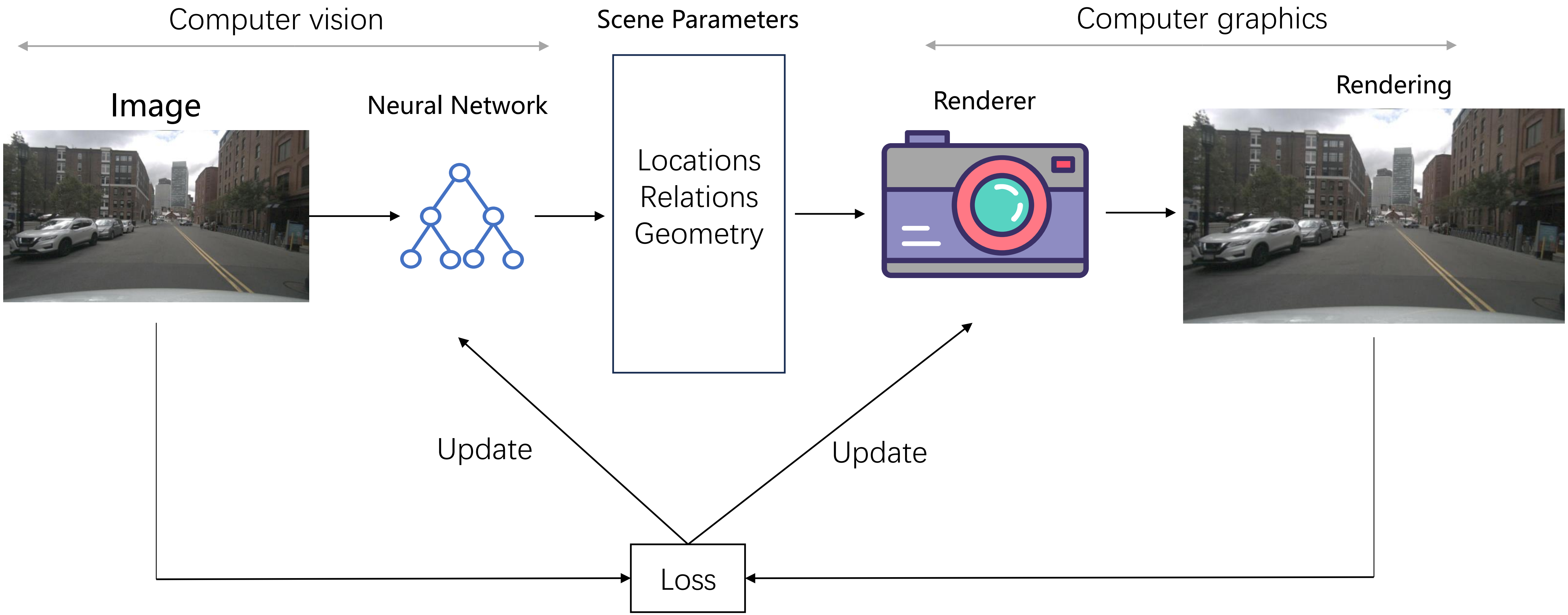
Computer graphics with computer vision



Computer vision

Scene Parameters

Computer graphics

Image

Neural Network

Locations
Relations
Geometry

Renderer

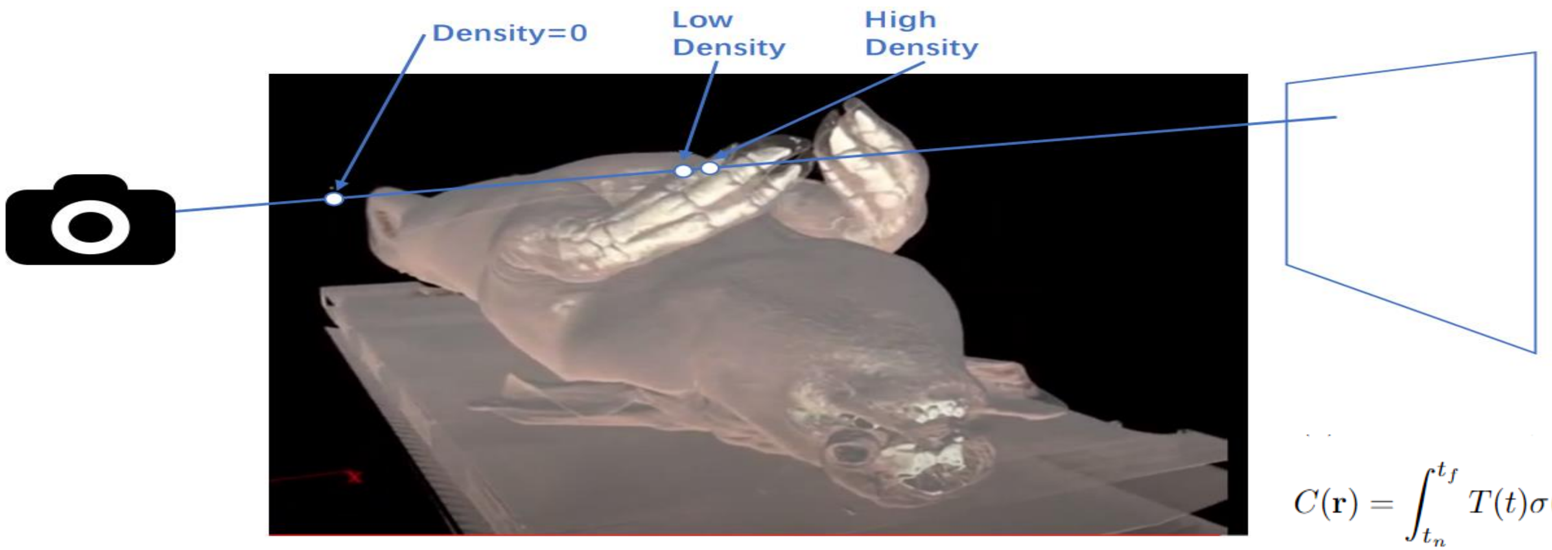
Rendering

Update

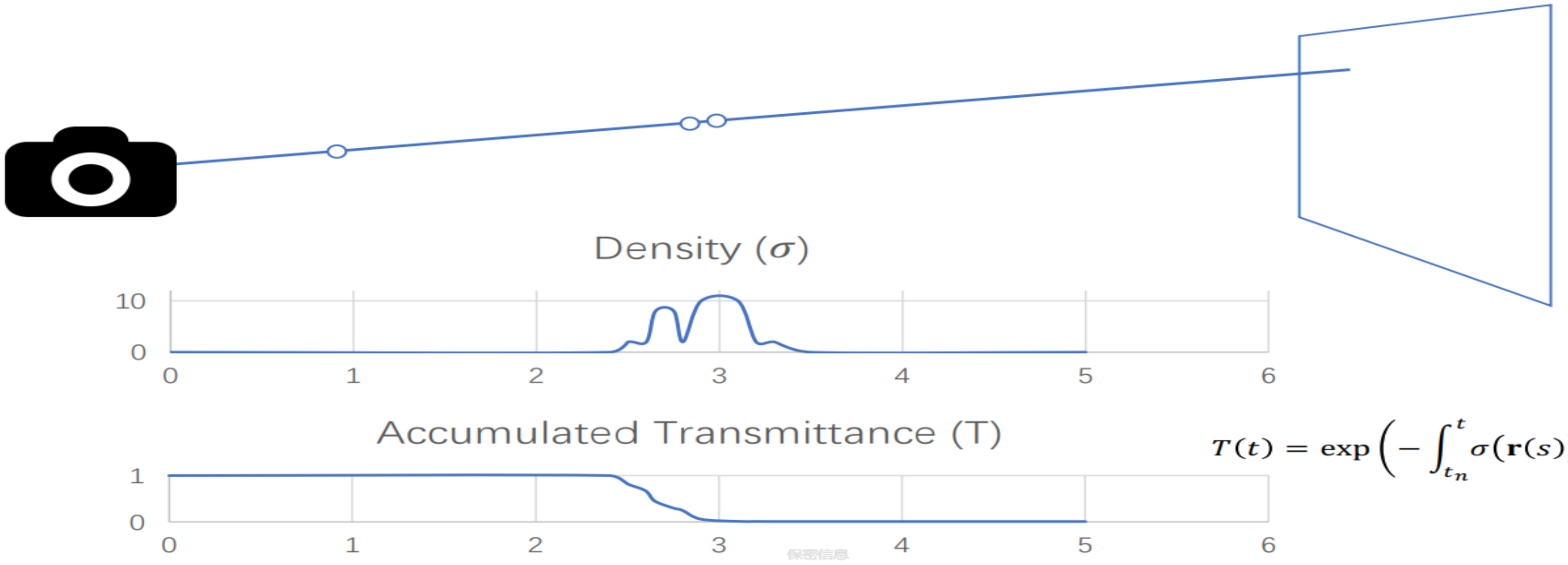
Update

Loss

Volume Rendering

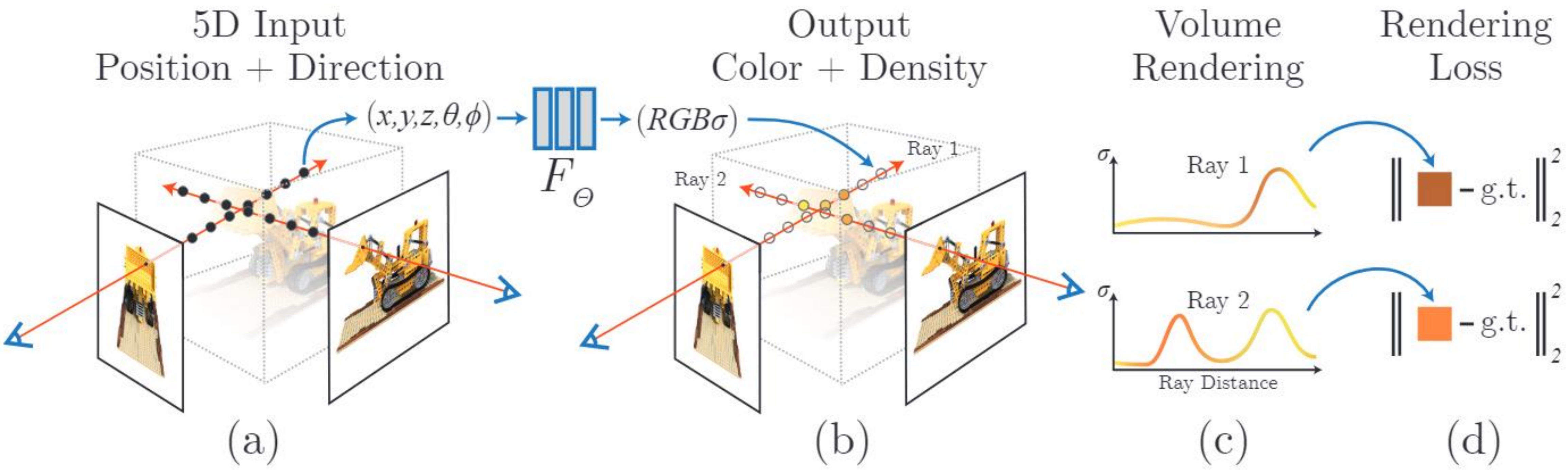


$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right).$$



$$\boldsymbol{T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right)}$$

NeRF

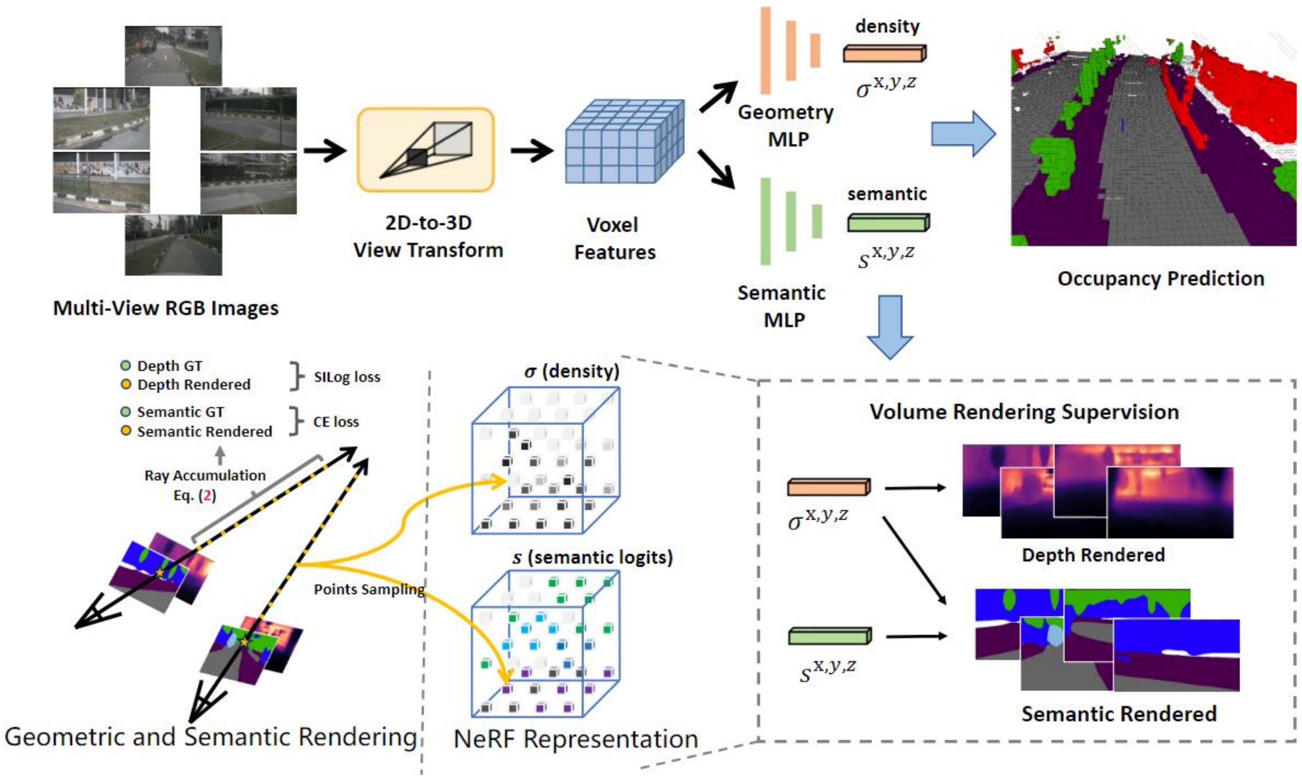


$$f_\theta: (x,d) \rightarrow (c,\sigma)$$

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

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

UniOcc



NeRF $f_\theta: (x, d) \rightarrow (c, \sigma)$

UniOcc $f_\theta: (x) \rightarrow (s, \sigma)$

Advantage

- Cost lower memory
- Higher resolution
- Easily learned

$$T(z_k) = exp(-\sum_{t=1}^{k-1} \sigma(z_t)\beta_t), \tag{3}$$

$$\alpha(z_k) = 1 - exp(-\sigma(z_k)\beta_k), \tag{4}$$

$$D = \sum_{k=1}^{N} T(z_k)\alpha(z_k)z_k, \tag{5}$$

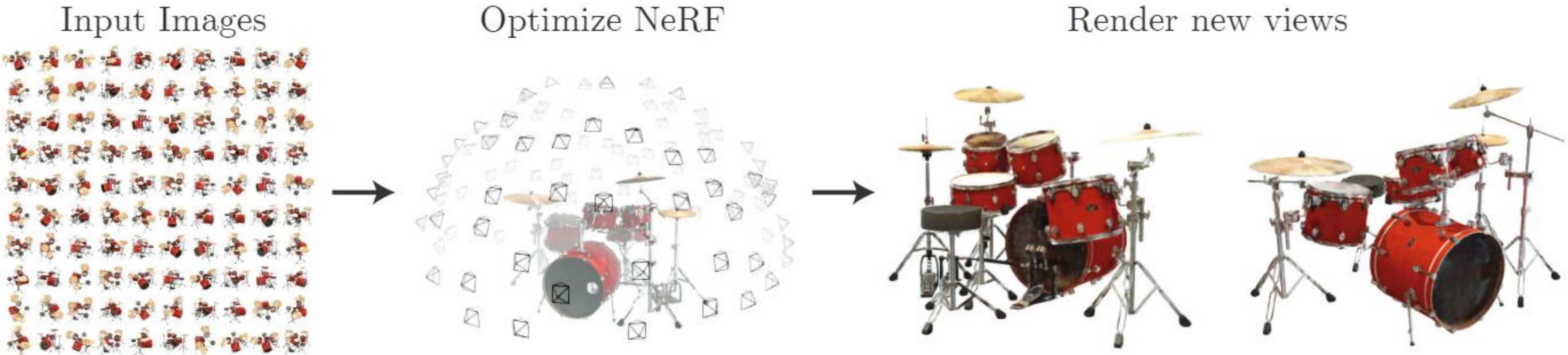
$$S = \sum_{k=1}^{N} T(z_k)\alpha(z_k)s(z_k), \tag{6}$$

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \overset{\overset{\alpha}{\uparrow}}{(1 - \exp(-\sigma_i \delta_i))} \mathbf{c}_i \, , \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right),$$

Thoughts

- How to Render
- S=

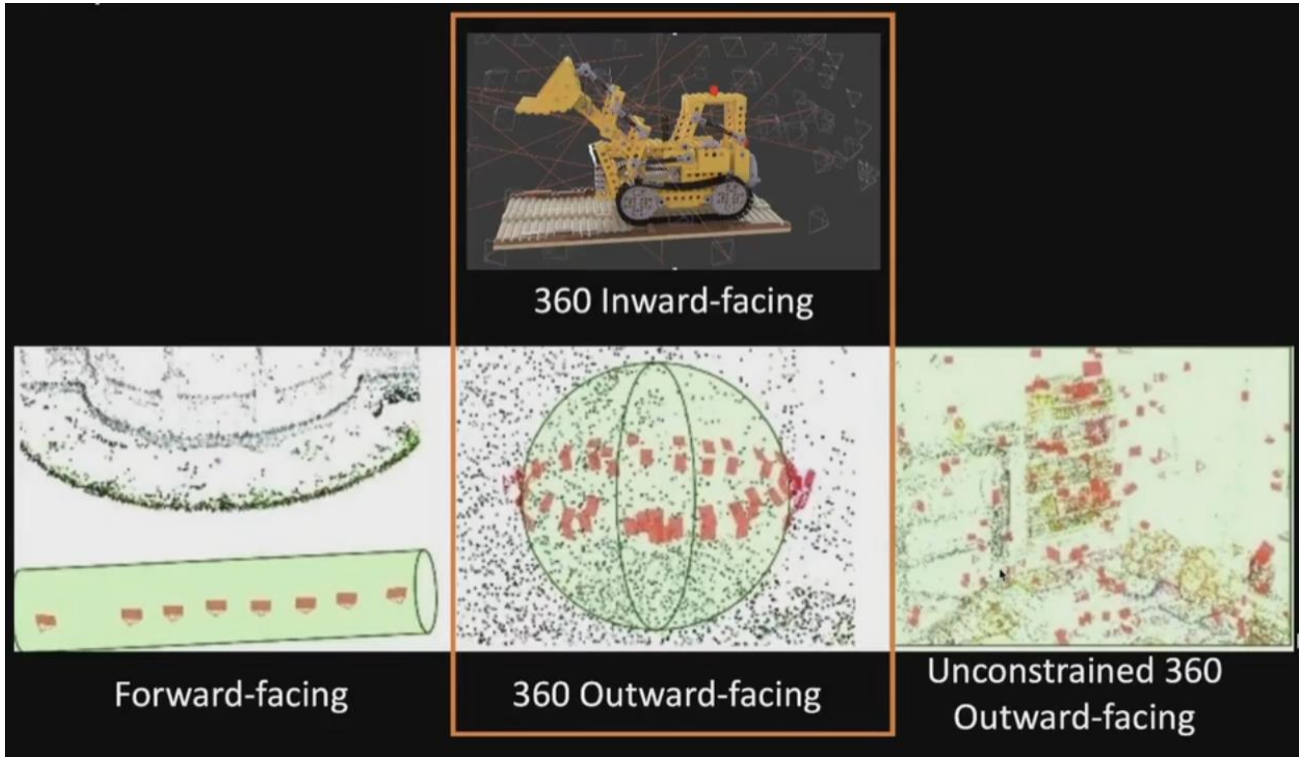
Challenge——Generalization

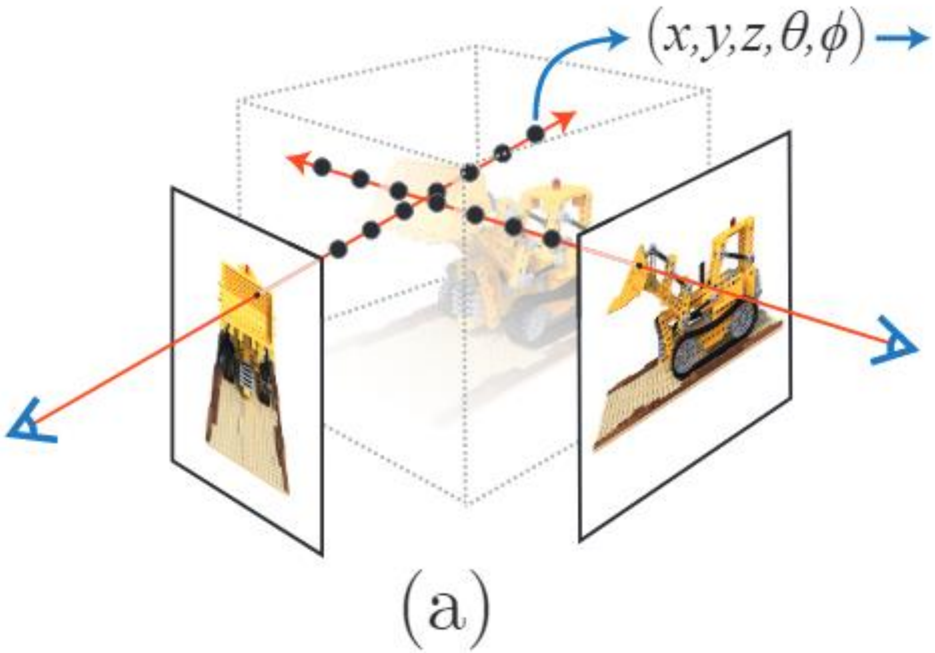


Challenge——Efficiency

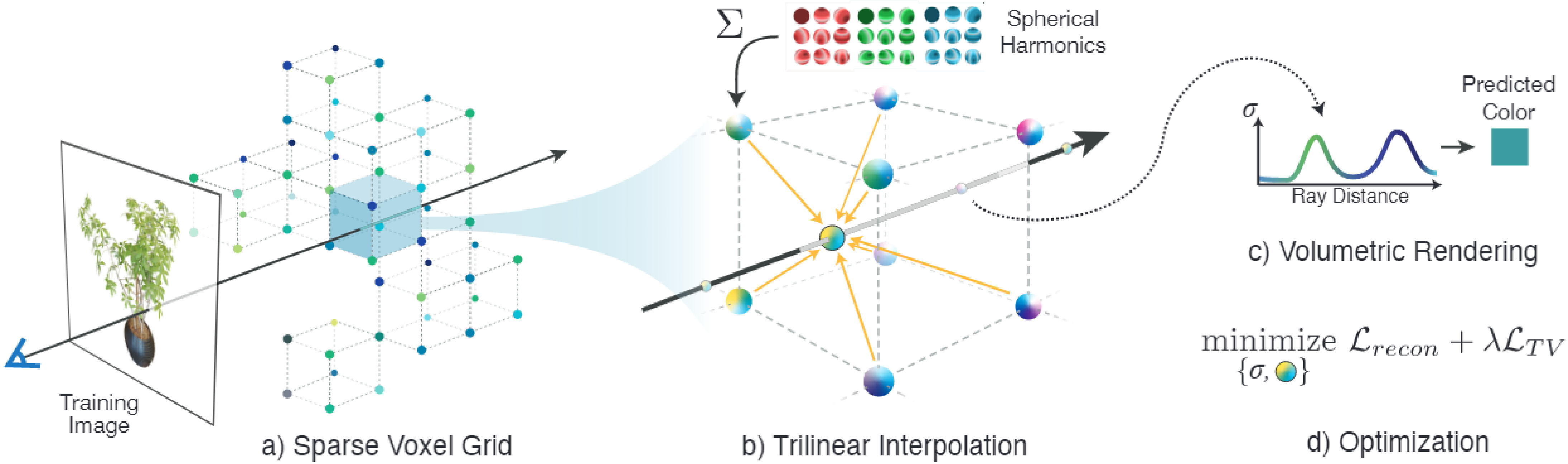
数据集	数据量	Time
Fern（真实forward场景）	~100图片	4hours
Lego（虚拟前景场景）	~100图片	3hours

Challenge——Ambiguity

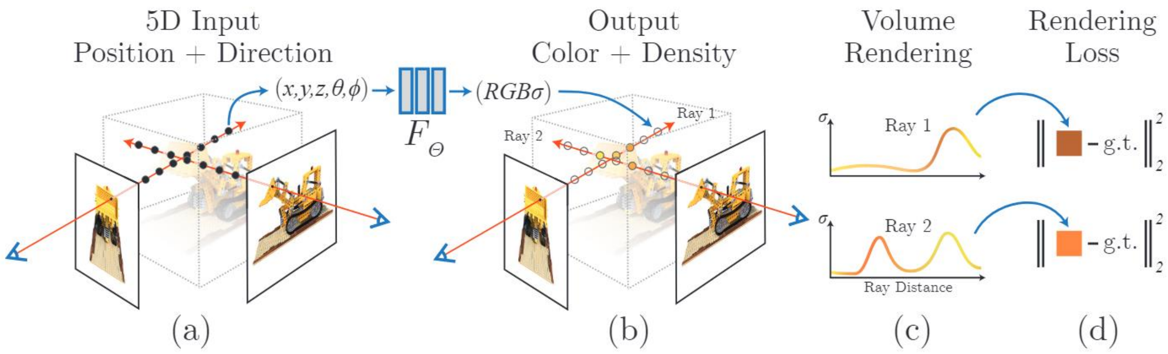




Plenoxels

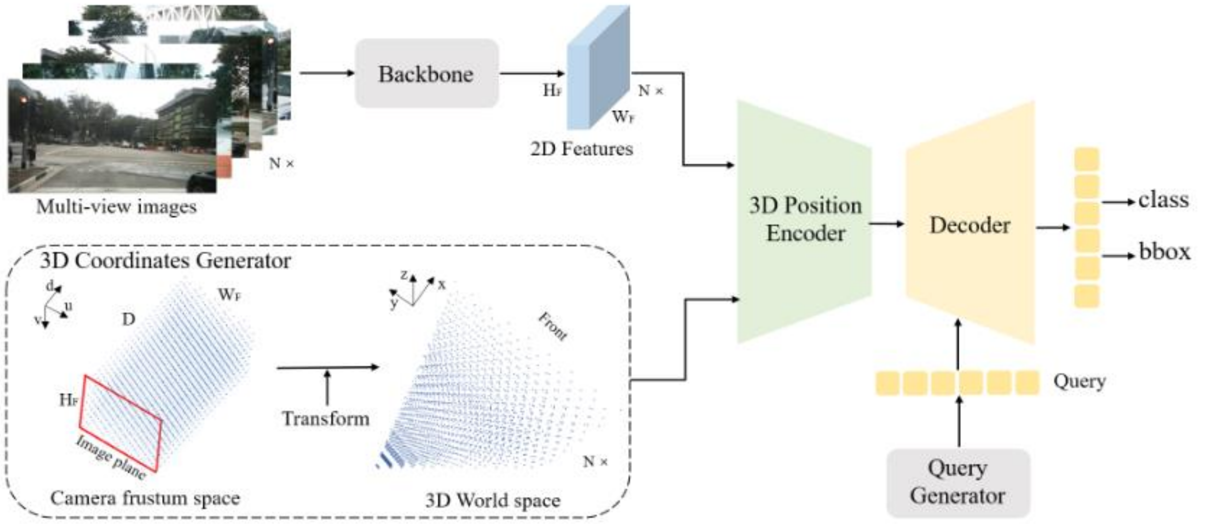

Training Image
a) Sparse Voxel Grid
Σ
Spherical Harmonics
b) Trilinear Interpolation
σ
Ray Distance
Predicted Color
c) Volumetric Rendering
$\underset{\{\sigma, ●\}}{\text{minimize}}\ \mathcal{L}_{recon} + \lambda \mathcal{L}_{TV}$
d) Optimization

PETR



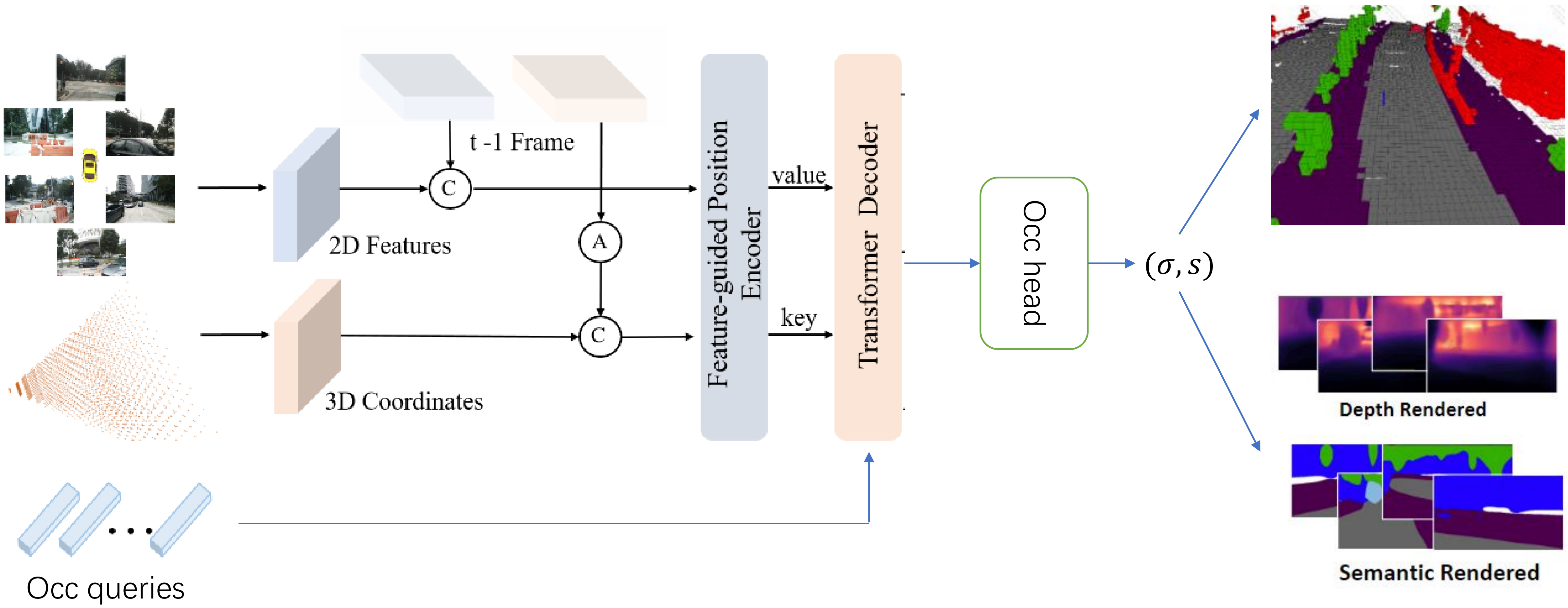
$$f_\theta: (x, d) \rightarrow (c, \sigma) \longrightarrow f_\theta: (\gamma(x), d) \rightarrow (c, \sigma)$$

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right).$$

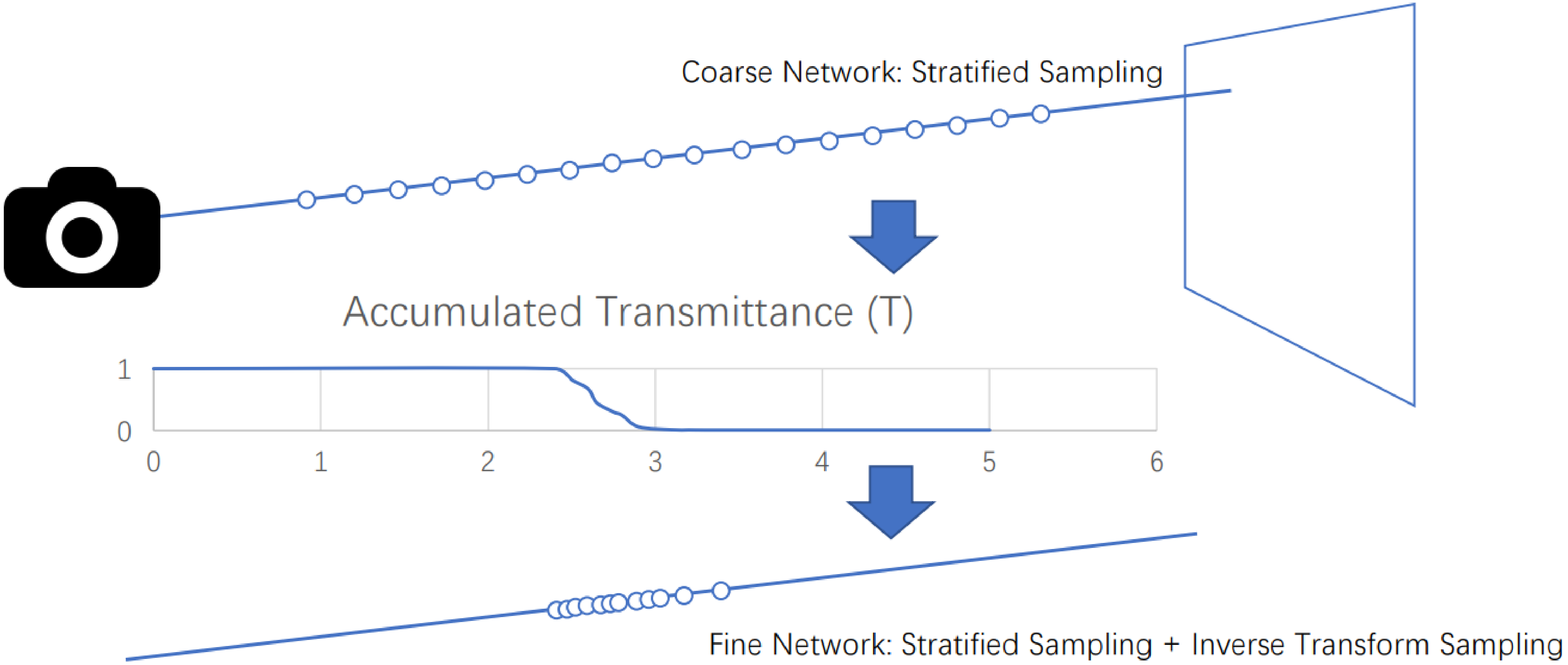


PETR

pipeline


t -1 Frame
C
2D Features
A
C
3D Coordinates
Feature-guided Position Encoder
value
key
Transformer Decoder
Occ head
(σ, s)
Depth Rendered
Semantic Rendered
Occ queries

Pipeline++ ->Coarse to fine

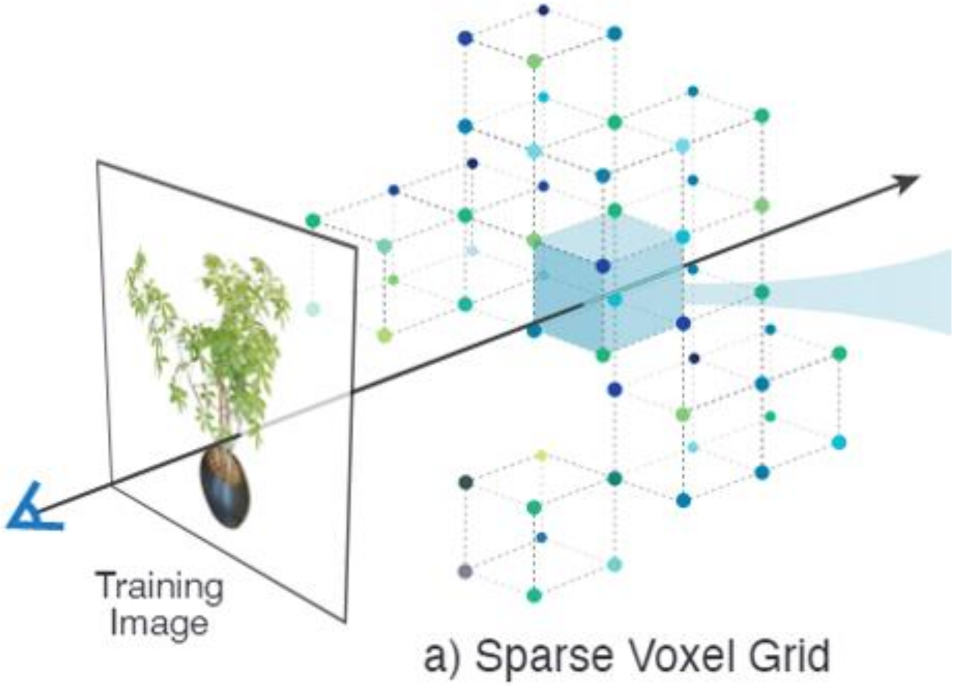


Coarse Network: Stratified Sampling

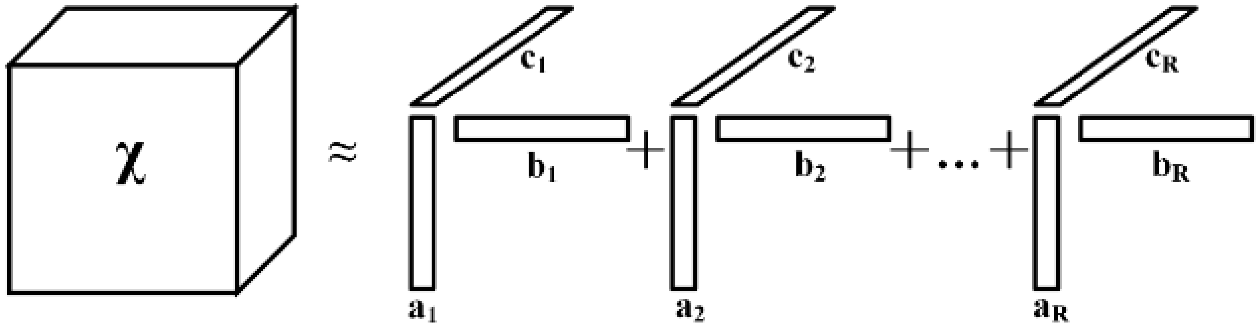
Accumulated Transmittance (T)

Fine Network: Stratified Sampling + Inverse Transform Sampling

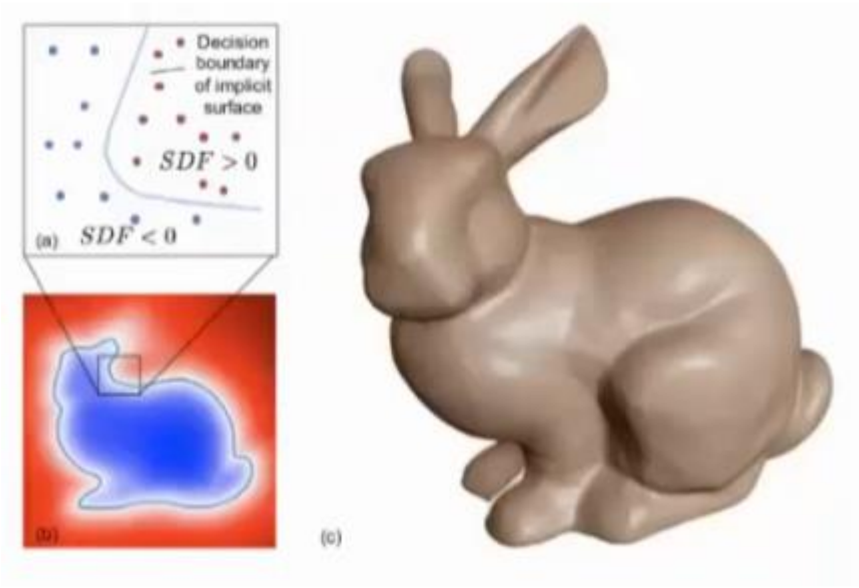
Pipeline++ ->Memory Cost



Tensor decomposition



Pipeline++ ->SDF、TSDF


Decision boundary of implicit surface
SDF > 0
(a) SDF < 0
(b)
(c)