CSC2457 3D & Geometric Deep Learning

Implicit Neural Representations with Periodic Activation Functions

Vincent Sitzmann*, Julien N. P. Martel*, Alexander Bergman, David B. Lindell,
Gordon Wetzstein

2021-03-30

Presenter: Zikun Chen

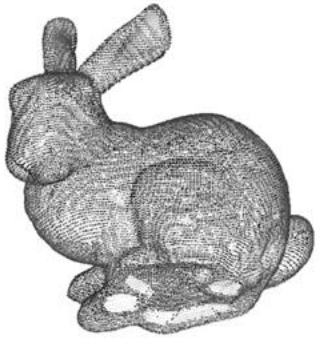
Instructor: Animesh Garg

UNIVERSITY OF TORONTO

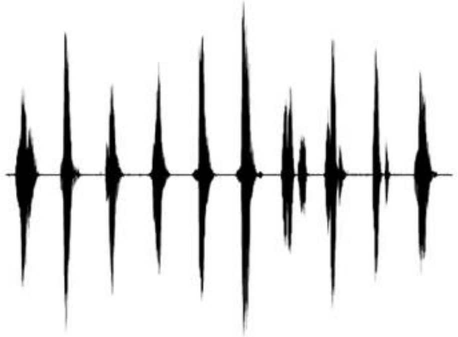
Representation of Signals: Discrete

Traditionally, discrete representations for signals are used.

Pixels



Point Cloud



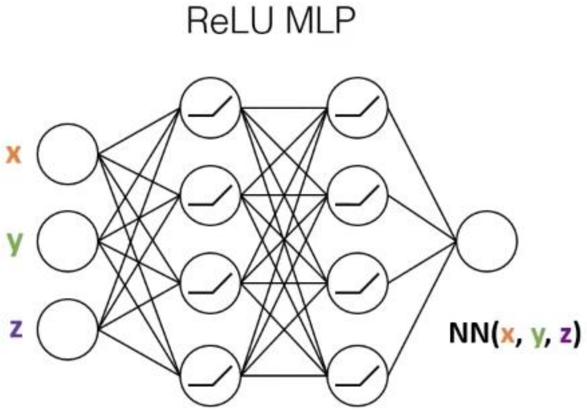
Samples of sound wave

Signal Parametrized by Neural Networks

In recent years, there's been significant research interest on implicit neural representations.

Input:

Coordinates



Implicit neural representation

Signed distance

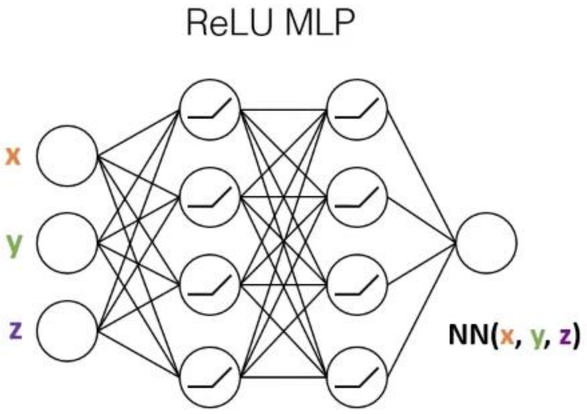
Mescheder et al. [2018]
Park et al. [2018]
Gropp et al. [2020]

Benefits of Implicit Neural Representations

- Unlike discrete representations:
  - **Agnostic** to grid resolution: model memory scales with **signal complexity**
  - **Differentiations** computed automatically

Input:

Coordinates



Implicit neural representation

Output: SDF

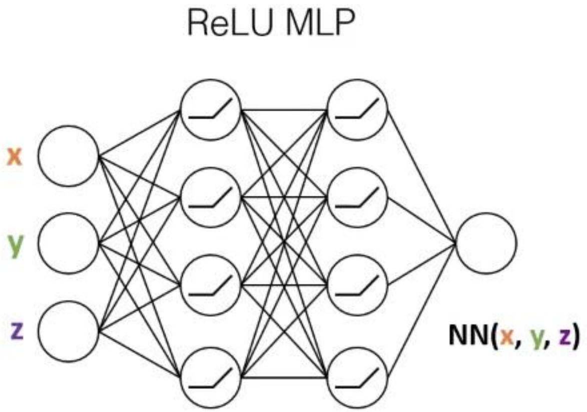
Mescheder et al. [2018]
Park et al. [2018]
Gropp et al. [2020]

Problems of Implicit Neural Representations

- Compared to discrete representations, can fail to encode **high frequency** details

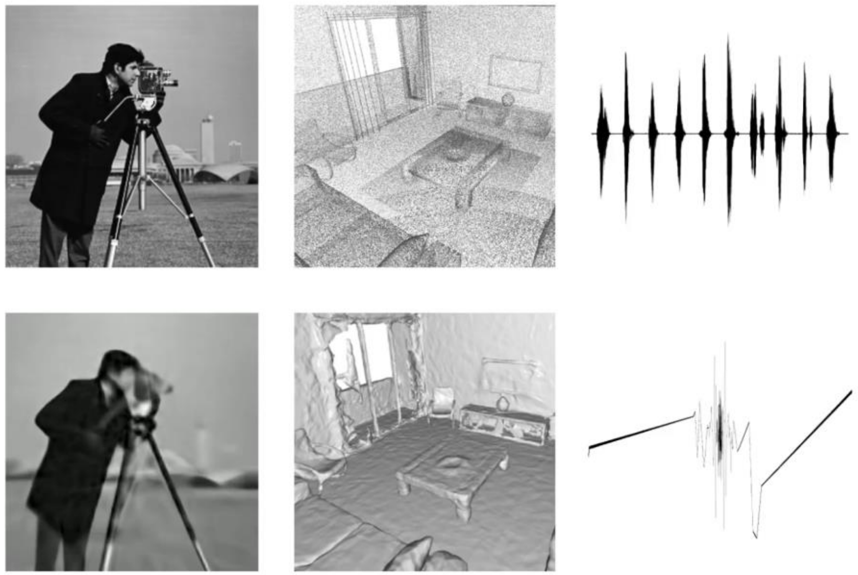
Input:

Coordinates



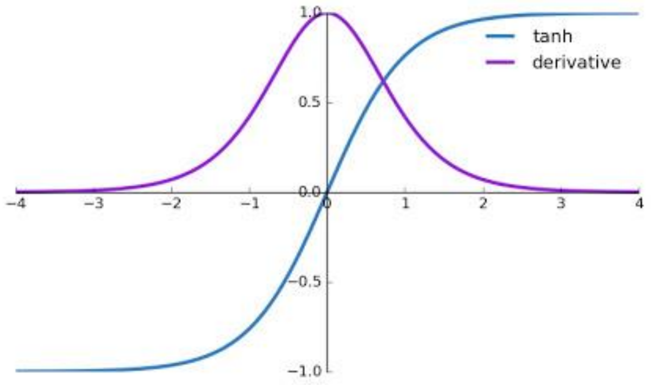
Implicit neural representation

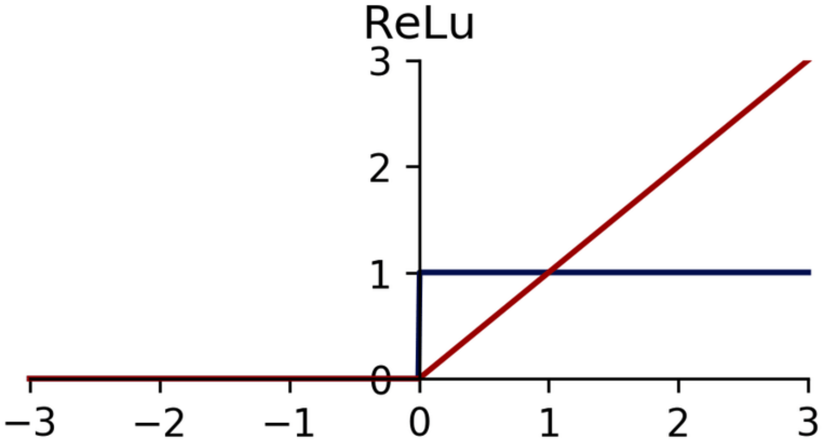
Pixels Point Cloud Samples of sound wave

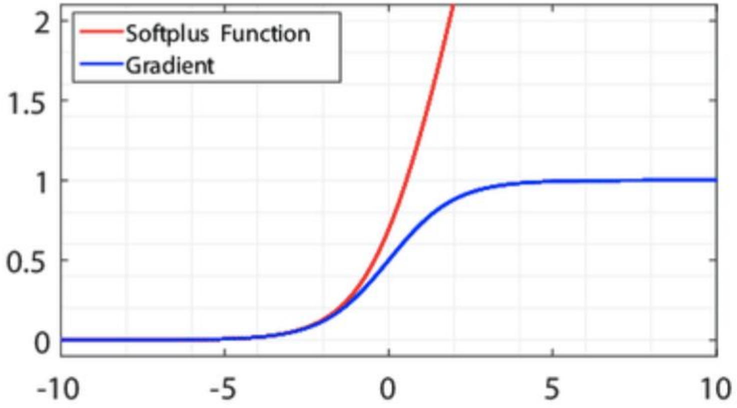


blurry edges/ missing curtains/ missing frequencies

Problems of ReLU, etc.



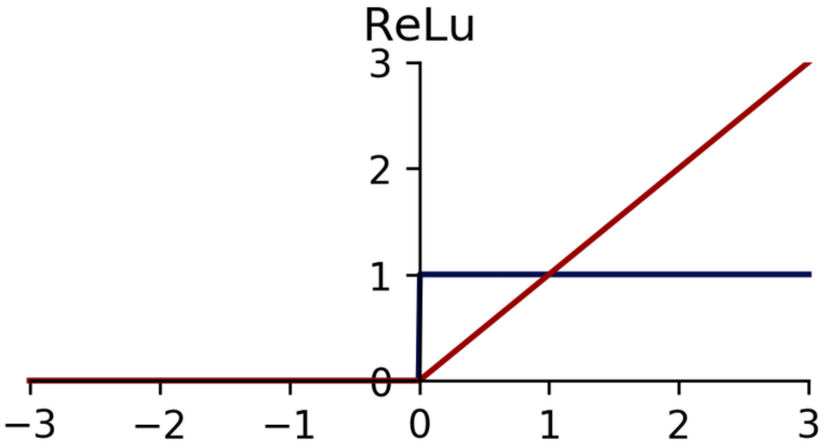


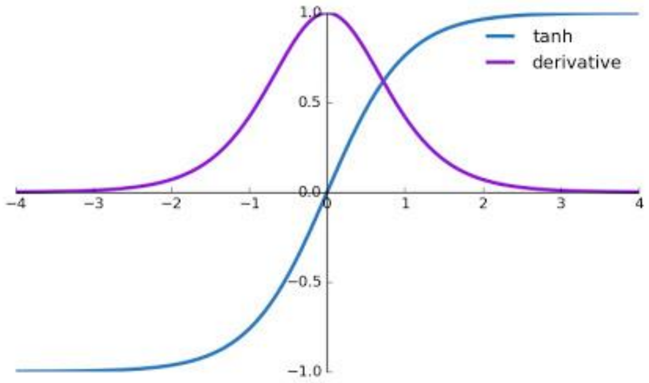


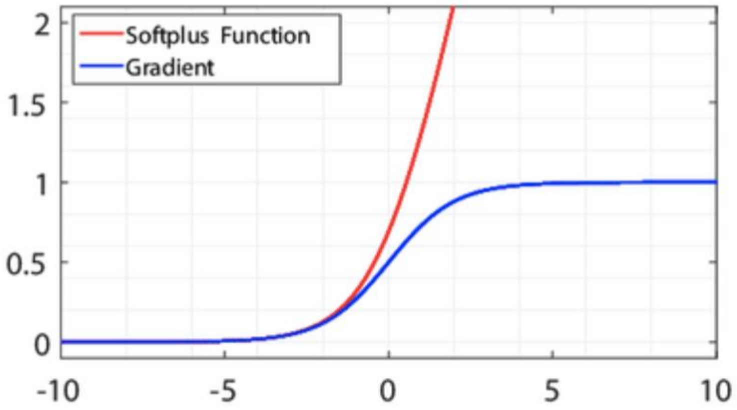
- ReLU: Linearity-> second or higher order derivatives == 0
  - Losing information in higher-order derivatives of signals

- Other activations (softplus, tanh): derivatives not well behaved

Problems of ReLU, etc.







- ReLU: Linearity-> second or higher order derivatives == 0
  - Losing information in higher-order derivatives of signals
- Other activations (softplus, tanh): derivatives not well behaved

We want: second or higher order derivatives !=0 & tractable derivative behaviours!

SIREN: Sinusoidal Representation Networks

Contributions

- Prior work:
  - periodic activation:
    - **Hypernetwork functional image representation**
      - Constructed a hypernetwork to produce weights of a target network, which parametrizes RGB images. Cosine was used as the activation function of the target network.
      - didn’t study behaviours of derivatives or other applications of cosine activation
    - **Taming the waves: sine as activation function in deep neural networks**
      - Lack of preliminary investigation of potential benefits of sine
- SIREN proposes:
  - A simple MLP architecture for implicit neural representations that uses **sine** as activation function
- Compared to prior work, this paper:
  - Proposes a continuous implicit neural representation using periodic activation that fits complicated natural signals, as well as their **derivatives**, robustly
  - Provides an initialization scheme for this type of network and validates that weights can be learned using hypernetworks
  - Demonstrates a wide range of applications

Problem Setting

Find a class of functions Φ that satisfies the relation F:

$$F\left(\mathbf{x}, \Phi, \nabla_{\mathbf{x}}\Phi, \nabla_{\mathbf{x}}^2\Phi, \ldots\right) = 0, \quad \Phi : \mathbf{x} \mapsto \Phi(\mathbf{x}).$$

On the continuous domain of x.

Φ: implicitly defined by the relation F;

Neural networks that parametrize Φ: implicit neural representations

Problem Setting

$$F\left(\mathbf{x}, \Phi, \nabla_{\mathbf{x}}\Phi, \nabla_{\mathbf{x}}^{2}\Phi, \ldots\right) = 0, \quad \Phi : \mathbf{x} \mapsto \Phi(\mathbf{x}).$$

⇩

$$\text{find } \Phi(\mathbf{x})$$

$$\text{subject to } \mathcal{C}_m\left(\mathbf{a}(\mathbf{x}), \Phi(\mathbf{x}), \nabla\Phi(\mathbf{x}), ...\right) = 0, \; \forall \mathbf{x} \in \Omega_m, \; m = 1, \ldots, M$$

Problem Setting: example

Each Input x

$$\mathbf{x} \in \mathbb{R}^2$$

spatial coords.

Output Φ(x) supervised by

$$f(\mathbf{x}) \in \mathbb{R}^3$$

RGB values

Find Φ that minimizes

$$\int_{\Omega} \|\Phi(\mathbf{x}) - f(\mathbf{x})\| \, \mathrm{d}\mathbf{x}$$

Cm: 2-norm of Φ(x)-f(x)

Problem Setting: example

Each Input x

$$\mathbf{x} \in \mathbb{R}^2$$

spatial coords.

Output Φ(x) supervised by

Edge Image

Gradients of the target image

$$\nabla f(\mathbf{x})$$

Find Φ that minimizes

$$\int_{\Omega} \|\nabla\Phi(\mathbf{x}) - \nabla f(\mathbf{x})\| \, \mathrm{d}\mathbf{x}$$

Cm: 2-norm of Φ’(x)-f’(x)

SIREN: Architecture

$$\Phi\left(\mathbf{x}\right) = \mathbf{W}_n\left(\phi_{n-1} \circ \phi_{n-2} \circ \ldots \circ \phi_0\right)\left(\mathbf{x}\right) + \mathbf{b}_n, \quad \mathbf{x}_i \mapsto \phi_i\left(\mathbf{x}_i\right) = \sin\left(\mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i\right)$$

$\phi_i : \mathbb{R}^{M_i} \mapsto \mathbb{R}^{N_i}$ is the i^{th} layer of the network

$$\mathbf{W}_i \in \mathbb{R}^{N_i \times M_i}$$

$$\mathbf{b}_i \in \mathbb{R}^{N_i}$$

$$\mathbf{x}_i \in \mathbb{R}^{M_i}$$

- Basically MLPs with Sine activations

SIREN: Initialization scheme

- **Crucial.** Without carefully chosen uniformly distributed weights, SIREN doesn’t perform well
- Key idea: preserve the distribution of activations, such that the final output at initialization does not depend of the number of layers

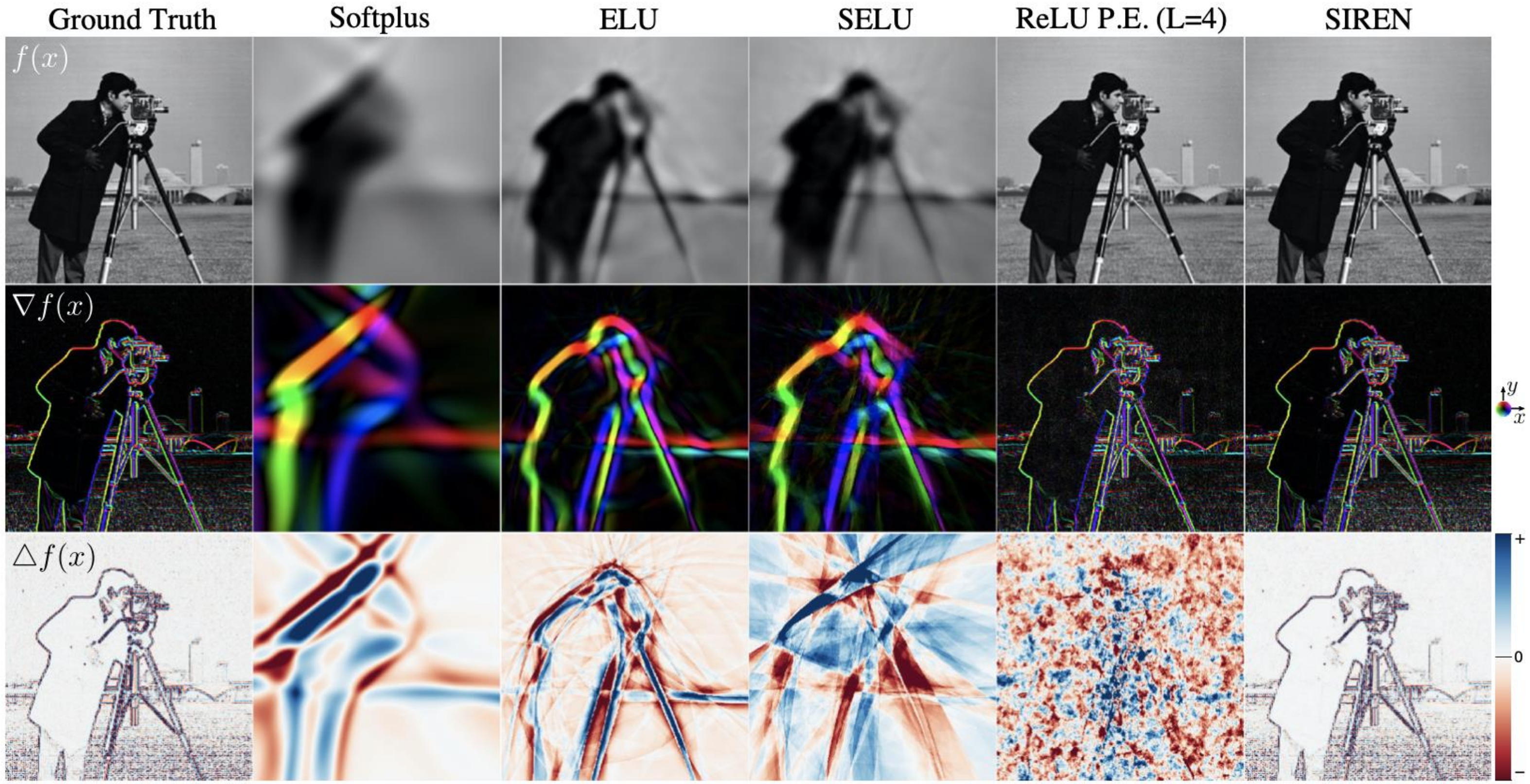
SIREN: Initialization scheme

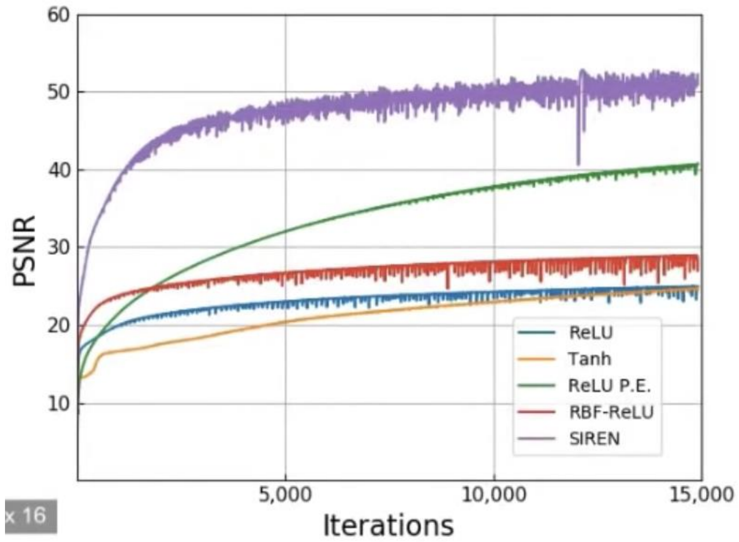
- Assuming that x~U(-1,1), x ϵ $\mathbf{x} \in \mathbb{R}^n$
- Initialize the weights W of the first layer such that sin(30*Wx+b) spans multiple periods over [-1,1]

SIREN: Initialization scheme

- Assuming that x~U(-1,1), x ($\mathbf{x} \in \mathbb{R}^n$
- For other layers with input x in n-dimensional space, initialize weights according to U(-√(⍰ 6/n), √(⍰ 6/n))
    - Ensures that input to each activation follows N(0,1) approximately

Experiments: Image reconstruction

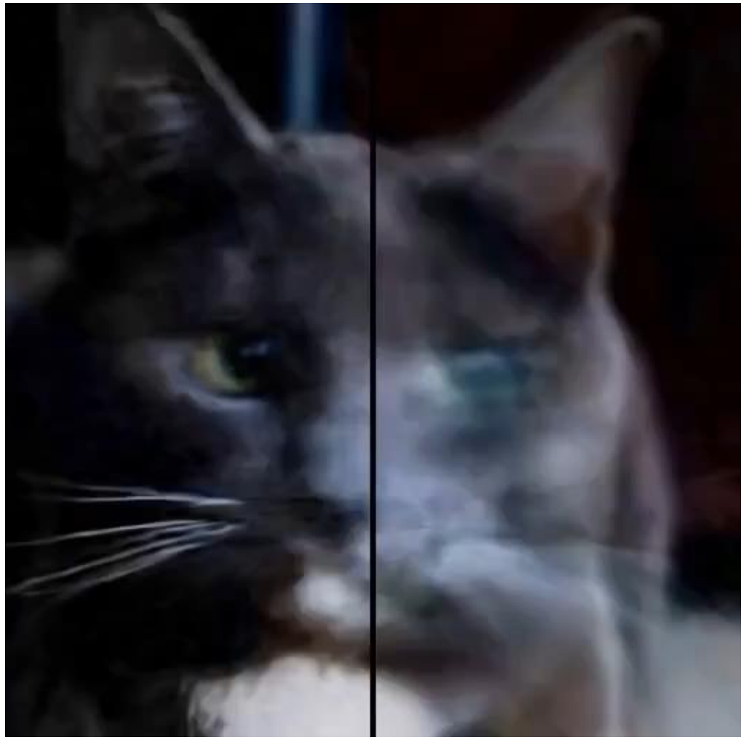




- Derivatives not well behaved: Losing gradient informations in reconstructed images

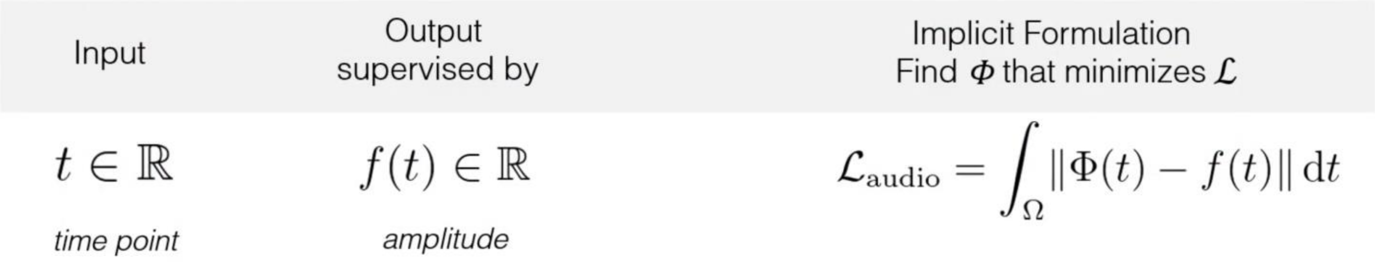
- SIREN: gradient/ laplacian well preserved

Experiments: Video reconstruction

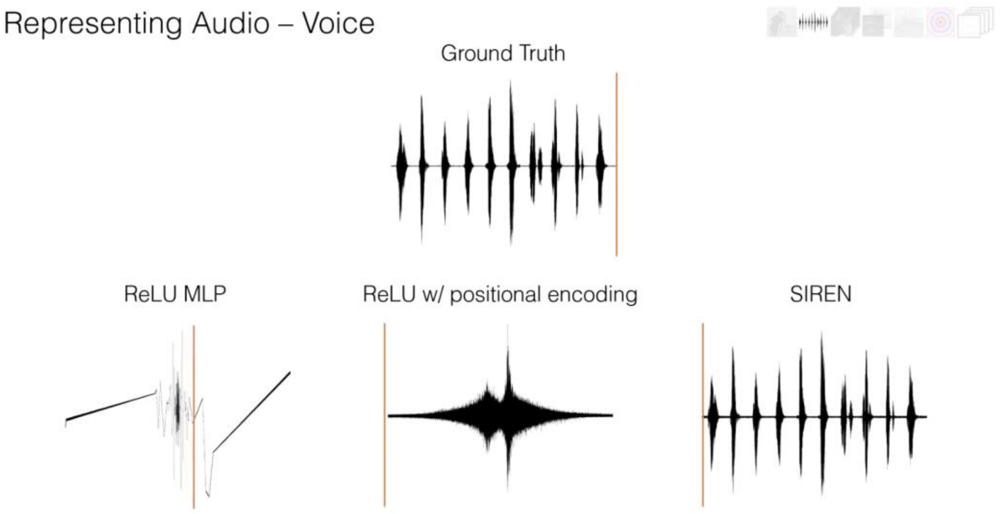


- SIREN: more high frequency details preserved

Experiments: Audio reconstruction



Input	Output supervised by	Implicit Formulation Find Φ that minimizes $\mathcal{L}$
$t \in \mathbb{R}$ *time point*	$f(t) \in \mathbb{R}$ *amplitude*	$\mathcal{L}_{\text{audio}} = \int_{\Omega} \lVert \Phi(t) - f(t) \rVert \, \mathrm{d}t$



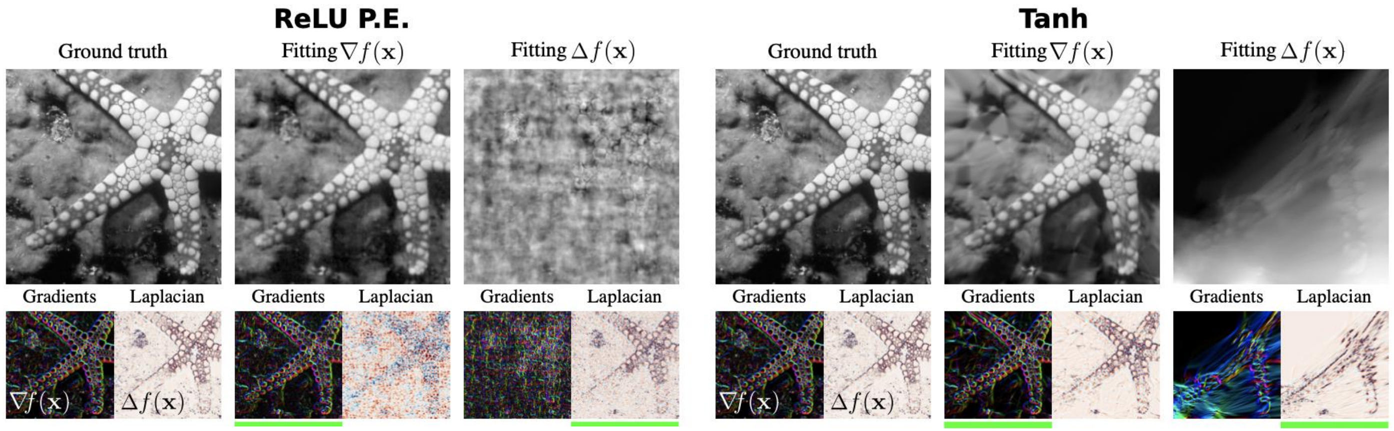
Experiments: Poisson equation

- gradients ∇f , Laplacian Δf = ∇ · ∇f

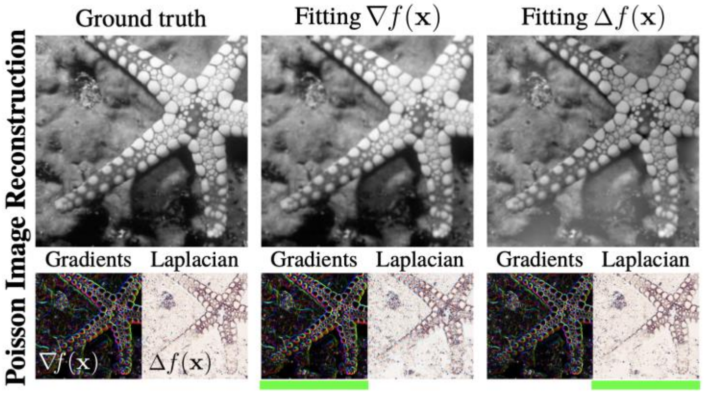
$$\mathcal{L}_{\mathrm{grad.}} = \int_{\Omega} \|\boldsymbol{\nabla}_{\mathbf{x}} \Phi(\mathbf{x}) - \boldsymbol{\nabla}_{\mathbf{x}} f(\mathbf{x})\| \, d\mathbf{x},$$

$$\text{or} \quad \mathcal{L}_{\mathrm{lapl.}} = \int_{\Omega} \|\Delta \Phi(\mathbf{x}) - \Delta f(\mathbf{x})\| \, d\mathbf{x}.$$

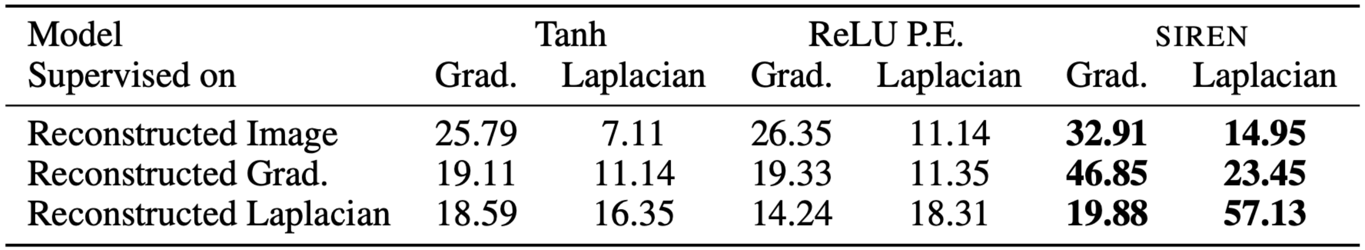
Experiments: Poisson equation- Image reconstruction



- tanh&ReLU P.E.: both failed for fitting laplacian
- SIREN: minor issues due to the ill-posed problem nature



SIREN



Model	Tanh		ReLU P.E.		SIREN	
Supervised on	Grad.	Laplacian	Grad.	Laplacian	Grad.	Laplacian
Reconstructed Image	25.79	7.11	26.35	11.14	**32.91**	**14.95**
Reconstructed Grad.	19.11	11.14	19.33	11.35	**46.85**	**23.45**
Reconstructed Laplacian	18.59	16.35	14.24	18.31	**19.88**	**57.13**

Quantitative results for PSNR

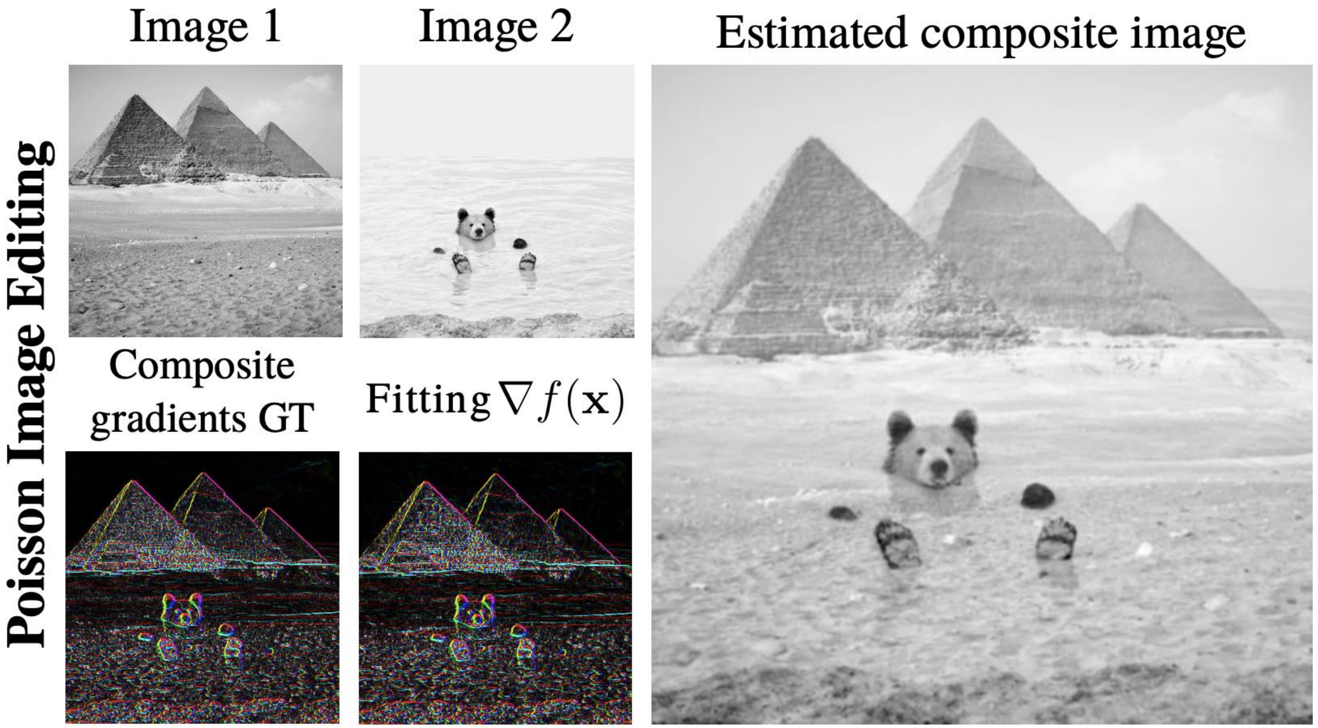
Experiments: Poisson equation- Image editing

- Fit 2 SIRENs f1, f2 for image reconstruction with loss function

$$\mathcal{L}_{\text{grad.}} = \int_{\Omega} \left\| \boldsymbol{\nabla}_{\mathbf{x}} \Phi(\mathbf{x}) - \boldsymbol{\nabla}_{\mathbf{x}} f(\mathbf{x}) \right\| \, d\mathbf{x},$$

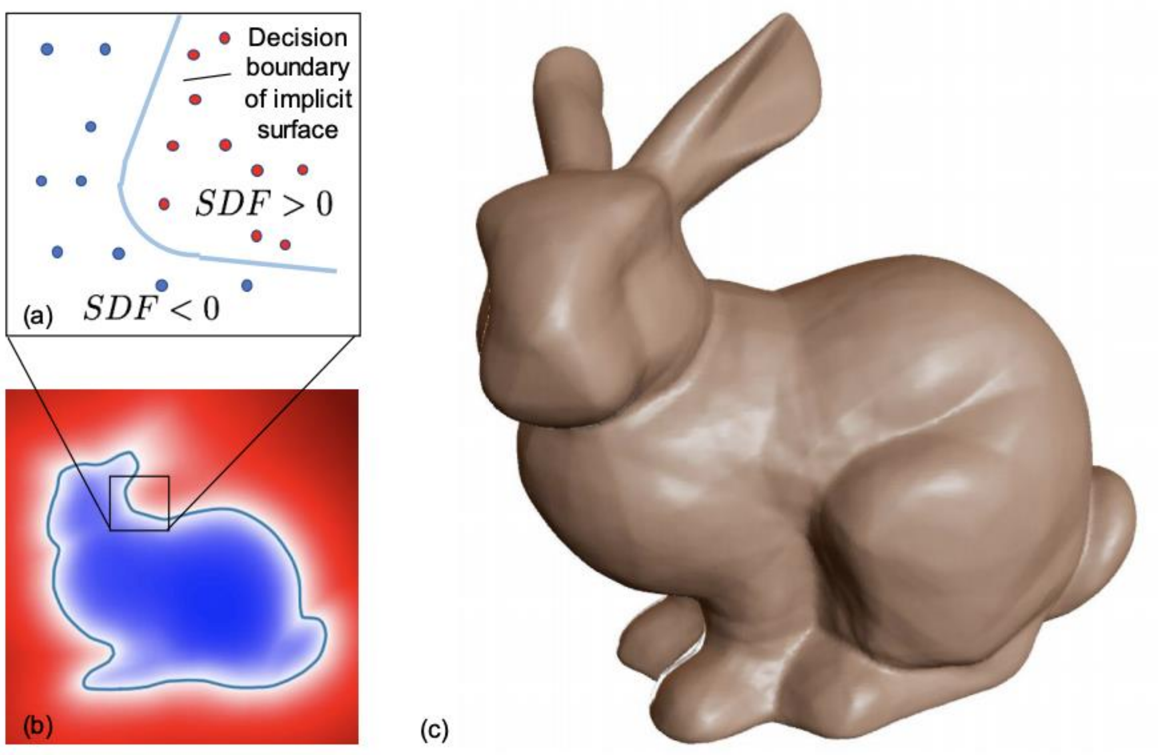
- Fit a third SIREN for image editing with the same loss function as above, except that $\nabla_x f(x)=\alpha \cdot \nabla f_1(x)+(1-\alpha) \cdot \nabla f_2(x)$, $\alpha \in [0,1]$

Experiments: Poisson equation- Image editing



Decent results with gradient supervision only!

Quick recap: Signed distance function (SDF)



Park et al., 2019

Experiments: Representing shapes with SDF

$$\mathcal{L}_{\text{sdf}} = \int_{\Omega} \big\| |\nabla_{\mathbf{x}}\Phi(\mathbf{x})| - 1 \big\| d\mathbf{x} + \int_{\Omega_0} \|\Phi(\mathbf{x})\| + \big(1 - \langle \nabla_{\mathbf{x}}\Phi(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle\big) d\mathbf{x} + \int_{\Omega \setminus \Omega_0} \psi\big(\Phi(\mathbf{x})\big) d\mathbf{x}$$

Gradient of SDF ==1

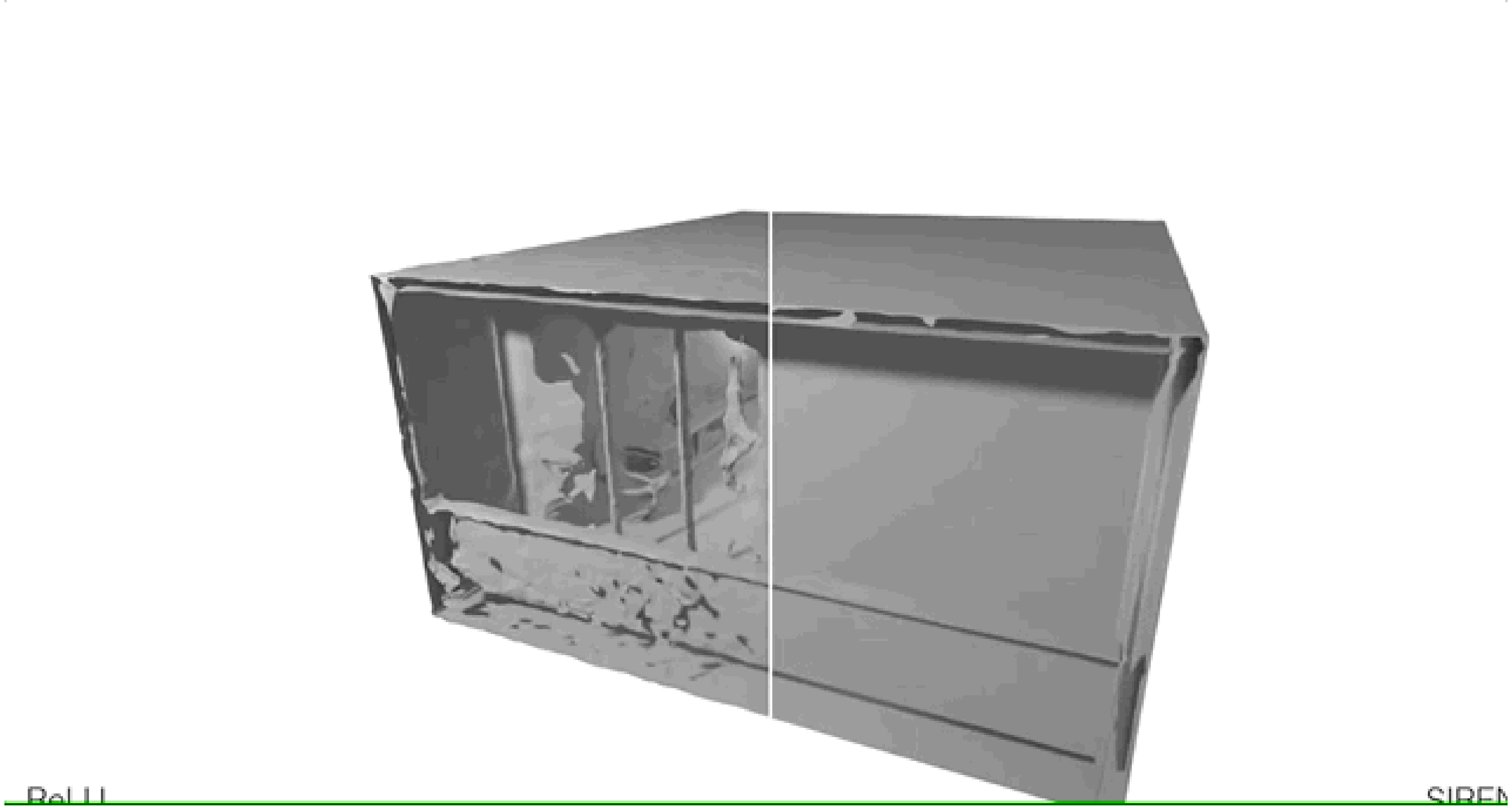
SDF = 0 when x is on the surface

$\nabla x\Phi(x)$ == normals

$\psi(x) = \exp(-\alpha \cdot |\Phi(x)|)$,
$\alpha \gg 1$
Penalize $\Phi(x)$ ==0 when x is not on the surface

Experiments: Representing shapes with SDF

Fine details missing in the baseline(left)



Experiments: Solving Differential Equations

Helmholtz equation

$$H(m)\,\underline{\Phi(\mathbf{x})} = \underline{-f(\mathbf{x})}, \textbf{ with } H(m) = \left(\Delta + \underline{m(\mathbf{x})}\,w^2\right).$$

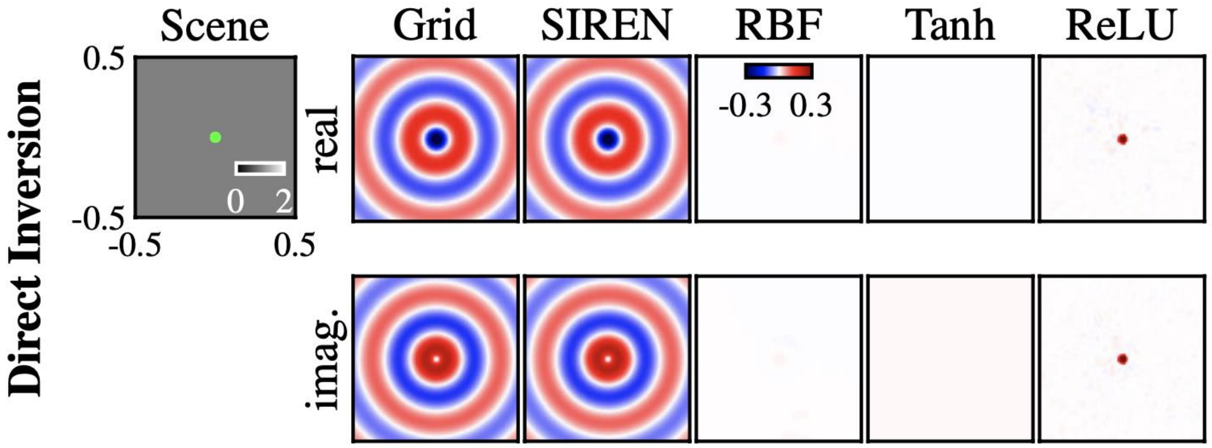
Wave field (unknown)

Known source function

1/(c(x))^2, c(x): wave velocity

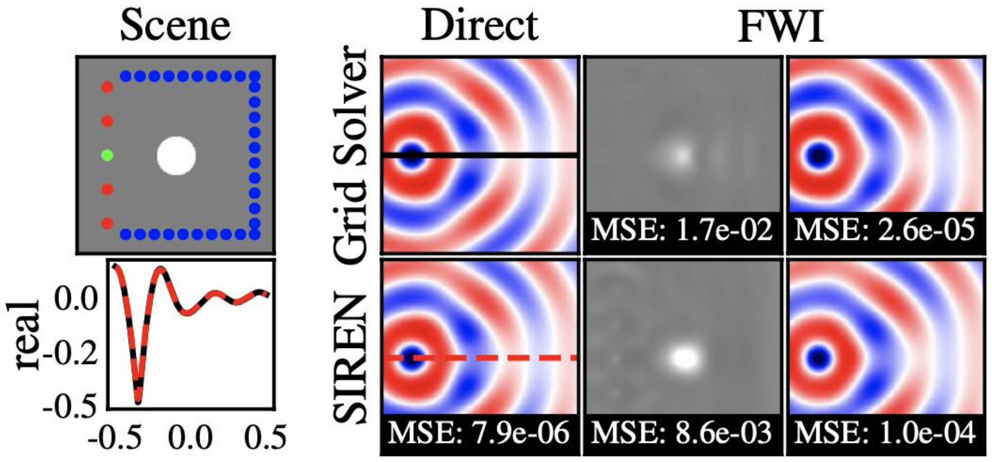
Experiments: Solving Differential Equations

c(x) known



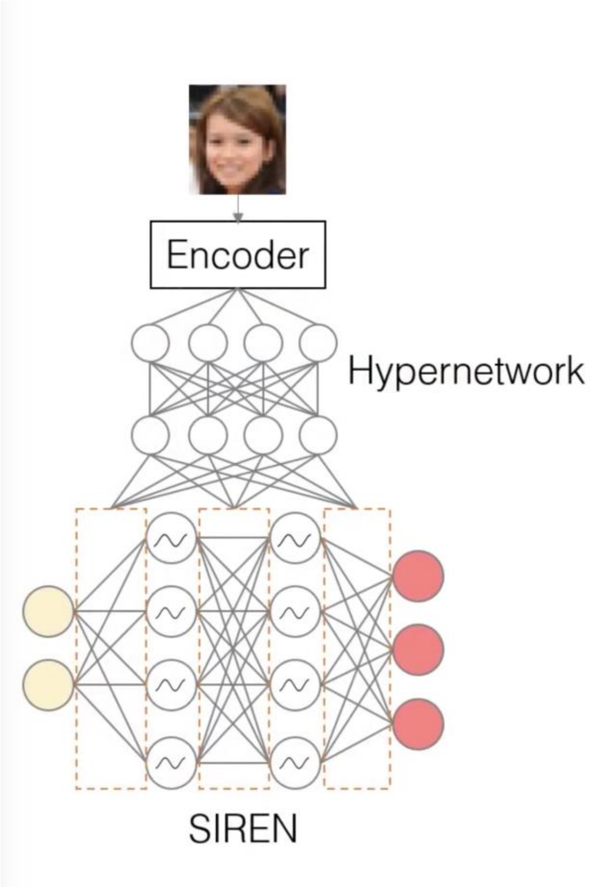
Only output of SIREN was able to match results from a numerical solver

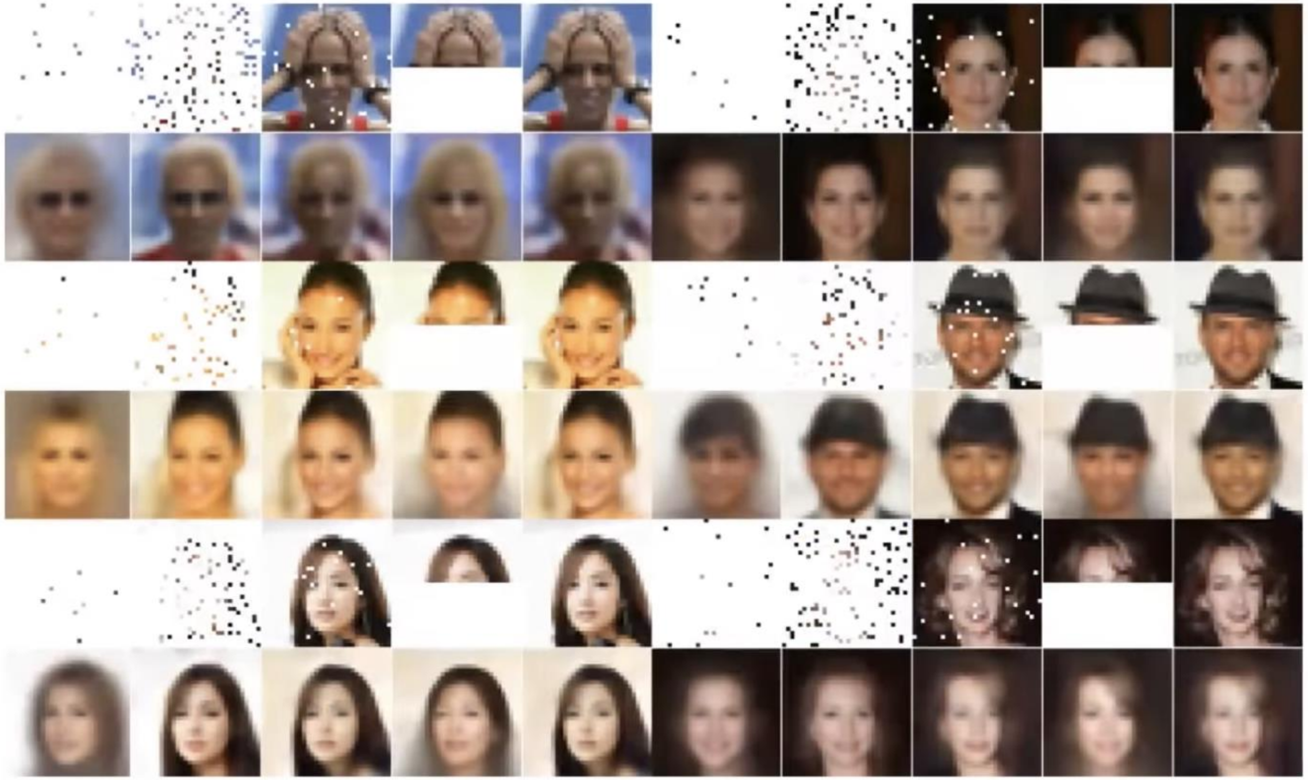
c(x) unknown



SIREN velocity model outperformed principled solver

Experiments: Learning priors





Discussions

- Poisson image editing is nothing new: Mixing gradients
- The authors didn’t give a formal definition of “well-behaved”, which is apparently an important property of sine activation.
- The authors mentioned that SIREN initialized improperly had bad performance, but didn’t link this back to the “key idea” in the initialization scheme. Could’ve done an ablation study?
- PyTorch/ Tensorflow also follows a similar default initialization scheme except that it also depends on output dimensions: sqrt(6 / (fan_in + fan_out))

Contributions (Recap)

- Prior work:
  - periodic activation:
    - **Hypernetwork functional image representation**
      - Constructed a hypernetwork to produce weights of a target network, which parametrizes RGB images. Cosine was used as the activation function of the target network.
      - didn’t study behaviours of derivatives or other applications of cosine activation
    - **Taming the waves: sine as activation function in deep neural networks**
      - Lack of preliminary investigation of potential benefits of sine
- SIREN proposes:
  - A simple MLP architecture for implicit neural representations that uses **sine** as activation function
- Compared to prior work, this paper:
  - Proposes a continuous implicit neural representation using periodic activation that fits complicated natural signals, as well as their **derivatives**, robustly
  - Provides an initialization scheme for this type of network and validates that weights can be learned using hypernetworks
  - Demonstrates a wide range of applications