Deep Image Prior
&
Neural Fields

Representing images by neural networks

Supervised Learning

- Empirical distribution of training data $(x, y) \sim \hat{p}(x, y)$

$$(\mathbb{X}, \mathbb{Y}) = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$$

$$\begin{aligned} \boldsymbol{\psi}_{ML} &= \arg\max_{\boldsymbol{\psi}} q(\mathbb{Y}|\mathbb{X}, \boldsymbol{\psi}) \\ &= \arg\max_{\boldsymbol{\psi}} \sum_{i=1}^{m} \log q(y^{(i)}|x^{(i)}, \boldsymbol{\psi}) \end{aligned}$$

$$\boldsymbol{\psi}_{ML} = \arg\max_{\boldsymbol{\psi}} \mathbb{E}_{(x,y)\sim\hat{p}(x,y)} \log q(y|x, \boldsymbol{\psi})$$

Supervised Learning

- Training data: $(x, y) \sim \hat{p}(x, y)$
- Choose distribution $q(y|x; \psi)$

- NN represents a function $f_\theta(x)$, which is not a direct prediction of y, but of ψ (e.g. mean)

- ML principle give us the cost function as

$$-\log q(y|x; f_\theta(x))$$

Example – Gaussian Model

- Training data distribution: $p(x, y) = p(y|x)p(x)$
- Estimator of mean y: f(x)
- Model distribution: $q(y|x) \equiv N(y|f(x), \sigma^2)$

$$\hat{f}(x) = \arg\min_{f(x)} \mathbb{E}_{p(x,y)}[-\log q(y|f(x))]$$

$$L = \int \left[\int (y - f(x))^2 p(y|x) dy \right] p(x) dx + c$$

$$\frac{\partial L}{\partial f} \propto \int (y - f(x)) p(y|x) dy = 0$$

$$\hat{f}(x) = \int y p(y|x) dy = \mathbb{E}_{p(y|x)}[y]$$

Example – Laplace Model

- Training data distribution: $p(x, y) = p(y|x)p(x)$
- Estimator of mean y: f(x)
- Model distribution: $q(y|x) \equiv \text{Laplace}(y|f(x), \gamma)$

$$\hat{f}(x) = \arg\min_{f(x)} \mathbb{E}_{p(x,y)}[-\log q(y|f(x))]$$

$$L = \int \left[\int |y - f(x)|\, p(y|x)dy \right] p(x)dx + c$$

$$\frac{\partial L}{\partial f} \propto \int \text{sign}(y - f(x))p(y|x)dy = 0$$

$\hat{f}(x) = m$ … median: $P(y \leq m|x) = P(y \geq m|x)$

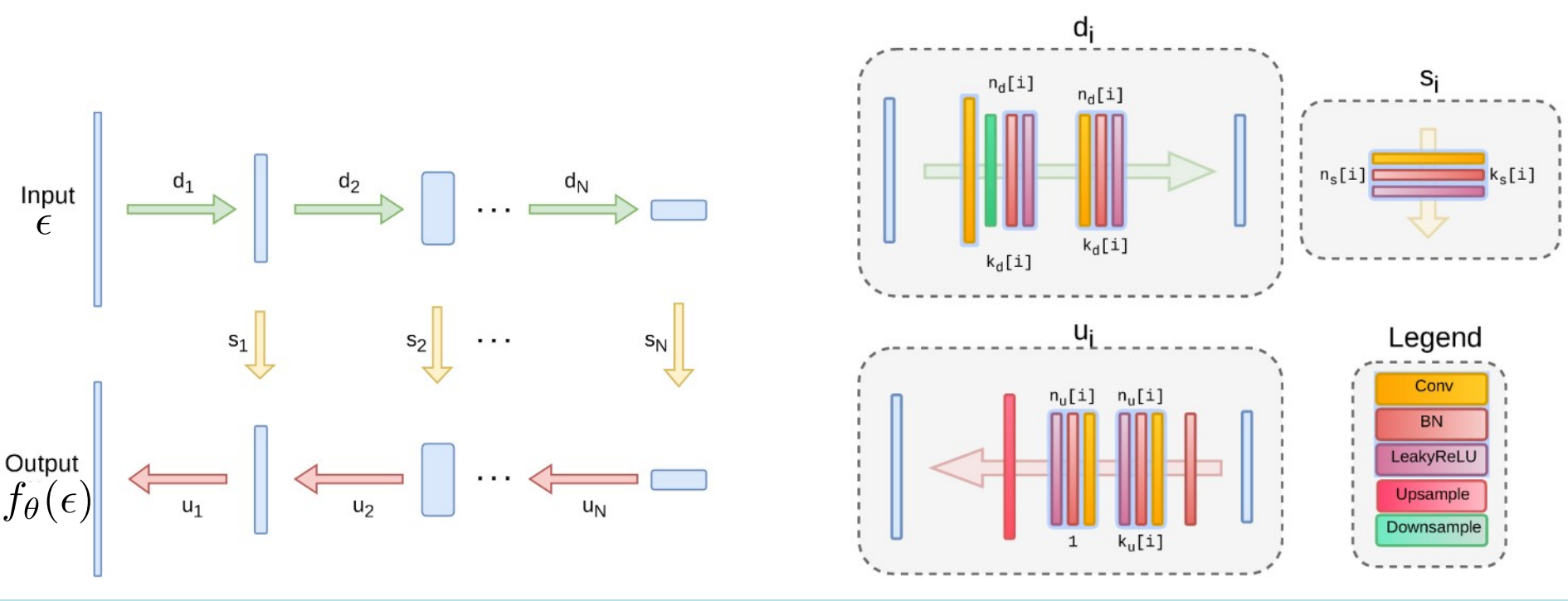
Deep Image Prior

- Denoising $z = u + n$ $\min_u F(u; z)$
- no prior $F(u; z) = \|u - z\|^2$
- with prior $F(u; z) = \|u - z\|^2 + R(u)$
- with Deep Image Prior $\min_\theta F(f_\theta(\epsilon); z)$

$$F(f_\theta(\epsilon); z) = \|f_\theta(\epsilon) - z\|^2$$

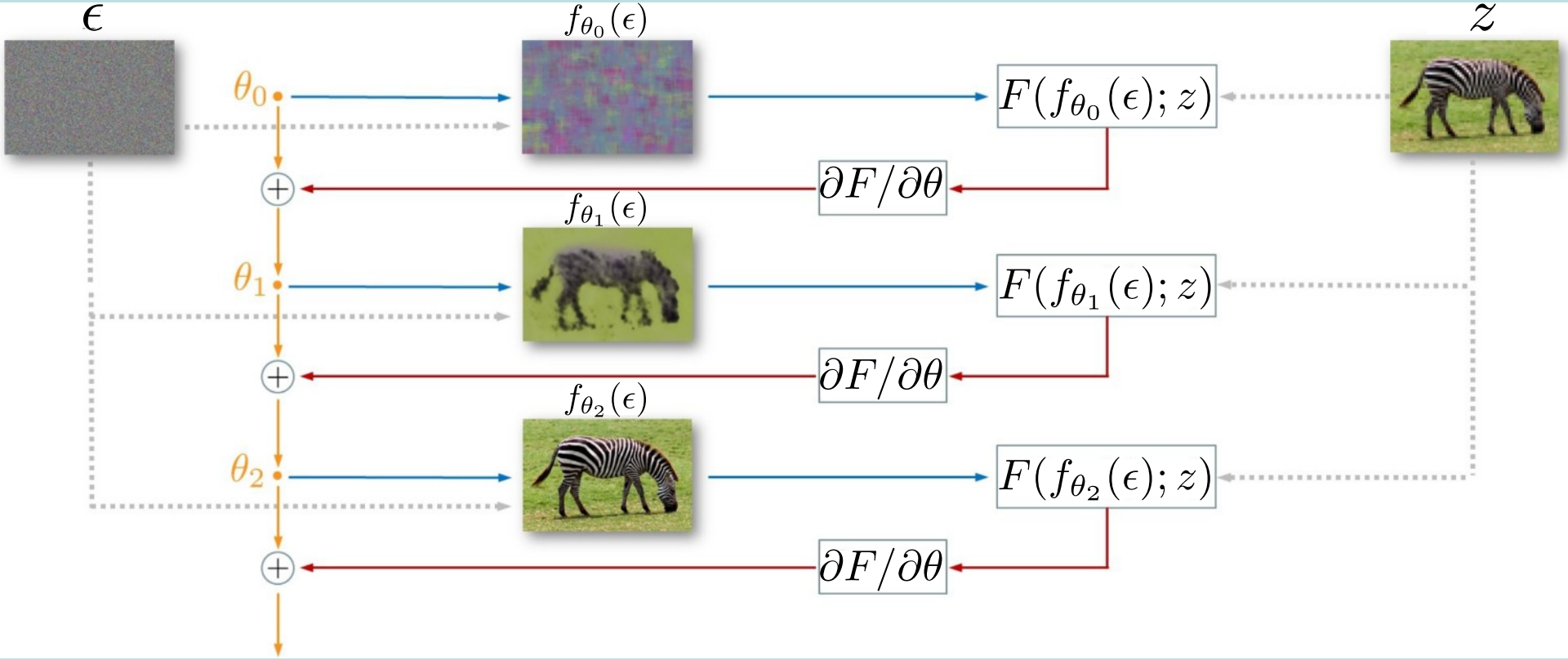
Deep Image Prior: arxiv.org/abs/1711.10925

- $f_\theta(\epsilon)$ is U-Net architecture

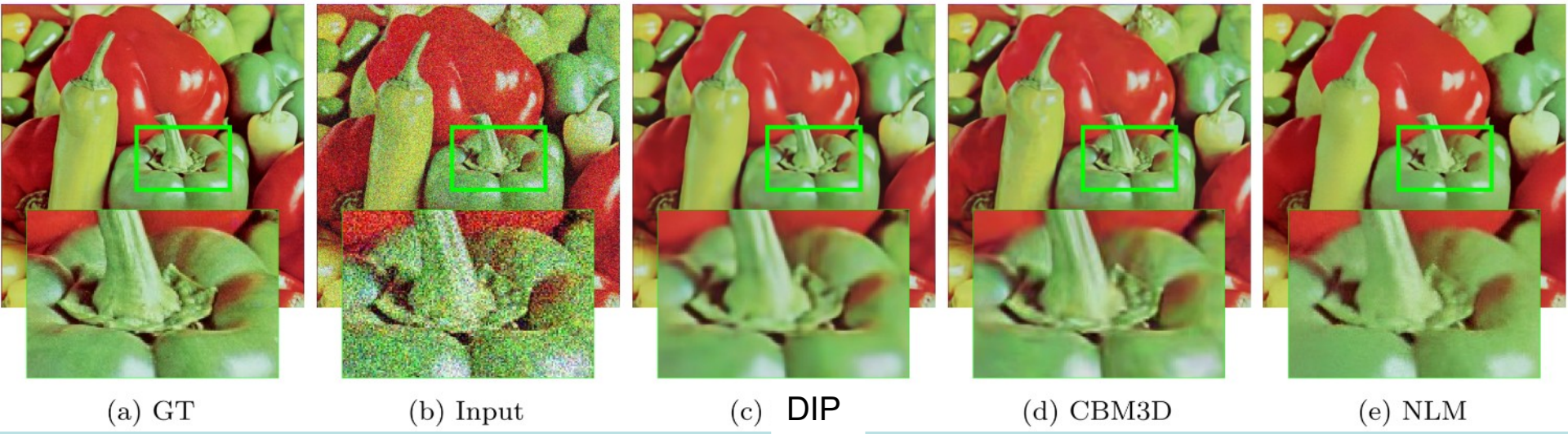


Zero-shot learning

- Gradient descent



Denoising example



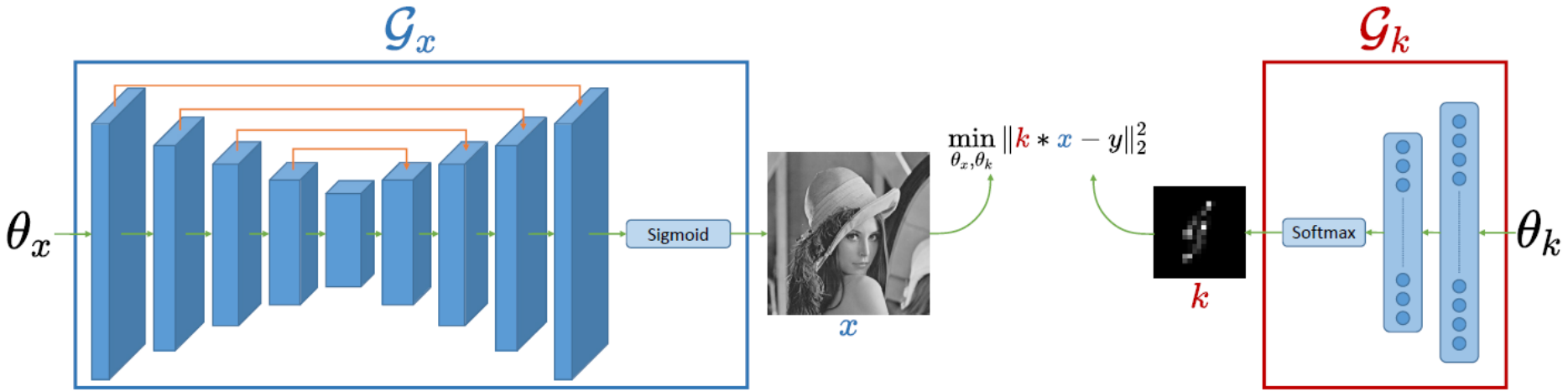
(a) GT (b) Input (c) DIP (d) CBM3D (e) NLM

Super-resolution example

(a) HR image (b) Bicubic upsampling (c) No prior (d) TV prior (e) Deep image prior

Blind Deconvolution

$$\min_{\theta_x,\theta_k} \|k * x - y\|_2^2, \quad \text{s.t. } 0 \le x \le 1 \text{ and } 0 \le k, \ \|k\|_1 = 1, \quad \text{where } x = \mathcal{G}_x(\theta_x), \ k = \mathcal{G}_k(\theta_k)$$



U-Net like encoder-decoder convolutional net with skip connections and final sigmoid

Fully-connected net with final soft-max

Implicit Neural Representation

- also called Neural Fields
- represent an image(object) function with MLP

$$f(x) : \mathbb{R}^m \rightarrow \mathbb{R}^n$$

  - e.g. color image $f(x) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$

Zero-shot learning

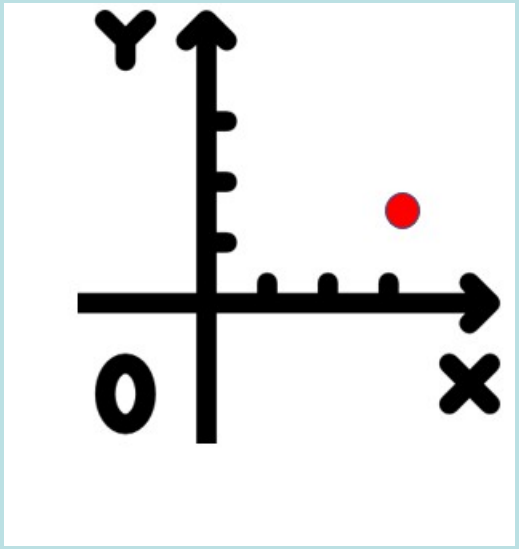
- e.g. fitting image u(x):

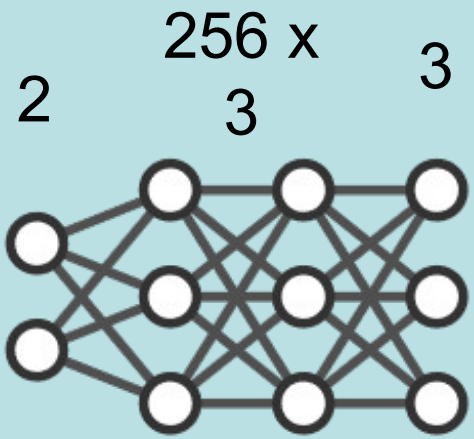
$$\min_\theta \sum_i \| f_\theta(x_i) - u(x_i) \|^2$$

- or more complex losses

Neural Fields

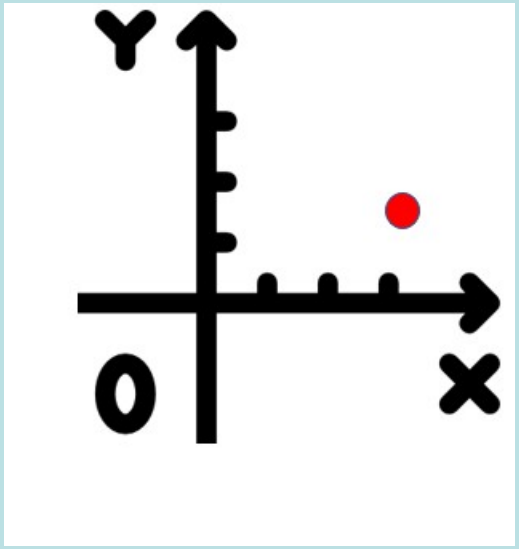
- Pros:
  - Compressed representation
  - Analytic derivatives $\nabla f(x)$
    using back-propagation
- Cons:
  - train for every case

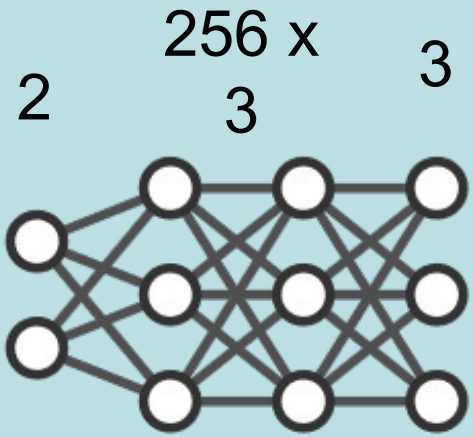

Y
0
X

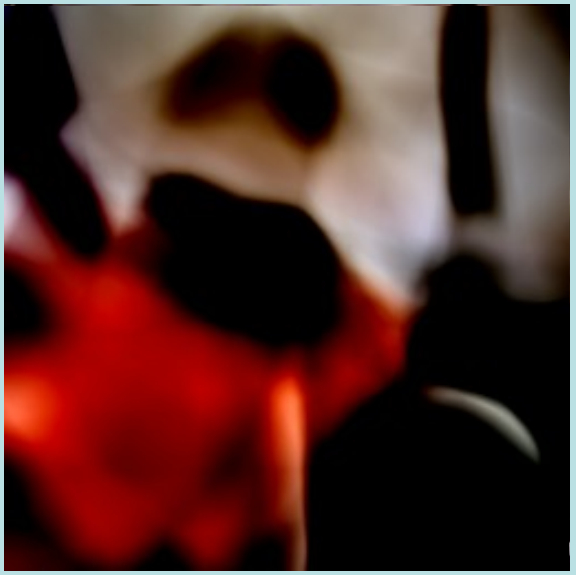

2
256 x
3
3


?

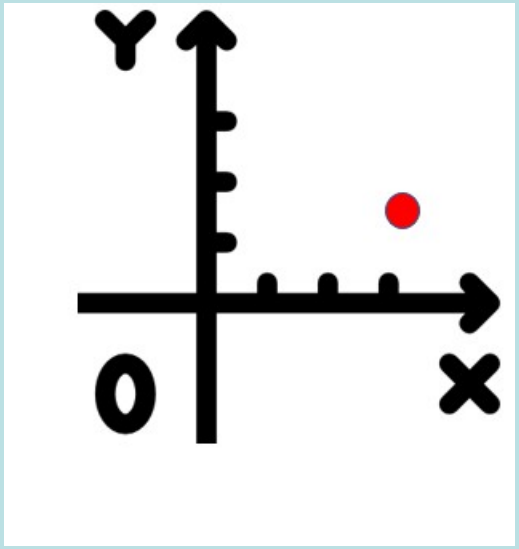
- ReLU

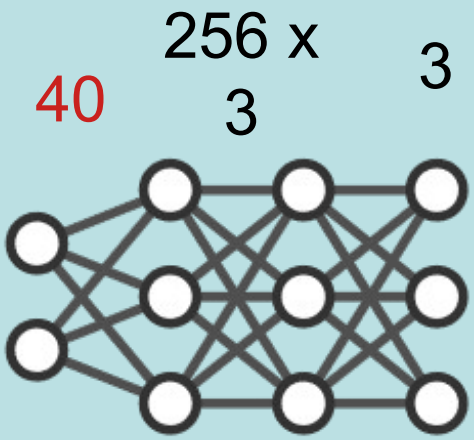






- Positional encoding





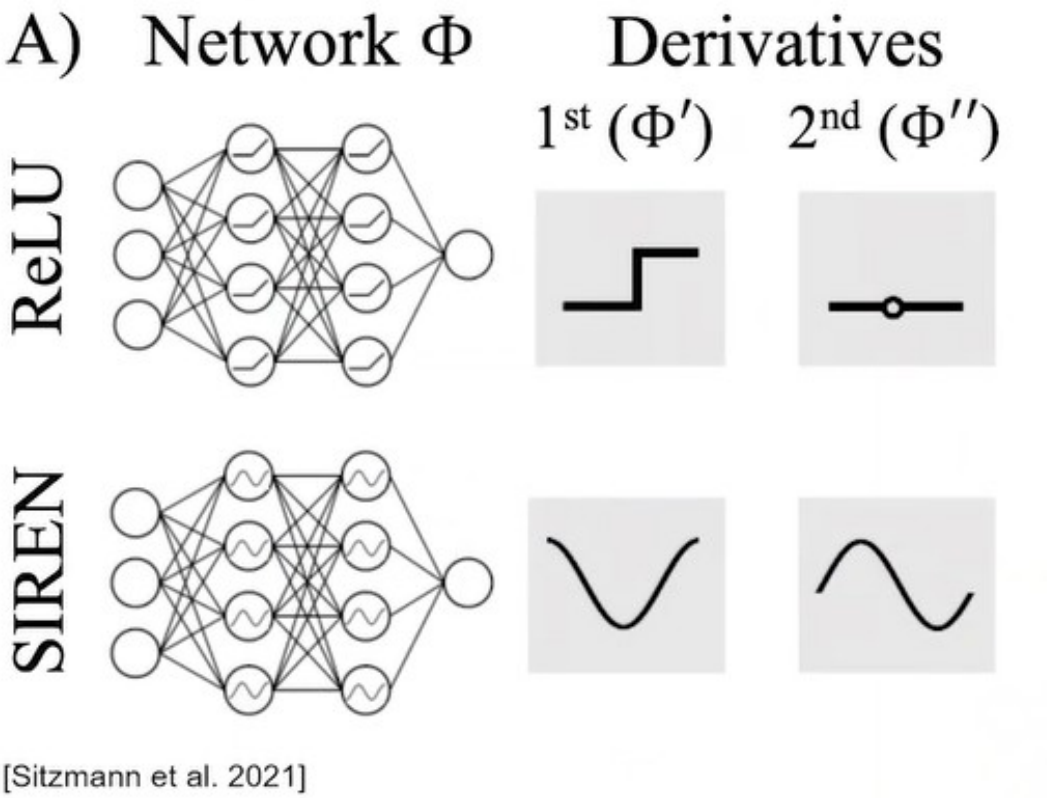
$$\gamma(\mathbf{x}) = [\gamma_1(\mathbf{x}), \gamma_2(\mathbf{x}), ..., \gamma_m(\mathbf{x})]$$

$$\gamma_{(2i)}(x) = sin(2^{i-1}\pi x),$$

$$\gamma_{(2i+1)}(x) = cos(2^{i-1}\pi x)$$

Tancik et al., 2020

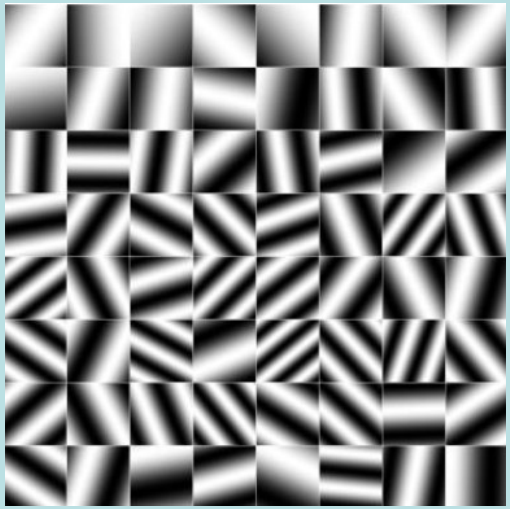
SIREN



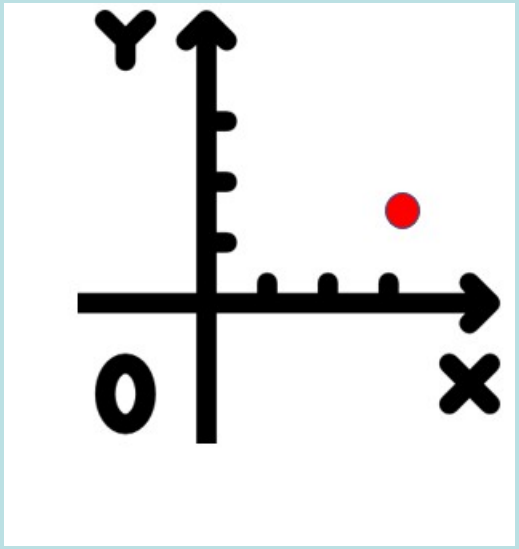
[Sitzmann et al. 2021]

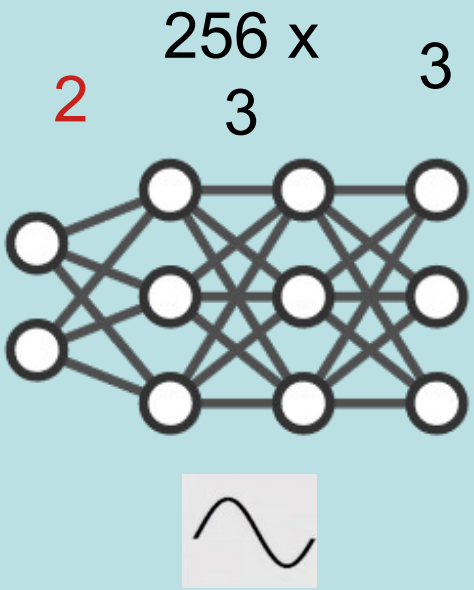
$$\mathrm{ReLU}(\mathbf{W}\mathbf{x} + \mathbf{b})$$

$$\sin(\mathbf{W}\mathbf{x} + \mathbf{b})$$

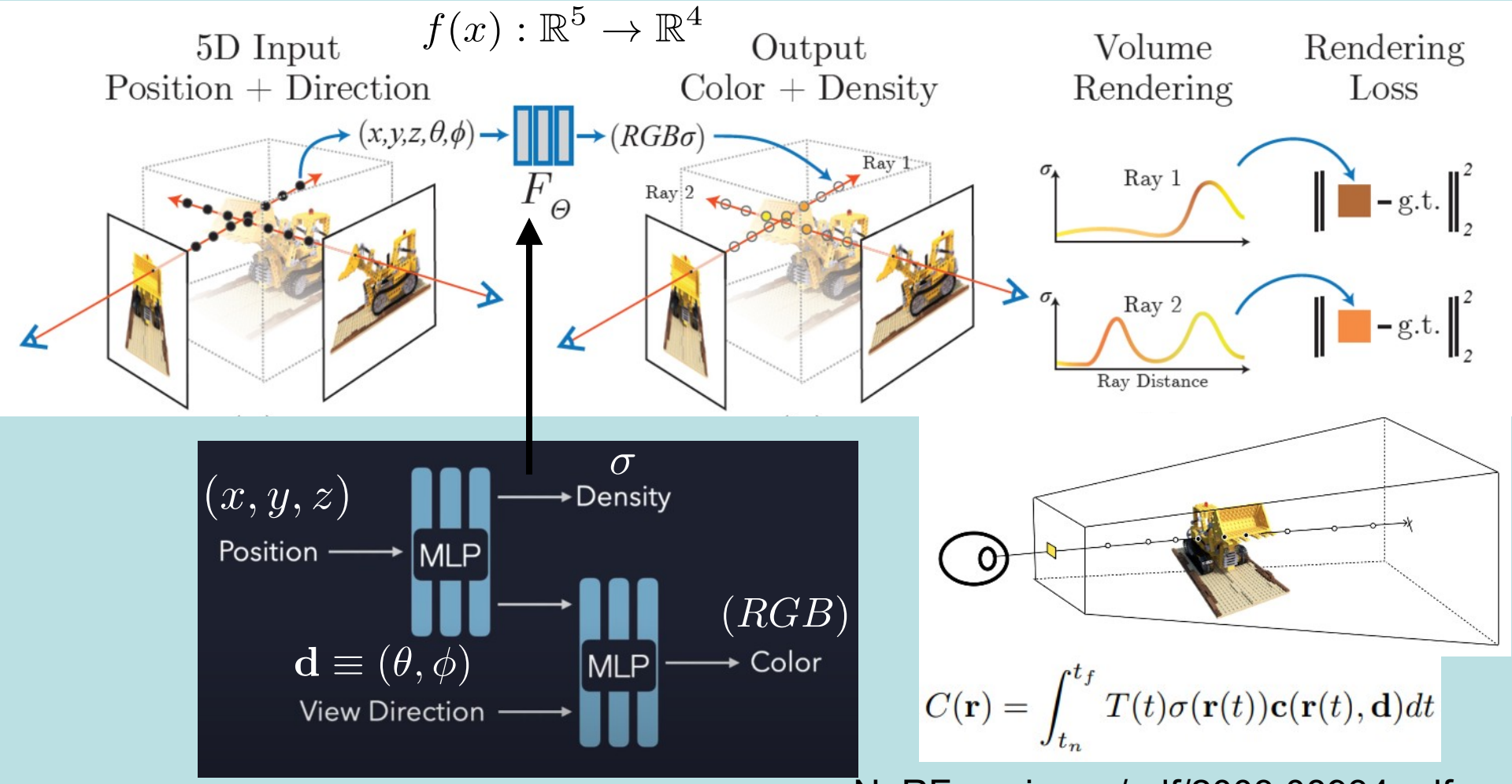


- Sinus activation

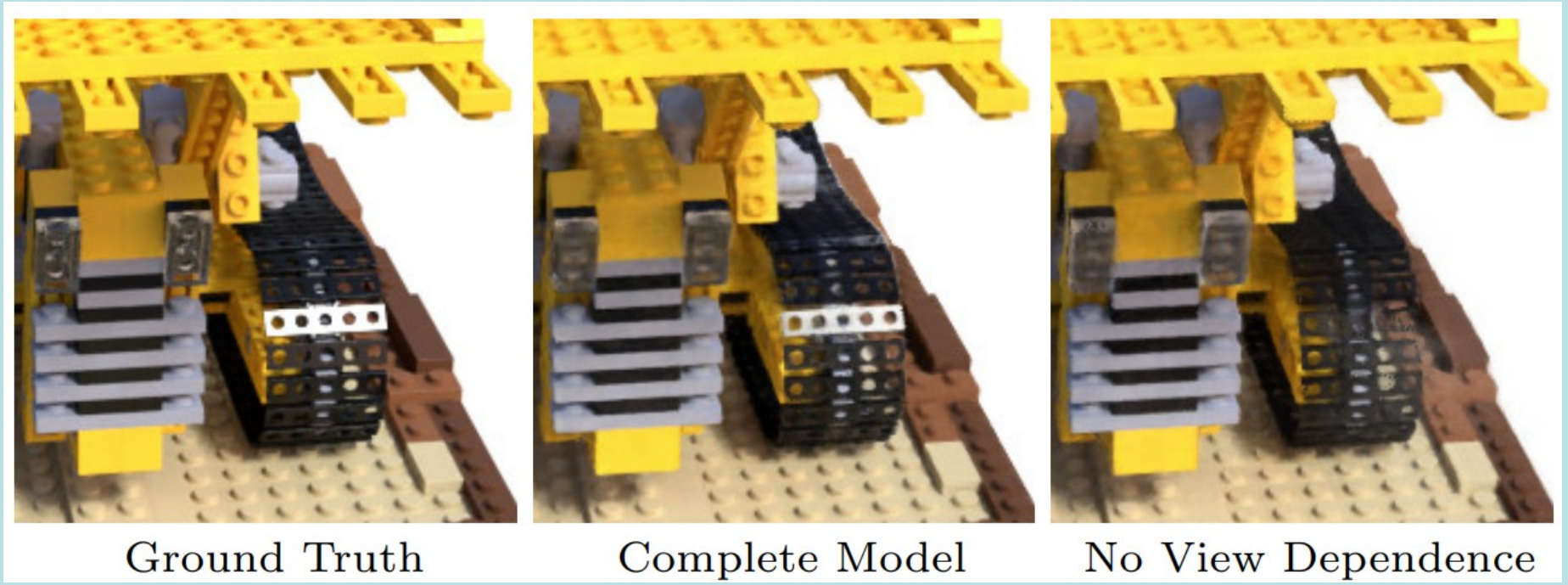




NeRF



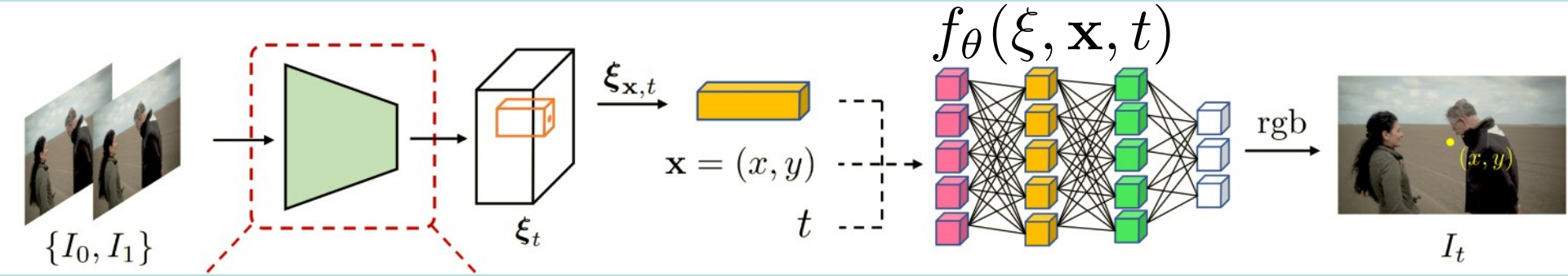
NeRF: arxiv.org/pdf/2003.08934.pdf


Ground Truth
Complete Model
No View Dependence

NVIDIA Instant NeRF

CURE

- Video



CURE: arxiv.org/pdf/2203.00137.pdf