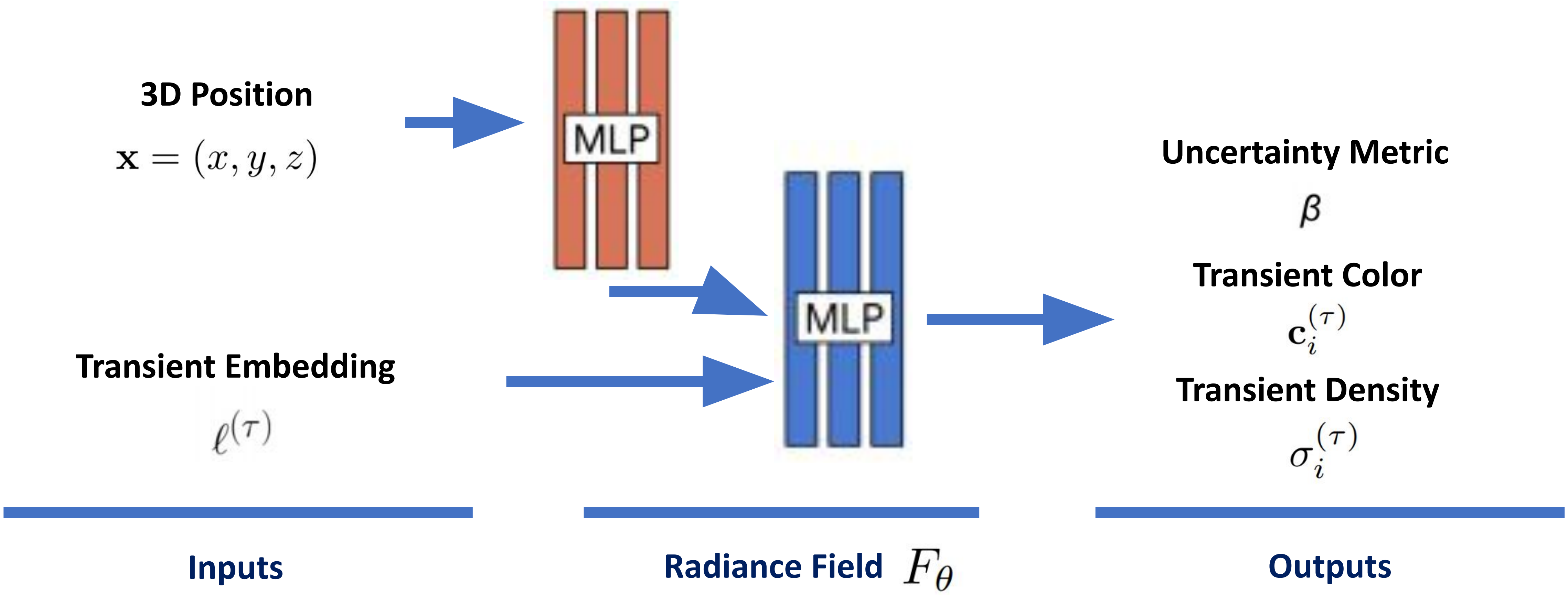


NeRF-W – Static Image Latents

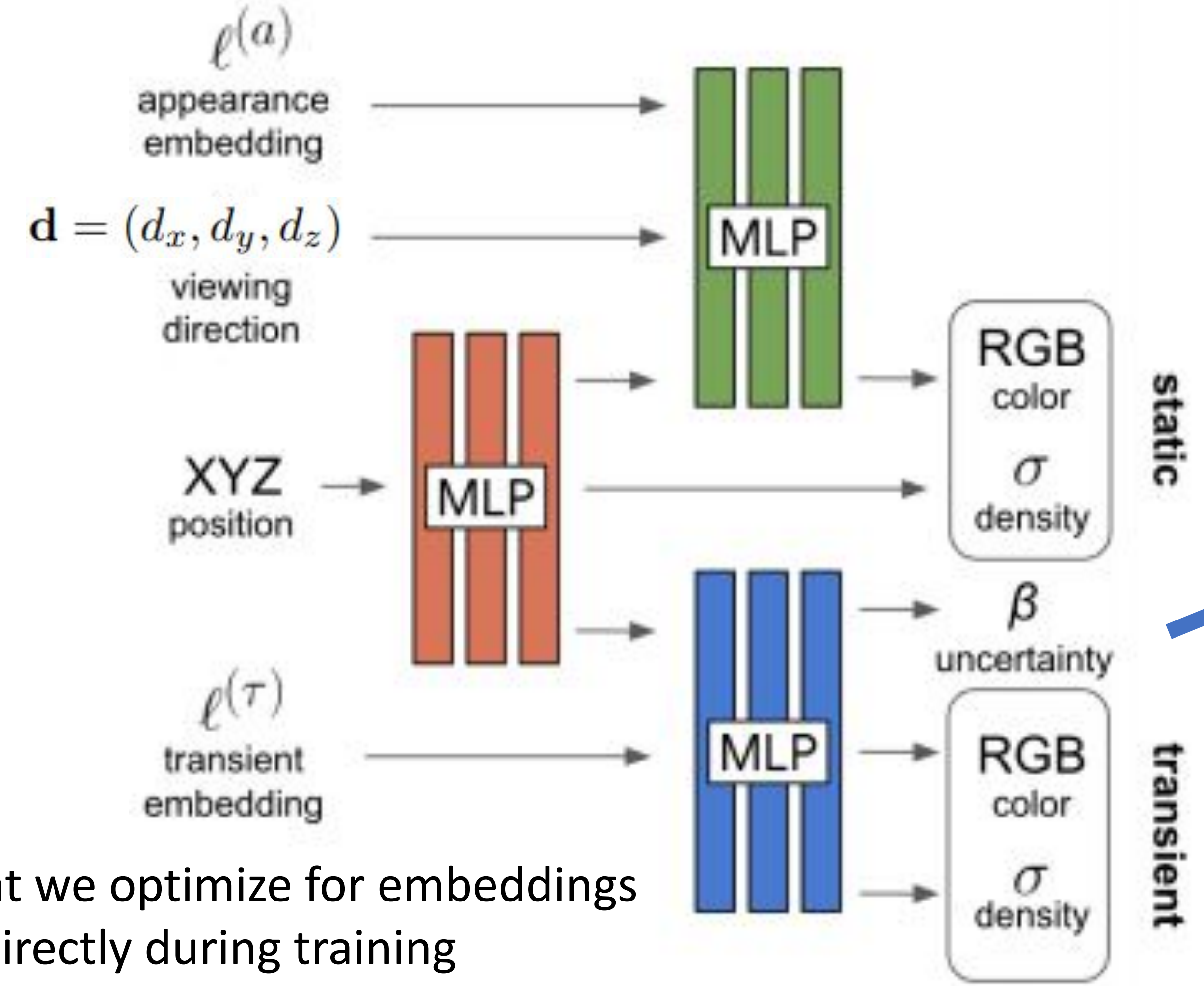


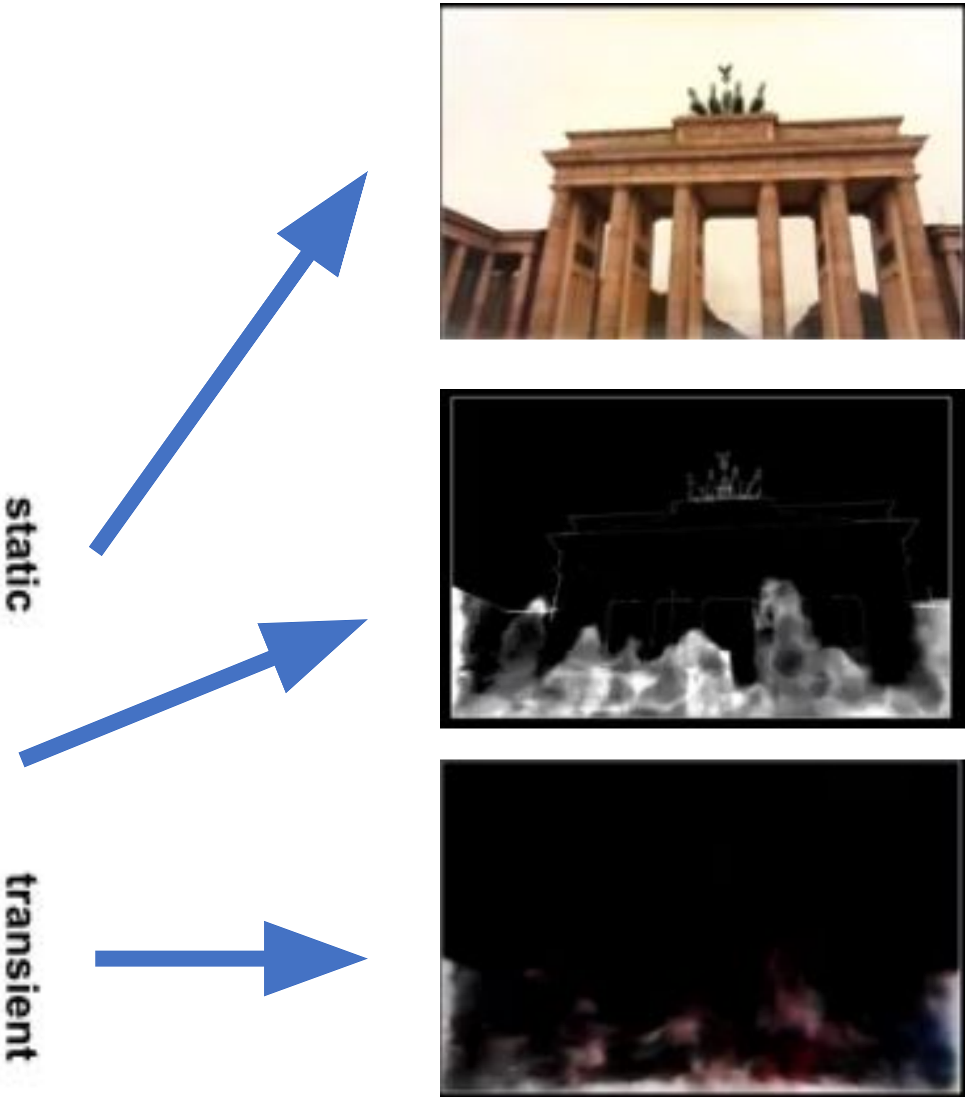
NeRF-W – Transiency

- NeRf-W also adds a second head to model transient phenomena



NeRF-W – Whole Model





*Note that we optimize for embeddings directly during training

NeRF-W – Training

NeRF-W - Math

- Adapts NeRF to variable lightning and photometric changes by introducing a dependence on the images $\{\mathcal{I}_i\}_{i=1}^N$.

$$\hat{\mathbf{C}}(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma) \longrightarrow \hat{\mathbf{C}}_i(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}_i, \sigma)$$

NeRF **NeRF-W**

NeRF-W - Math

$$\hat{\mathbf{C}}(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma)$$

$$= \sum_{k=1}^{K} T(t_k)\,\alpha(\sigma(t_k)\delta_k)\,\mathbf{c}(t_k)$$

$$\text{where} \quad T(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma(t_{k'})\delta_{k'}\right)$$

σ Volume Density

$\delta_i = t_{i+1} - t_i$

$\alpha(x) = 1 - \exp(-x)$

NeRF

$$\hat{\mathbf{C}}_i(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}_i, \sigma)$$

Any ideas on how the our volumetric rendering is changed with the addition of our transient density and color?

NeRF-W

NeRF-W - Math

$$\hat{\mathbf{C}}(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma)$$

$$= \sum_{k=1}^{K} T(t_k)\, \alpha(\sigma(t_k)\delta_k)\, \mathbf{c}(t_k)$$

$$\text{where} \quad T(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma(t_{k'})\delta_{k'} \right)$$

σ Volume Density

$\delta_i = t_{i+1} - t_i$

$\alpha(x) = 1 - \exp(-x)$

NeRF

$$\hat{\mathbf{C}}_i(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}_i, \sigma)$$

$$= \sum_{k=1}^{K} T_i(t_k) \Big(\alpha(\sigma(t_k)\delta_k)\mathbf{c}_i(t_k) + \underline{\alpha\Big(\sigma_i^{(\tau)}(t_k)\delta_k\Big)\mathbf{c}_i^{(\tau)}(t_k)} \Big)$$

$$\text{where } T_i(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \Big(\sigma(t_{k'}) + \underline{\sigma_i^{(\tau)}(t_{k'})}\Big)\delta_{k'} \right)$$

Alpha Composite of
Transient Counterparts

NeRF-W

NeRF-W - Loss

- Similar to NeRF, we also simultaneously optimize a fine and course model.

$$Loss = \sum_{ij} L_{coarse} + L_{fine}$$

No Transient Portions

	NeRF	NeRF-W
L_{coarse}	$\left\| \mathbf{C}(\mathbf{r}_{ij}) - \hat{\mathbf{C}}^c(\mathbf{r}_{ij}) \right\|_2^2$	$\frac{1}{2}\left\| \mathbf{C}(\mathbf{r}_{ij}) - \hat{\mathbf{C}}_i^c(\mathbf{r}_{ij}) \right\|_2^2$
L_{fine}	$\left\| \mathbf{C}(\mathbf{r}_{ij}) - \hat{\mathbf{C}}^f(\mathbf{r}_{ij}) \right\|_2^2$	$\frac{\left\| \mathbf{C}_i(\mathbf{r}) - \hat{\mathbf{C}}_i(\mathbf{r}) \right\|_2^2}{2\beta_i(\mathbf{r})^2} + \frac{\log \beta_i(\mathbf{r})^2}{2} + \frac{\lambda_u}{K}\sum_{k=1}^{K} \sigma_i^{(\tau)}(t_k)$

NeRF-W - Loss

Uncertainty $\beta_i(t) = \beta_{\min} + \log\left(1 + \exp\left(\underbrace{\tilde{\beta}_i(t)}_{\text{Model Prediction}}\right)\right)$

$$\underbrace{\frac{\left\|\mathbf{C}_i(\mathbf{r}) - \hat{\mathbf{C}}_i(\mathbf{r})\right\|_2^2}{2\beta_i(\mathbf{r})^2} + \frac{\log \beta_i(\mathbf{r})^2}{2}} + \underbrace{\frac{\lambda_u}{K}\sum_{k=1}^{K}\sigma_i^{(\tau)}(t_k)}$$

Negative Log-likelihood of $\mathbf{C}_i(\mathbf{r})$

According to a normal distribution with mean $\hat{\mathbf{C}}_i(\mathbf{r})$ and variance $\beta_i(\mathbf{r})^2$

The larger the variance, the less important the pixel (assumption that it belongs to transient object)

Regularization of Transient Density:

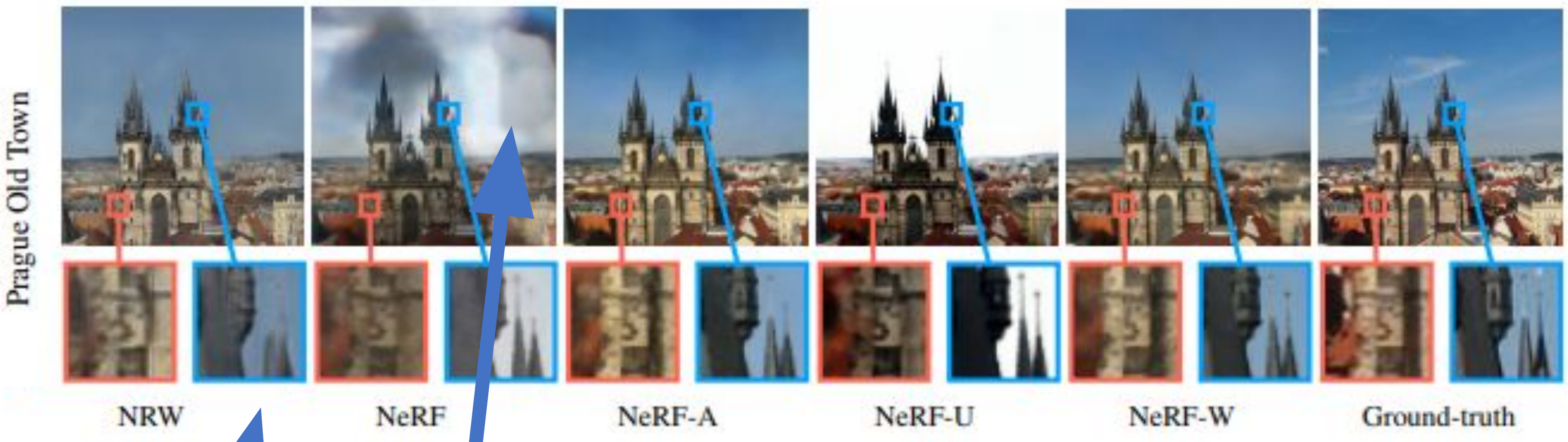
We don't want transient density to explain away static phenomena

NeRF-W

NeRF-W Evaluation Setup

- Evaluation Dataset:
  - Photo Tourism dataset – Image collection of famous landmarks in the wild
- Evaluation Metrics:
  - PSNR (peak signal-to-noise ratio)
  - MS-SSIM (multi-scale structural similarity)
  - LPIPS (learning perceptual image patch similarity)
- Ablations:
  - NeRF-A(ppearance) – NeRF-W without the transient head
  - NeRF-U(ncertainty) – NeRF-W without the appearance embedding
- Test time:
  - Transient head is not used.
  - Because appearance embeddings are optimized only for training images, test images have their appearance embeddings optimized on the left half of the image and are evaluated on the right half.

NeRF-W Results – Prague Old Town

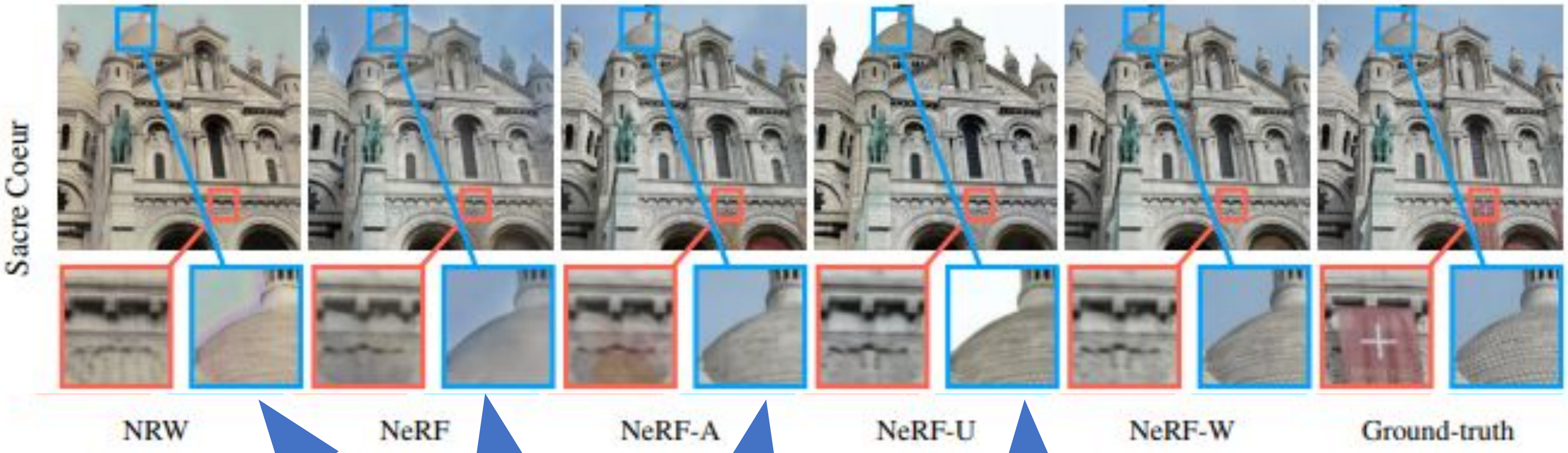


	PRAGUE		
	PSNR	MS-SSIM	LPIPS
NRW [22]	19.89	0.803	0.216
NERF	15.67	0.747	0.362
NERF-A	22.52	0.870	0.244
NERF-U	15.03	0.787	0.315
NERF-W	22.81	0.879	0.227

NRW sensitive to upstream errors In 3D geometry.

NeRF has foggy renderings and is prone to global color shifts

NeRF-W Results – Sacre Coeur

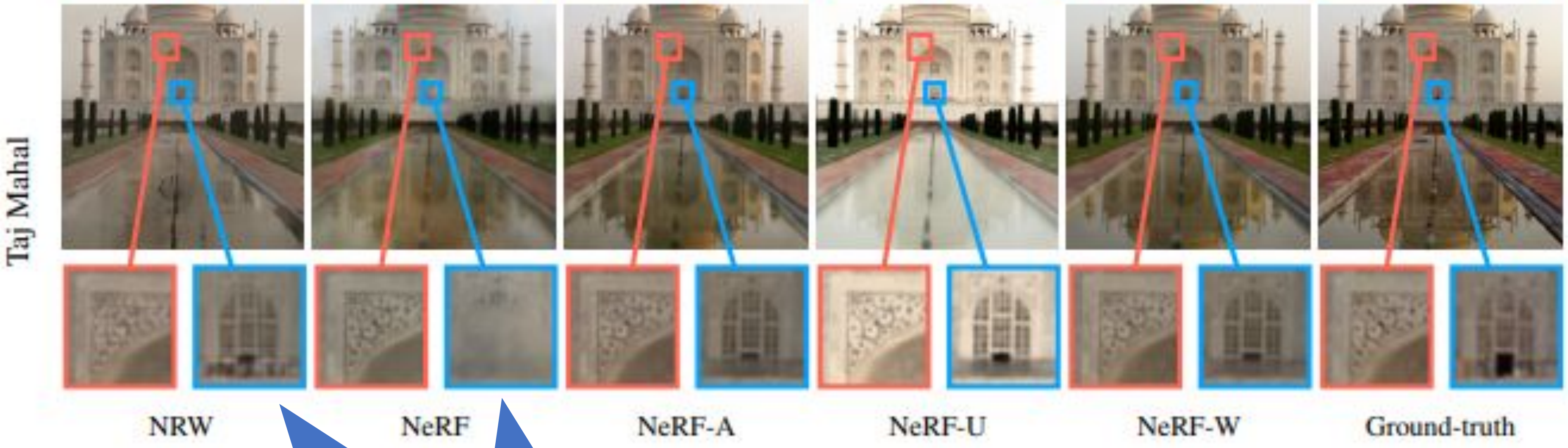


	SACRE COEUR		
	PSNR	MS-SSIM	LPIPS
NRW [22]	19.39	0.797	0.229
NERF	17.12	0.781	0.278
NERF-A	24.43	0.923	0.174
NERF-U	15.99	0.826	0.223
NERF-W	**25.34**	**0.939**	**0.151**

Unable to capture fine brickwork details.

Fine details, but unable to capture photometric effects

NeRF-W Results – Taj Mahal



	Taj Mahal		
	PSNR	MS-SSIM	LPIPS
NRW [22]	21.24	0.844	**0.201**
NeRF	15.77	0.697	0.427
NeRF-A	25.99	0.893	0.225
NeRF-U	10.23	0.778	0.373
NeRF-W	**26.36**	**0.904**	0.207

Blurry details once again on wall engravings and the window.

NeRF-W Results – Full Qualitative Table

	Brandenburg Gate			Sacre Coeur			Trevi Fountain			Taj Mahal			Prague			Hagia Sophia		
	PSNR	MS-SSIM	LPIPS	PSNR	MS-SSIM	LPIPS	PSNR	MS-SSIM	LPIPS	PSNR	MS-SSIM	LPIPS	PSNR	MS-SSIM	LPIPS	PSNR	MS-SSIM	LPIPS
NRW [22]	23.85	0.914	0.141	19.39	0.797	0.229	20.56	0.811	0.242	21.24	0.844	**0.201**	19.89	0.803	**0.216**	20.75	0.796	**0.231**
NeRF	21.05	0.895	0.208	17.12	0.781	0.278	17.46	0.778	0.334	15.77	0.697	0.427	15.67	0.747	0.362	16.04	0.749	0.338
NeRF-A	27.96	0.941	0.145	24.43	0.923	0.174	26.24	0.924	0.211	25.99	0.893	0.225	22.52	0.870	0.244	21.83	0.820	0.276
NeRF-U	19.49	0.921	0.174	15.99	0.826	0.223	15.03	0.795	0.277	10.23	0.778	0.373	15.03	0.787	0.315	13.74	0.706	0.376
NeRF-W	**29.08**	**0.962**	**0.110**	**25.34**	**0.939**	**0.151**	**26.58**	**0.934**	**0.189**	**26.36**	**0.904**	0.207	**22.81**	**0.879**	0.227	**22.23**	**0.849**	0.250

NeRF-W consistency outperforms NRW and NeRF on image quality metrics across the Photo Tourism dataset!

LPIPS (a perceptual metric) favors high-frequency texture reconstruction which is not explicitly trained for by NeRF-W due to not having a perceptual loss.

Qualitative Results – Temporal Consistency

One major improvement compared to prior works is in the temporal consistency when moving the viewpoint across time.

Temporal consistency shown in epipolar plane images– the camera is translated left to right in a straight line, and the pixels at the red line at each time step are stacked on top of each other.

Notice the lack of temporal artifacts in NeRF-W as compared to NRW and NeRF.

Qualitative Results – Temporal Consistency

We can see the improvement in temporal consistency in NeRF-W as compared to NRW in the following video (4:10)

Appearance Embedding Interpolation

Due to the usage of low-dimension appearance embeddings, the appearance of a viewpoint can be changed between several settings by interpolating the embedding.

Limitations/Critiques

- Rendering quality degrades in areas that are rarely observed during training or are observed in very oblique angles.
- Similar to NeRF, camera calibration errors can cause improperly imaged areas to be blurry.
- Like NeRF, NeRF-W is fixed to the training scene and cannot generalize to novel scenes.
- Test time image embeddings are not obtained in a graceful way – they have to be approximated using training image embeddings.

Contributions – NeRF-W

- Prior Work:
  - **Neural Rerendering in the Wild (NRW):**
    - Approach results in checkerboard and temporal artifacts under camera motion.
  - **Neural Radiance Fields (NeRF):**
    - Strict consistency assumptions result in inaccuracies when applied to photos in the wild
- NeRF-W proposes:
  - An extension to NeRF capable of dealing with photometric and environmental variations.
- Compared to past work, NeRF-W demonstrates:
  - Higher performance on image quality metrics such as PSNR and MS-SSIM.
  - Smoother appearance interpolation and temporal consistency in the presence of appearance variation.
  - Similar performance to NeRF in controlled settings.