NeRFactor:

Neural Factorization of Shape and Reflectance Under an Unknown Illumination

(Xiuming ZHANG et al. SIGGRAPH Asia, 2021)

Lulin Zhang, 10/02/2023

IPGP
Université Paris Cité
cnes
IGN
INSTITUT NATIONAL DE L'INFORMATION GÉOGRAPHIQUE ET FORESTIÈRE

NeRF (Neural Radiance Fields):

Render novel views by optimizing a continuous volumetric function.

Represent a scene using a fully-connected **deep network**.

3D location (x, y, z)
2D direction (θ, φ) ⇨ NeRF ⇨ Volume density σ
Emitted color (r, g, b)

Multi-View Images + SfM

Novel view

Novel view rendering

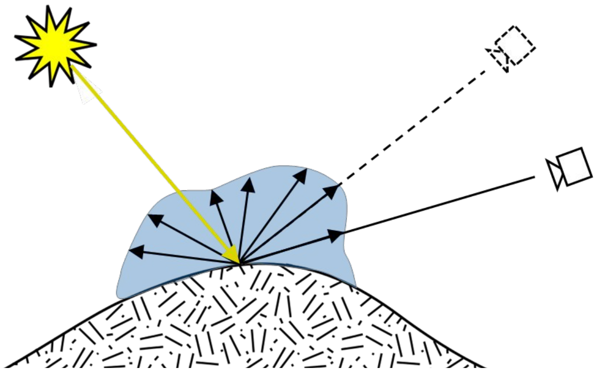
Unable to relight the scene

Outgoing light ~~= incoming light + reflectance~~

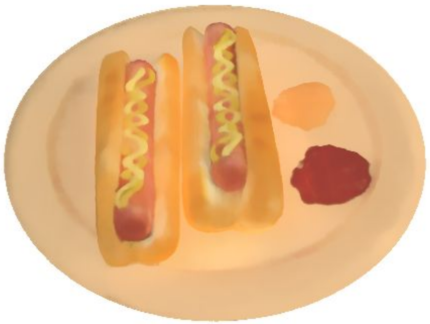
NeRFactor extends NeRF for relighting by modeling:

- Incoming **light**
- Surface **normal**
- **Reflectance**

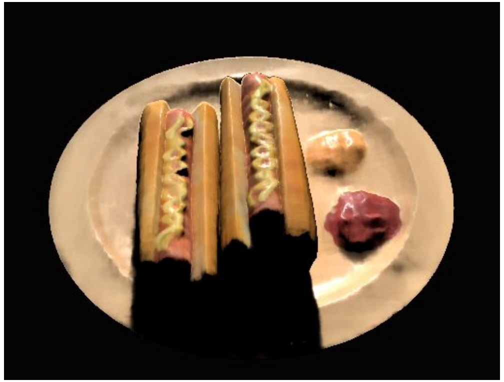
Reflectance:



BRDF: Bi-directional Reflectance Distribution Function



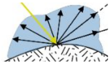
Albedo (Diffuse component)

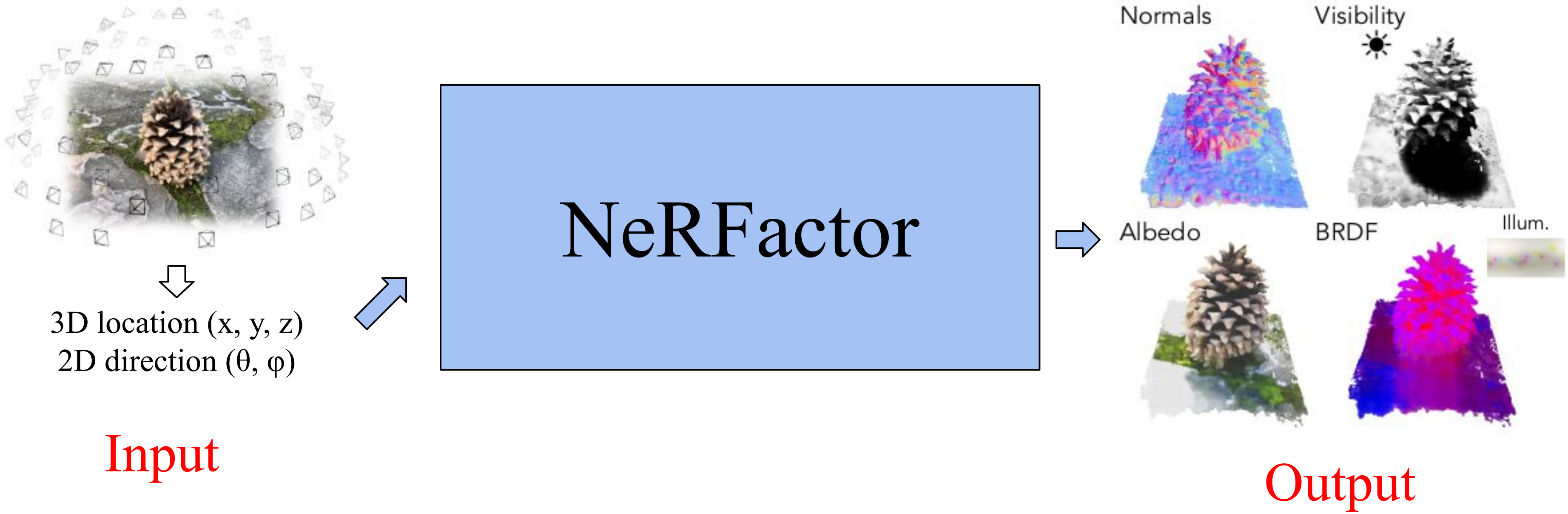


Specular spatially-varying **BRDF**

(related to materials)

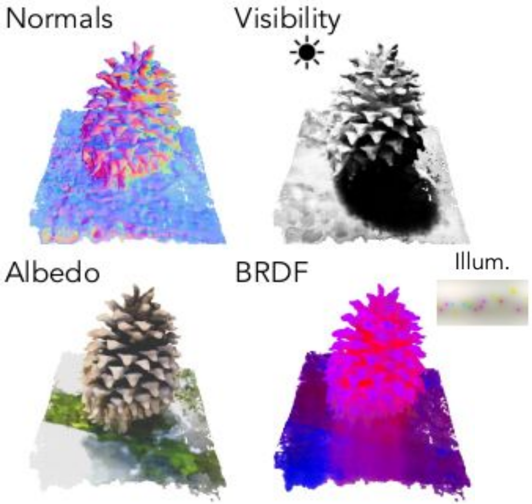
Recovering an object’s geometry and material properties from captured images:

	Previous work	NeRFactor
 **Illumination**	**Multiple known** illumination	**One unknown** illumination
 **BRDF**	**Analytic** BRDF (e.g. microfacet models)	**Data-driven** BRDF
Only images as input	✗ (e.g. scanned geometry)	✓
 **Spatially-varying reflectance**	✗	✓
 **Model visibility or shadows**	✗	✓
 **Represent object with multiple materials**	✗	✓

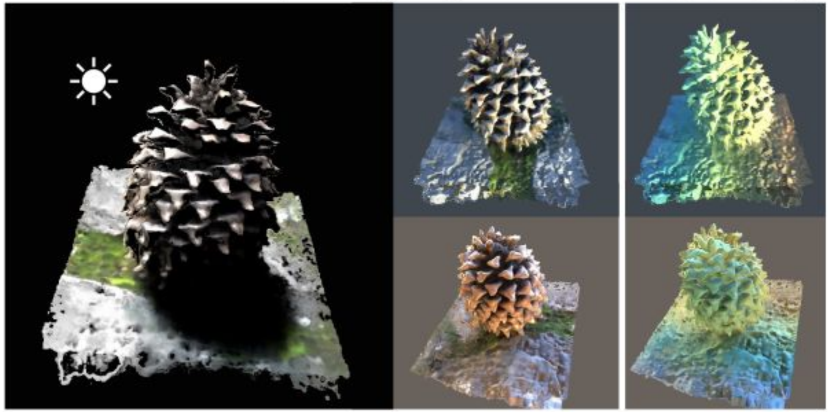

3D location (x, y, z)
2D direction (θ, φ)
Input
NeRFactor
Normals
Visibility
Albedo
BRDF
Illum.
Output

3D location (x, y, z)
2D direction (θ, φ)

NeRFactor



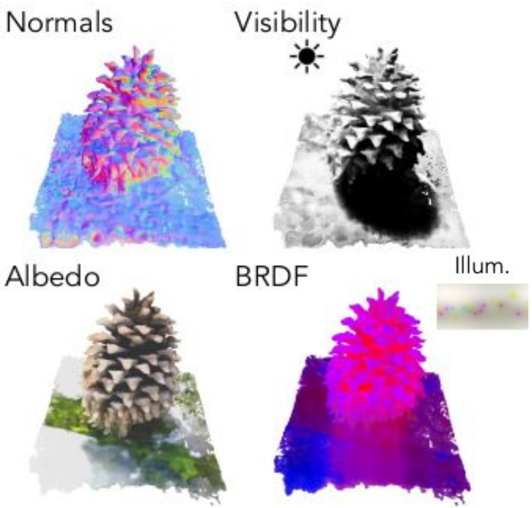
Applications:

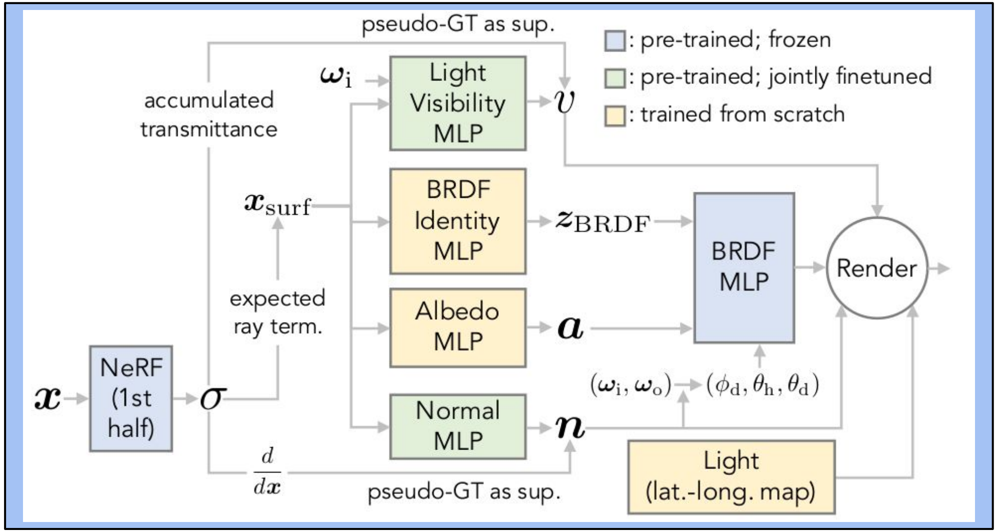


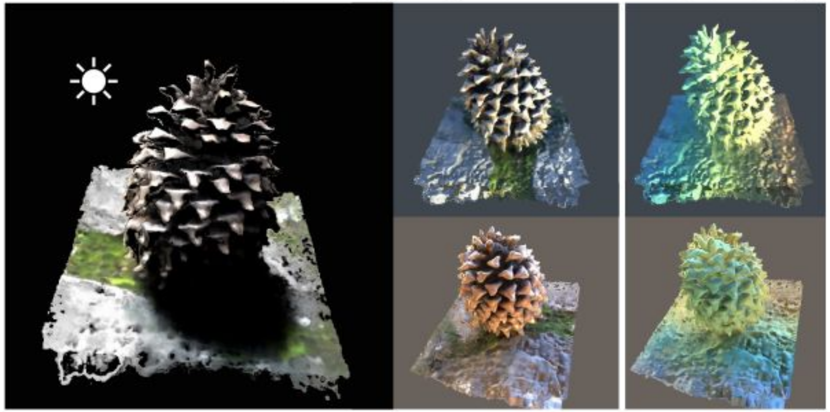
(1) Novel view rendering (2) Relighting (3) Material Editing

3D location (x, y, z)
2D direction (θ, φ)

NeRFactor



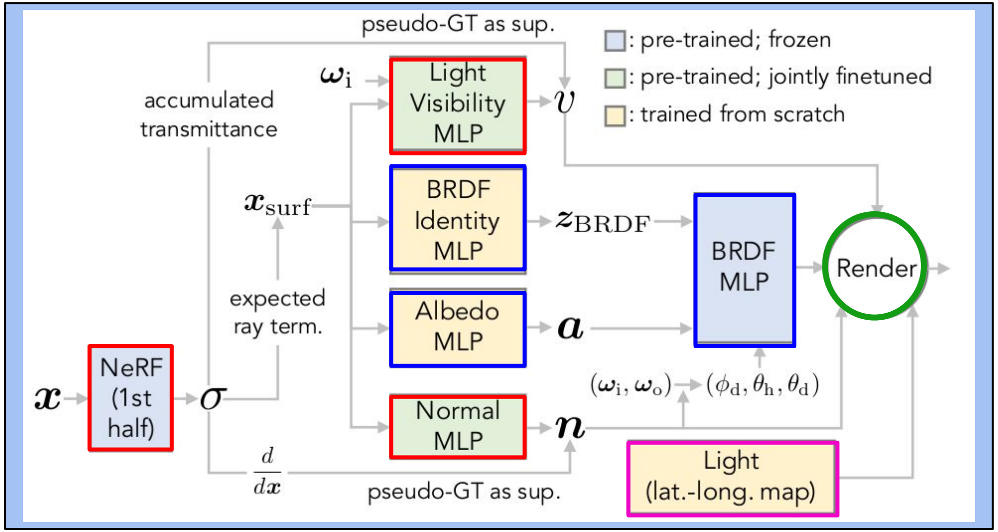




(1) Novel view rendering (2) Relighting (3) Material Editing

NeRFactor includes 4 blocks:

- **Shape**
- **Reflectance**
- **Light**
- **Render**

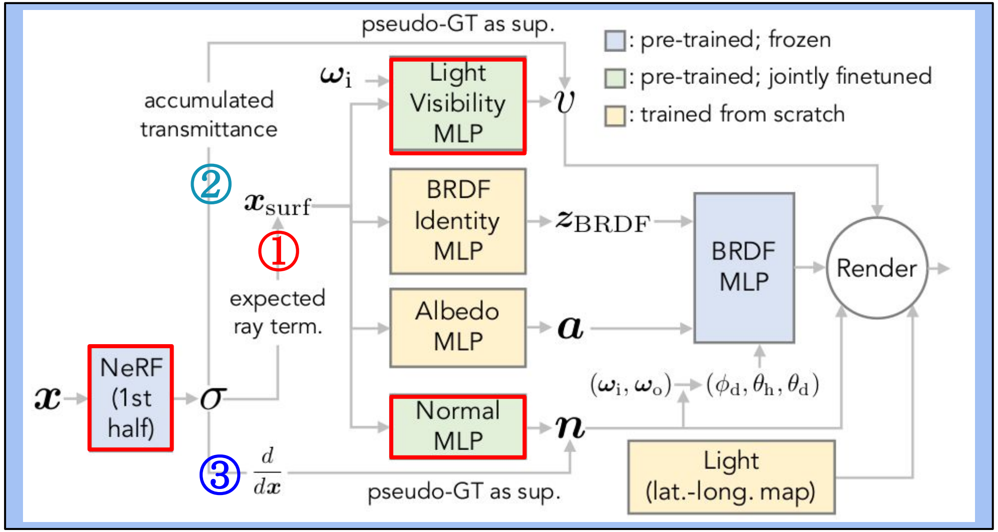


NeRFactor includes 4 blocks:

- **Shape**

  Train **vanilla NeRF**

3D location (x, y, z)
2D direction (θ, φ) ⇨ NeRF ⇨ Volume density σ
~~Emitted color (r, g, b)~~



① **Surface location:**

② **Light visibility:**

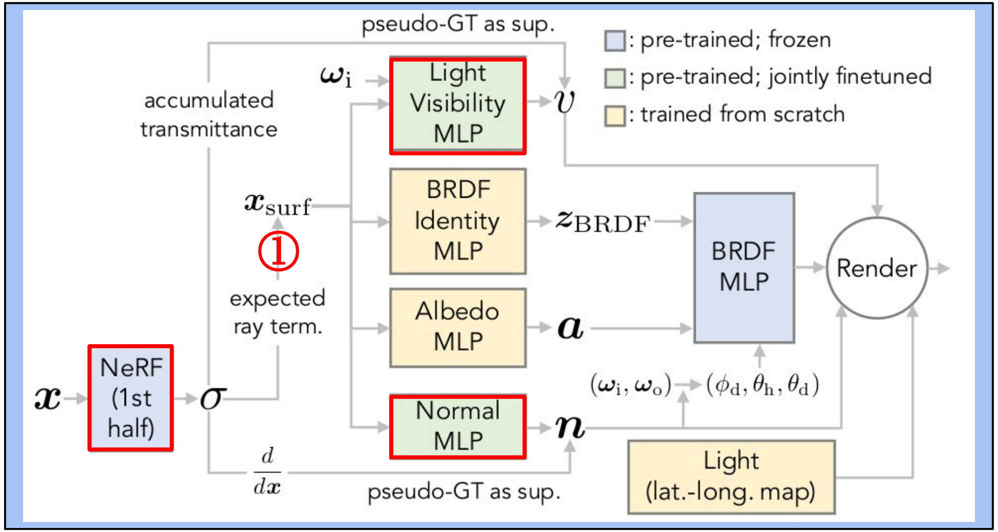
③ **Normal:**

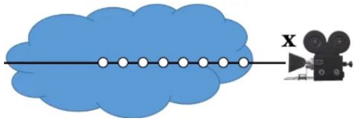
NeRFactor includes 4 blocks:

- **Shape**

  Train **vanilla NeRF**

3D location (x, y, z)
2D direction (θ, φ) ⇨ **NeRF** ⇨ Volume density σ
~~Emitted color (r, g, b)~~





① **Surface location:**

March through NeRF's σ-volume to camera

*Surface is more efficient than volume

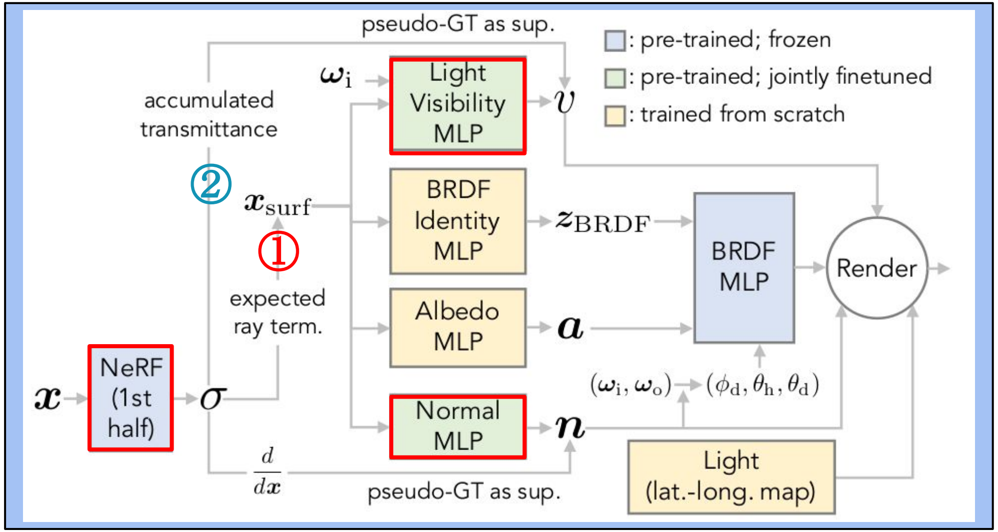
*as input later

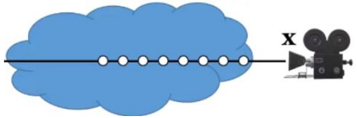
NeRFactor includes 4 blocks:

- **Shape**

  Train **vanilla NeRF**

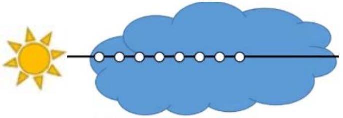
3D location (x, y, z)
2D direction (θ, φ) ⇒ NeRF ⇒ Volume density σ
~~Emitted color (r, g, b)~~





① **Surface location:**

March through NeRF's σ-volume to camera



② **Light visibility:**

March through NeRF's σ-volume to each light location

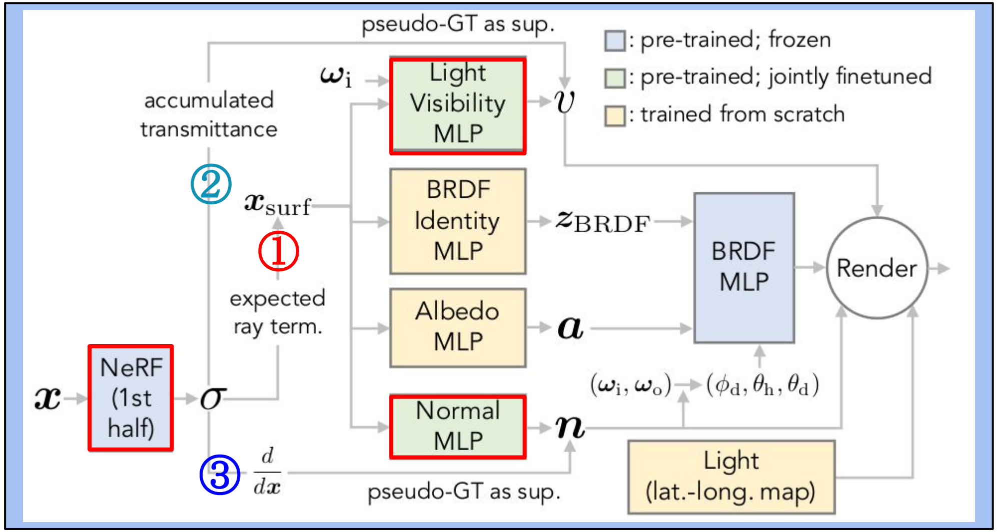
*as pseudo GT

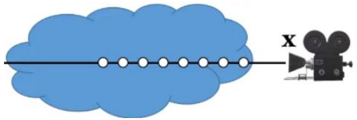
NeRFactor includes 4 blocks:

- **Shape**

  Train **vanilla NeRF**

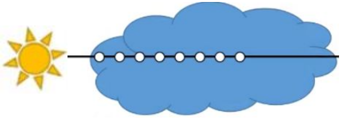
3D location (x, y, z)
2D direction (θ, φ) ⇨ NeRF ⇨ Volume density σ
~~Emitted color (r, g, b)~~





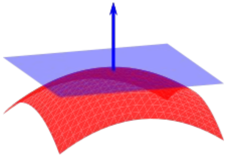
① **Surface location:**

March through NeRF's σ-volume to camera



② **Light visibility:**

March through NeRF's σ-volume to each light location



③ **Normal:**

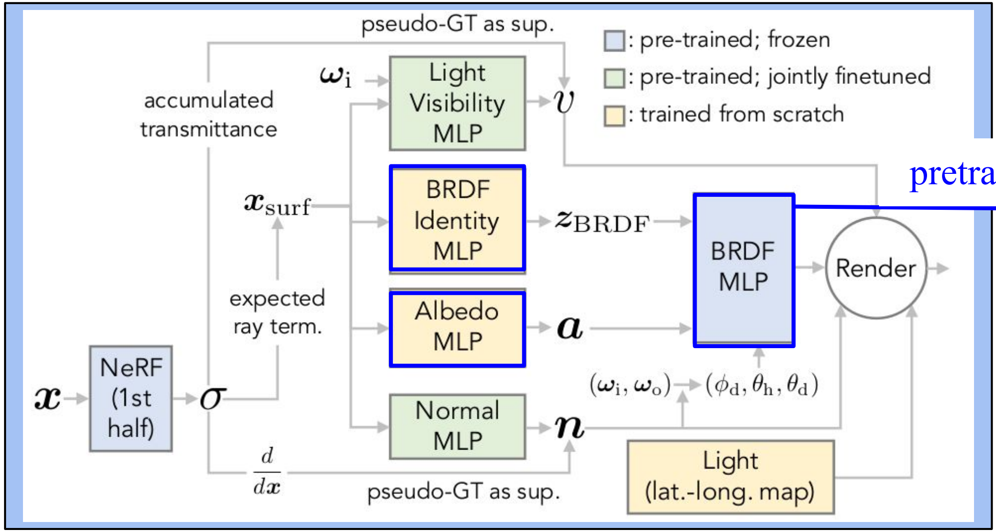
Calculate negative normalized gradient of NeRF's σ-volume

*as pseudo GT

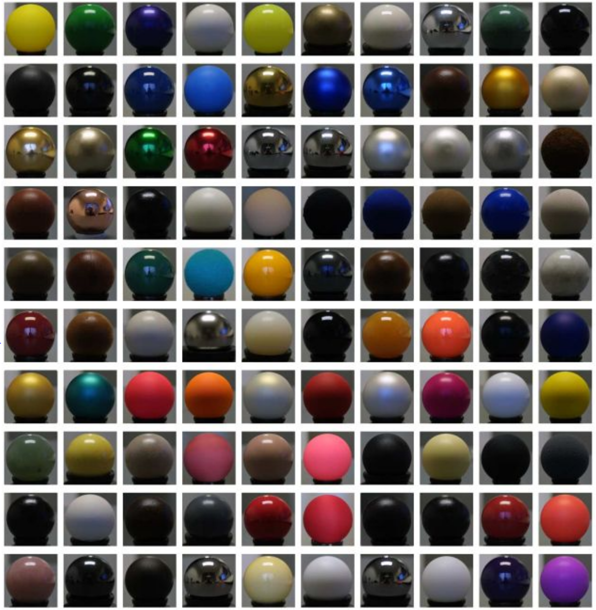
NeRFactor includes 4 blocks:

- **Shape**
- **Reflectance**

  Data-driven BRDF + Albedo



pretrained on



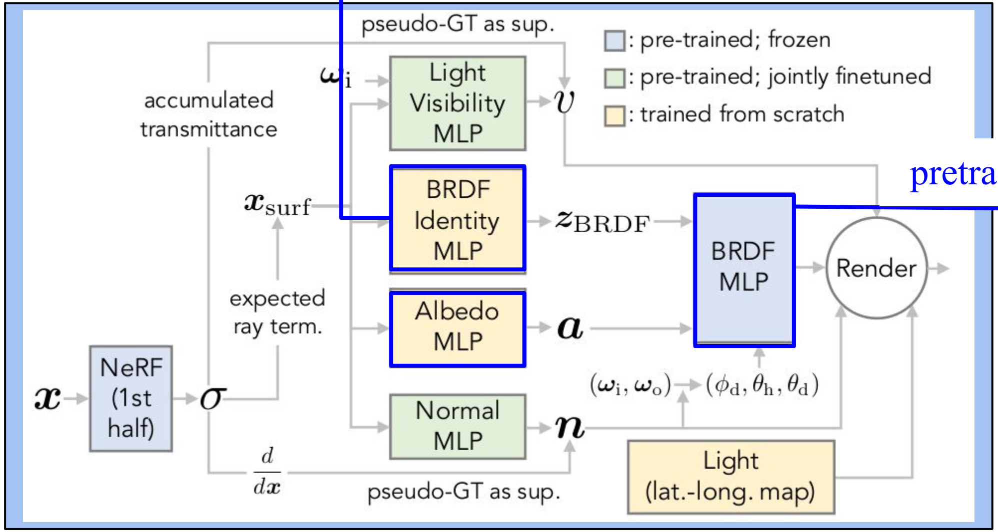
MERL dataset (real measured BRDFs)

NeRFactor includes 4 blocks:

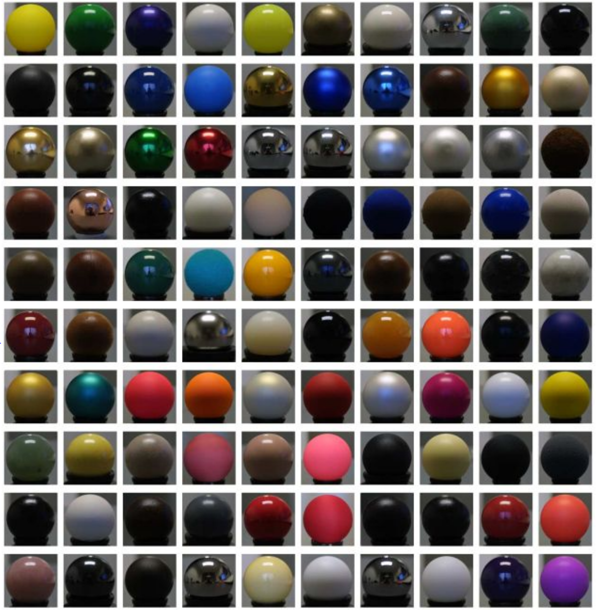
- **Shape**
- **Reflectance**

Data-driven BRDF + Albedo

Latent code: vector in an abstract M-D space



pretrained on



MERL dataset (real measured BRDFs)

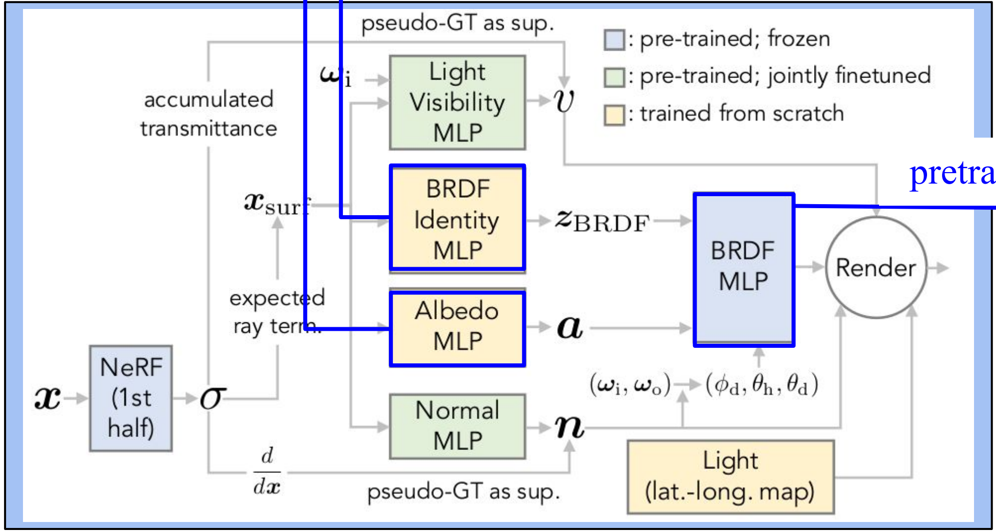
NeRFactor includes 4 blocks:

- **Shape**
- **Reflectance**

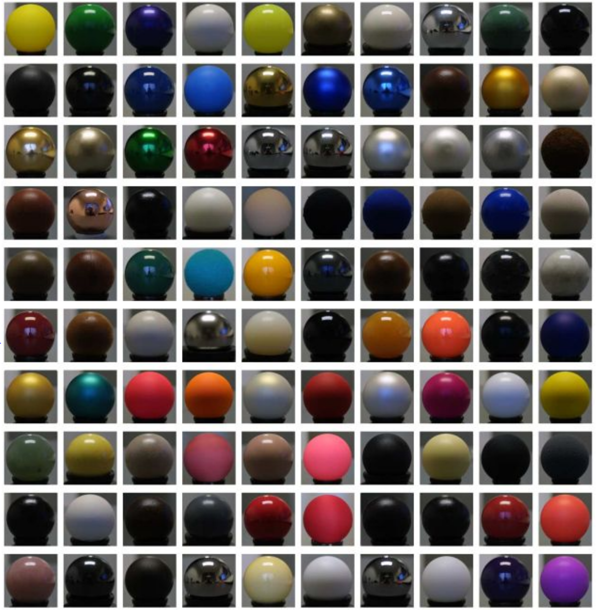
  Data-driven BRDF + Albedo

*Albedo: diffuse component of reflectance
*Model visibility: separating shadows from albedo

Latent code: vector in an abstract M-D space



pretrained on

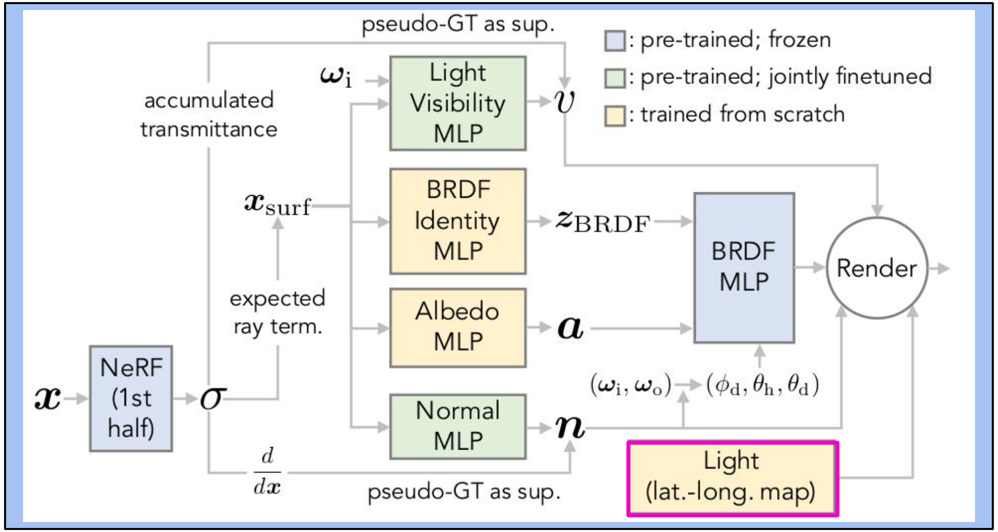


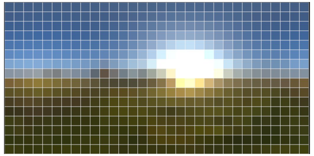
MERL dataset (real measured BRDFs)

NeRFactor includes 4 blocks:

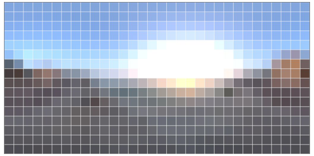
- **Shape**
- **Reflectance**
- **Light**

  Estimate an HDR **light probe image** with size of 16*32

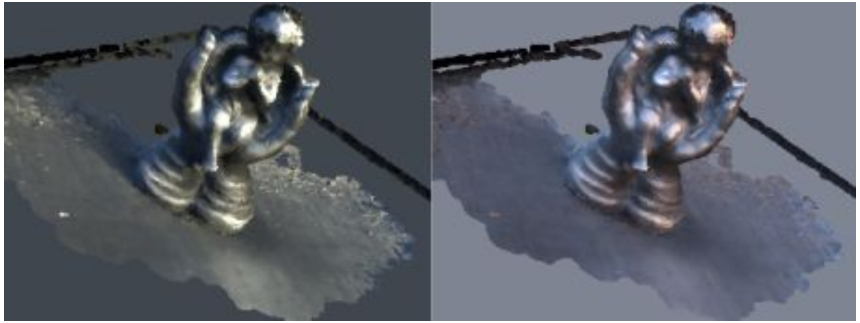




Sunrise



Sunset

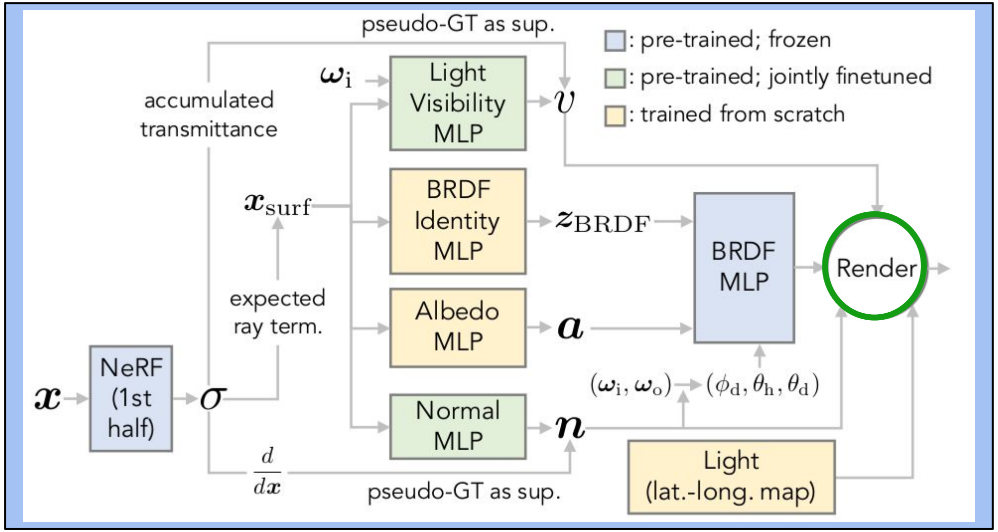


NeRFactor includes 4 blocks:

- **Shape**
- **Reflectance**
- **Light**
- **Render**

$$L_o(X, \hat{\omega}_o) = \int_{\mathbf{S}^2} L_i(X, \hat{\omega}_i) \, f_X(\hat{\omega}_i, \hat{\omega}_o) \, |\hat{\omega}_i \cdot \hat{n}| \, d\hat{\omega}_i$$

Outgoing light | Incoming light | **Reflectance** | **Normal**



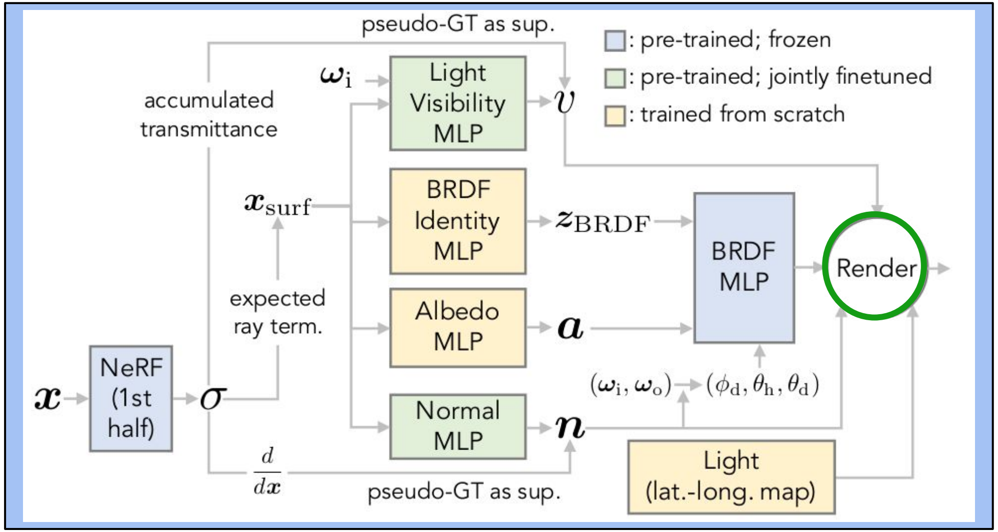
Supervisions:

(1) **Rendering GT**: pixel RGB values

(2) **BRDF GT**: real measured BRDFs in MERL

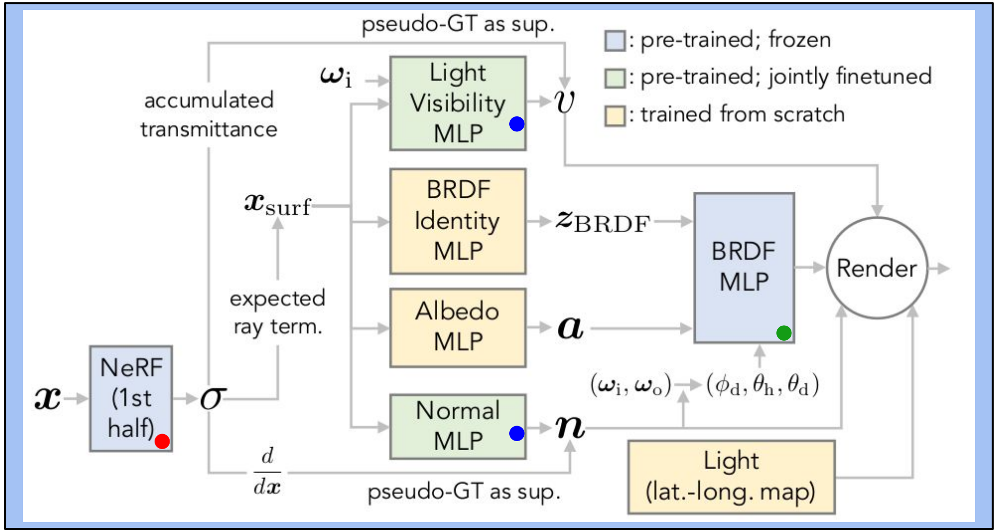
(3) **pseudo GT**: calculated from shape estimation

(4) **Smooth term:** smoothness is encouraged across spatial locations



Supervisions:

(1) **Rendering GT**: pixel RGB values
(2) **BRDF GT**: real measured BRDFs in MERL
(3) **pseudo GT**: calculated from shape estimation
(4) **Smooth term:** smoothness is encouraged across spatial locations



(1) Pre-train some MLPs with (1) (2) (3)

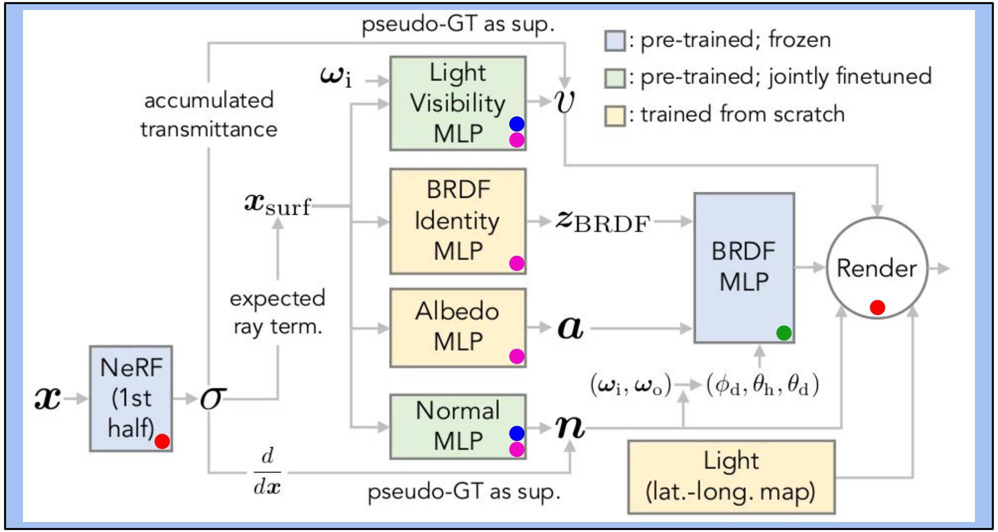
Supervisions:

(1) **Rendering GT**: pixel RGB values

(2) **BRDF GT**: real measured BRDFs in MERL

(3) **pseudo GT**: calculated from shape estimation

(4) **Smooth term:** smoothness is encouraged across spatial locations



(1) Pre-train some MLPs with (1) (2) (3)

(2) Jointly train some MLPs with (1) (4)

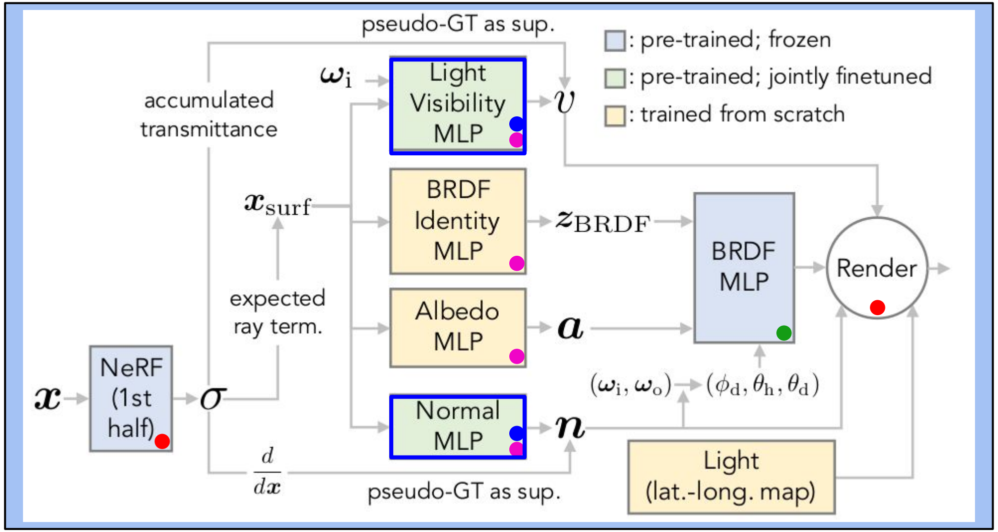
Supervisions:

(1) **Rendering GT**: pixel RGB values

(2) **BRDF GT**: real measured BRDFs in MERL

(3) **pseudo GT**: calculated from shape estimation

(4) **Smooth term:** smoothness is encouraged across spatial locations

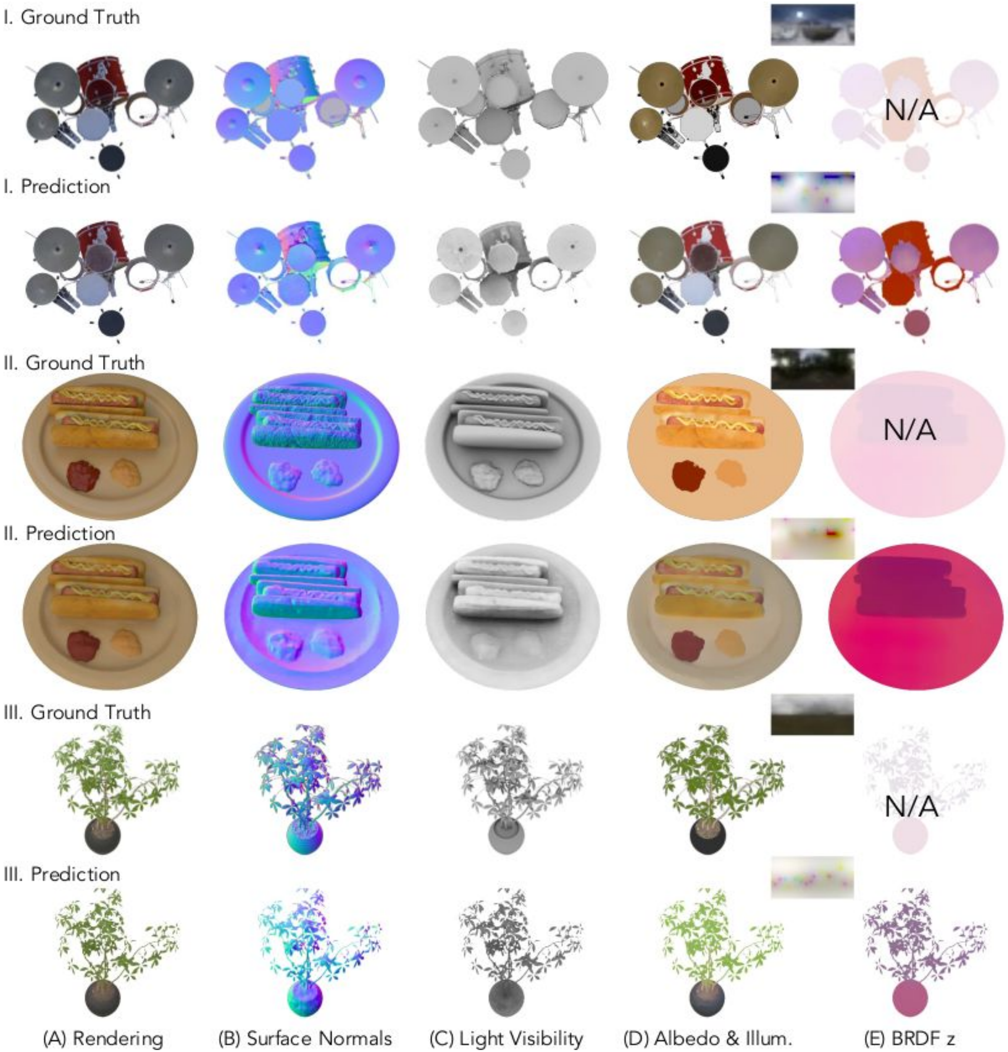


(1) Pre-train some MLPs with (1) (2) (3)

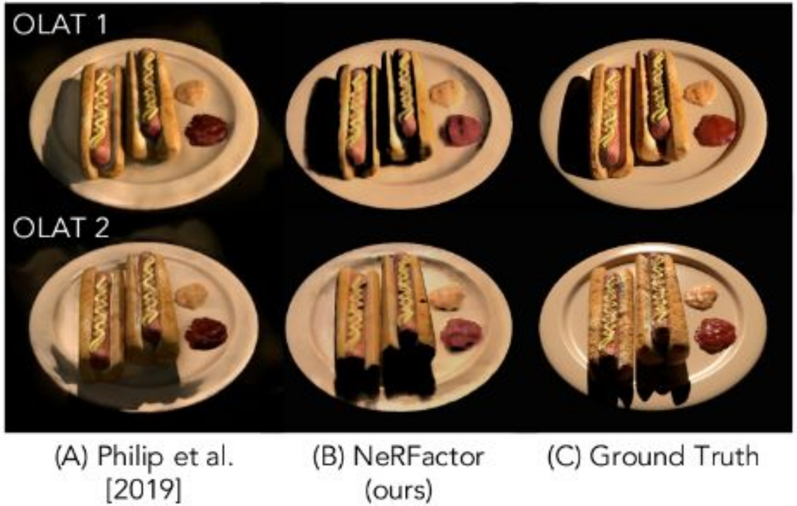
(2) Jointly train some MLPs with (1) (4)

Why pre-train first and jointly train later ?

To prevent the albedo or BRDF MLP from mistakenly attempting to explain away shadows.



Comparisons in point light relighting:



*poor geometry reconstruction is revealed in harsh lighting conditions.

(B) (C): NeRFactor predicts high-quality and smooth results.

(D): Albedo is recovered cleanly without shadow.

The predicted light probes correctly reflect the locations of the primary light sources.

(E) The predicted BRDFs correctly reflect different materials.

Thank you !