Neural Implicit Fields

Implicit vs explicit equations

- Explicit representations of a curve
  - $y = f(x)$

- Implicit representation of a curve
  - $f(x, y) = 0$

Implicit representations of 3D shape

- Shape can be represented by the *level sets* of a function $f\colon \mathbb{R}^3 \rightarrow \mathbb{R}$

- Occupancy:
  - $f(x, y, z)$ is the probability (x, y, z) is inside the object
  - Surface is given by $f(x, y, z) = 0.5$

- Signed distance fields
  - $f(x, y, z)$ is the signed distance of (x, y, z) from the surface
  - Sign is positive for points inside, negative for points outside
  - Surface is given by $f(x, y, z) = 0$

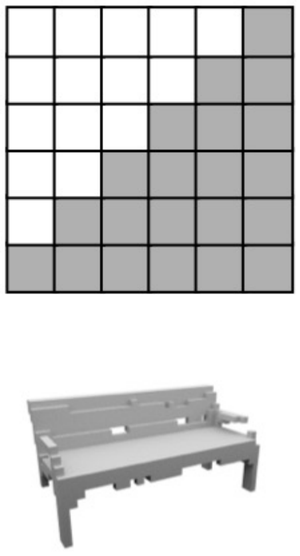
Neural implicit representations

- Traditionally f is tabular array
- But can approximate with a neural network

Mescheder, Lars, et al. "Occupancy networks: Learning 3d reconstruction in function space." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

Park, Jeong Joon, et al. "Deepsdf: Learning continuous signed distance functions for shape representation." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

Shape representations



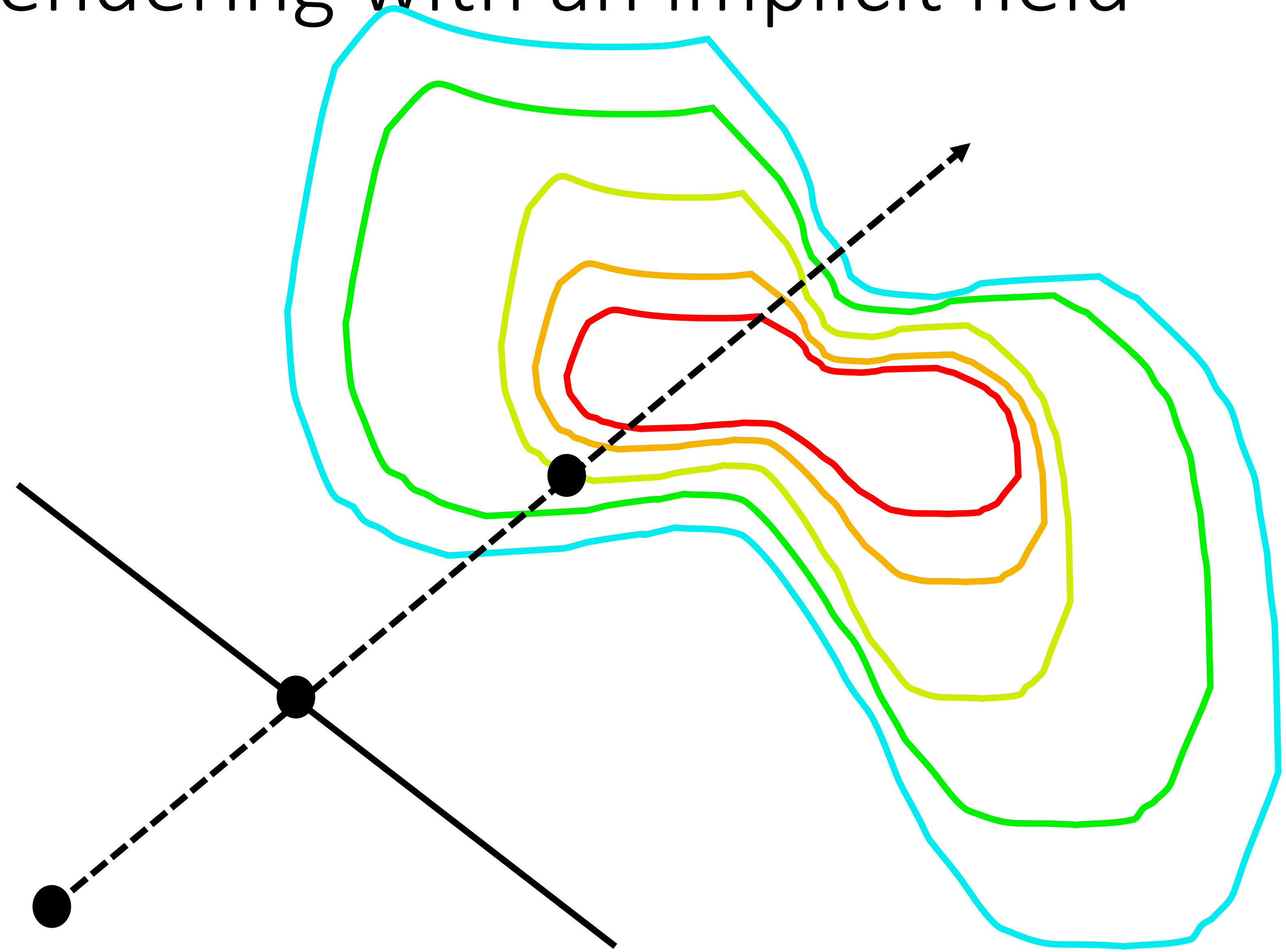
(a) Voxel

- Easy to produce
- Very expensive to store
- Limited resolution

Fitting an implicit field

- Occupancy
  - Essentially a binary classification problem
  - Sample points, label them as inside or outside the surface
- SDF
  - Essentially a regression problem
  - Sample points, label them with true signed distance
- In both cases, need watertight meshes to compute

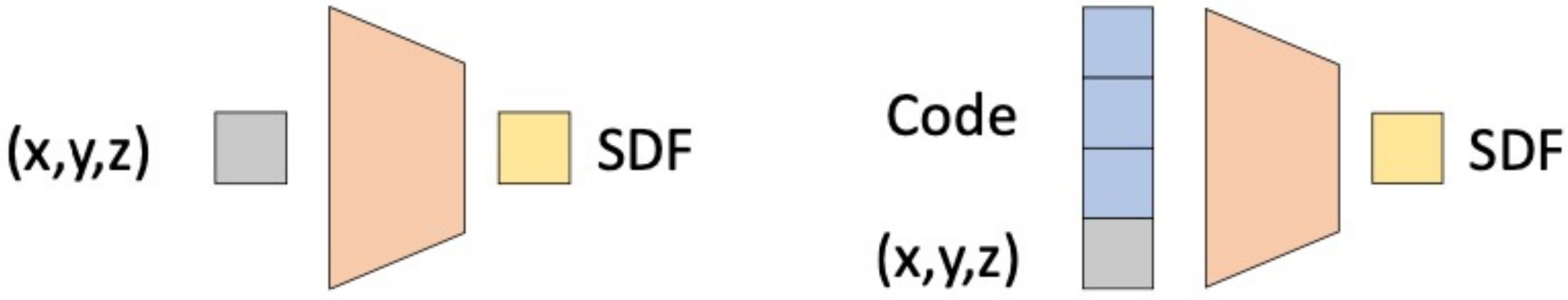
Rendering with an implicit field



Generalization with neural fields

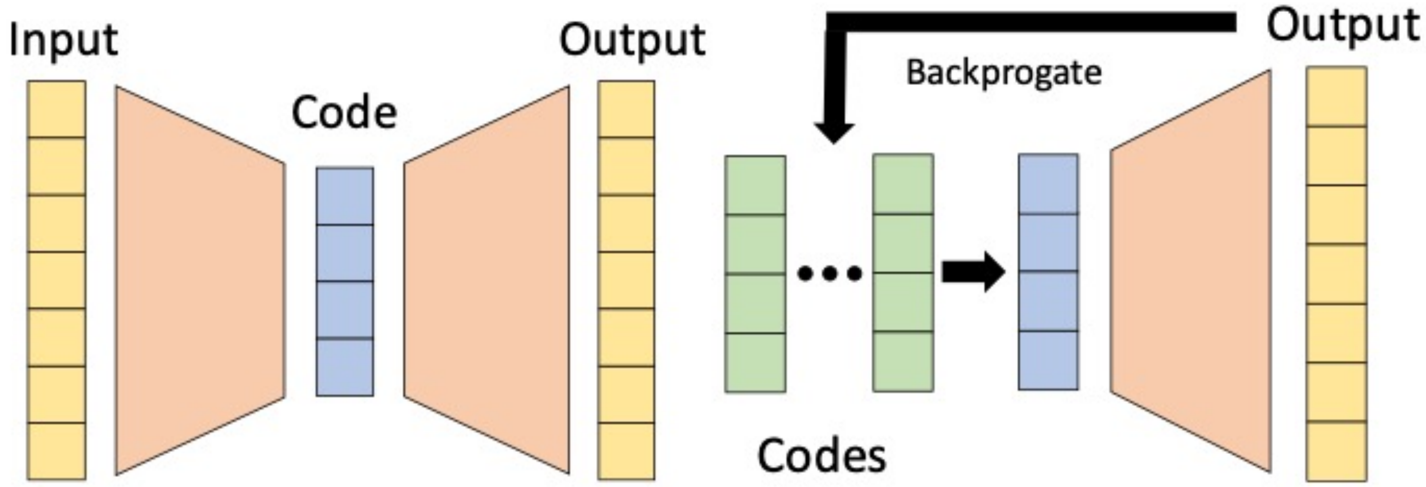
- Each neural field captures a particular shape
- Shape is encoded in the weights of the neural network
- How to generalize to new shapes?
    - Latent codes
    - Transfer learning

Implicit fields with latent codes



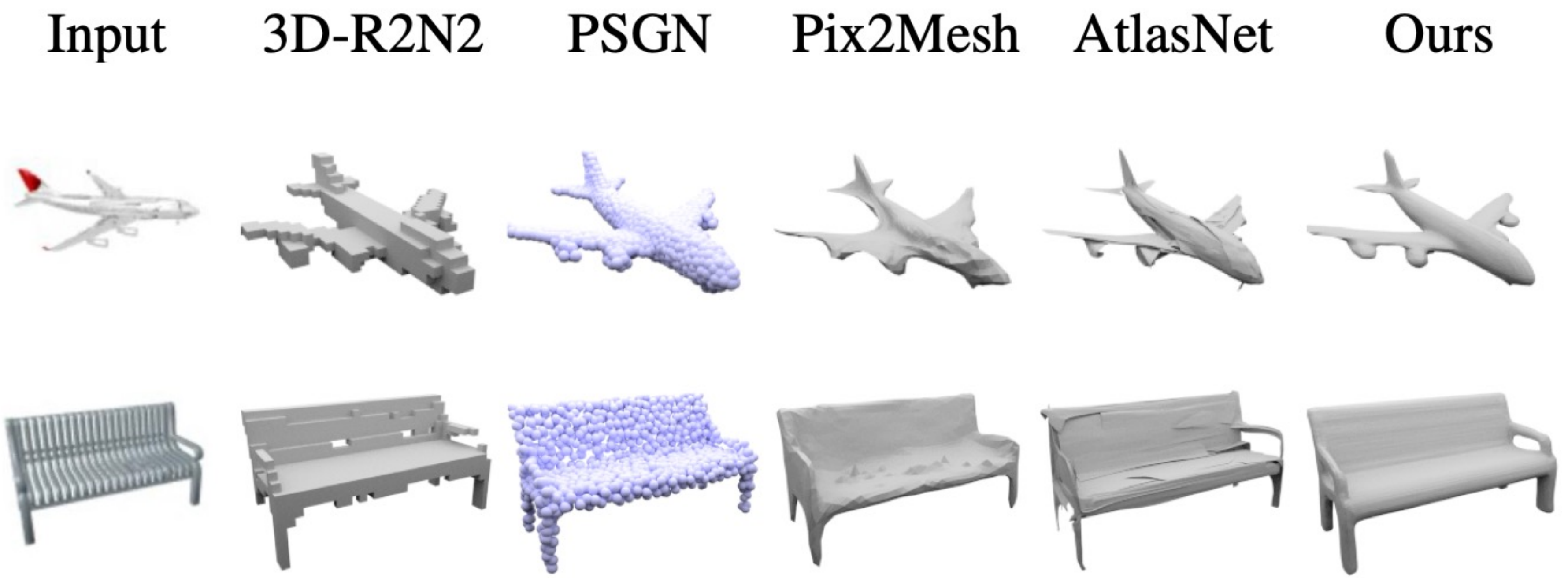
(a) Single Shape DeepSDF **(b)** Coded Shape DeepSDF

Producing latent codes for input shapes



(a) Auto-encoder **(b)** Auto-decoder

Using implicit fields for 3D reconstruction

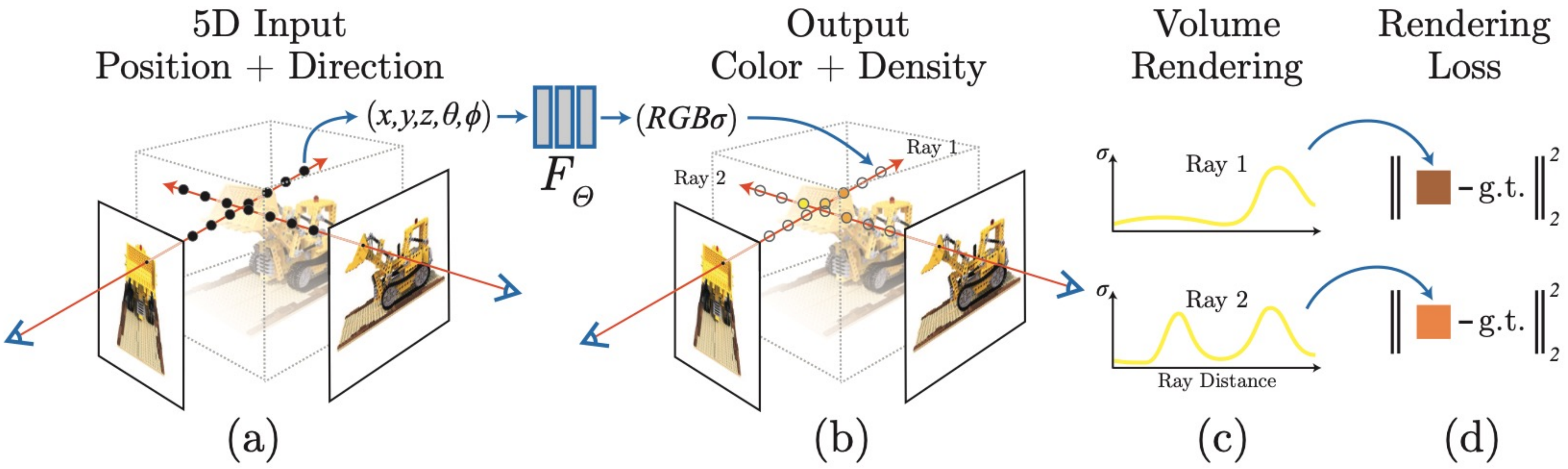


Generalizing neural fields through transfer learning

- Use meta-learning framework
- Learn *initialization for network* θ_0
- In each training iteration
  - Sample a shape
  - Perform SGD steps to update parameters to $\theta_0 + \Delta\theta$
  - Backpropagate final loss to update θ_0
- Compared to latent code approach, allows greater fidelity/cheaper networks since new shapes can use different weights

Sitzmann, Vincent, et al. "Metasdf: Meta-learning signed distance functions." *arXiv preprint arXiv:2006.09662* (2020).

Beyond shapes: Representing appearance

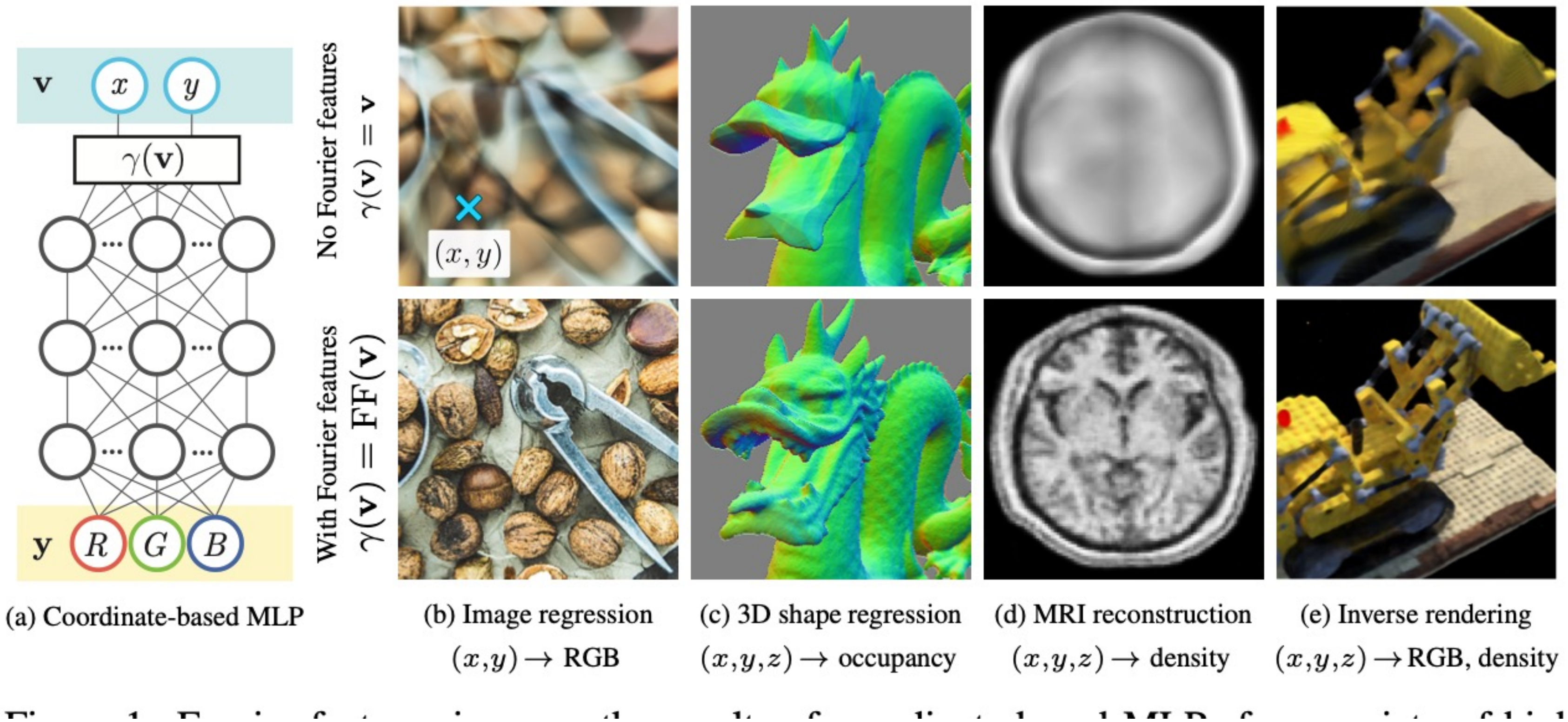


Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *European conference on computer vision*. Springer, Cham, 2020.

Representing high frequency details

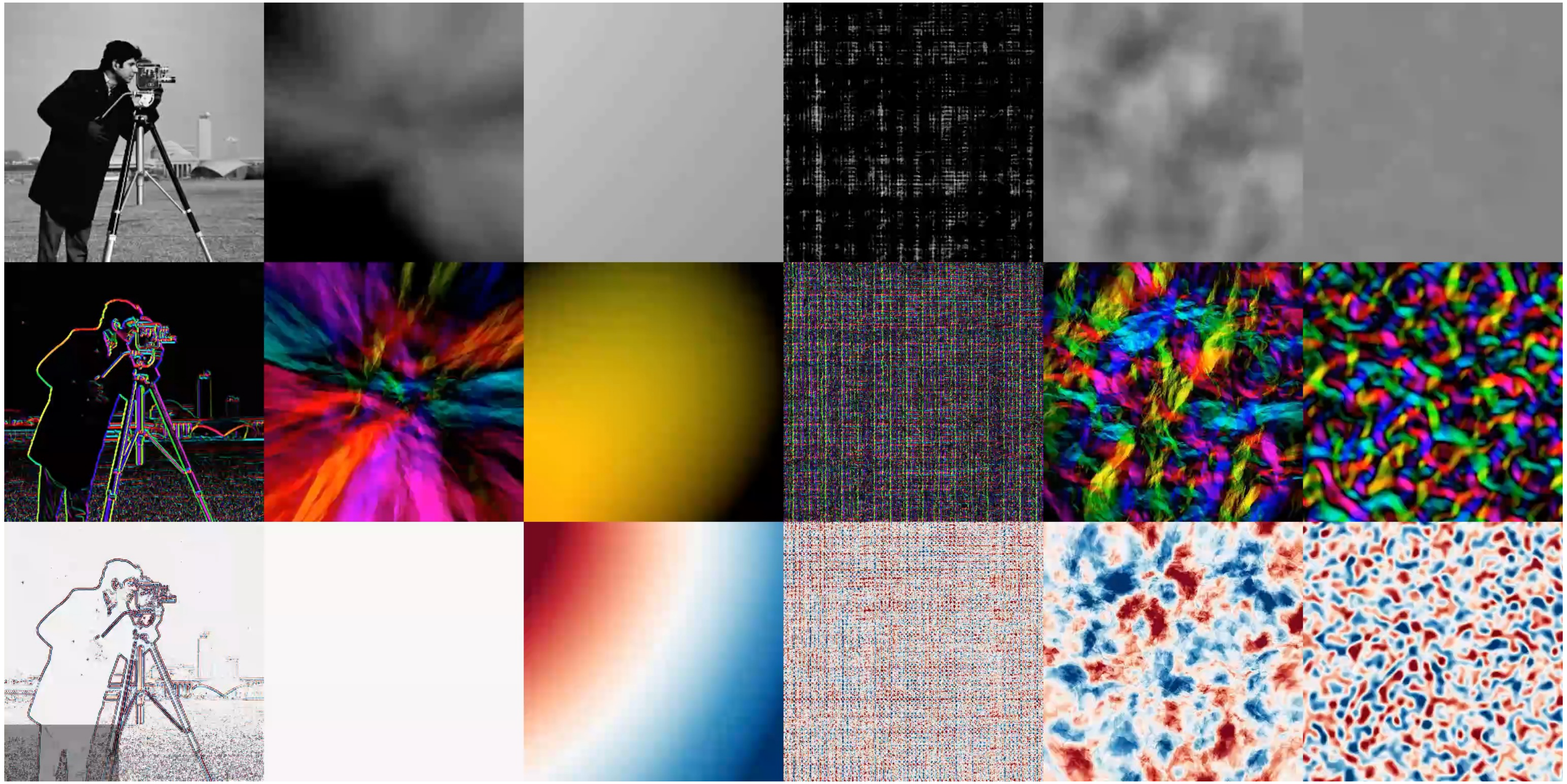
- Standard neural networks use ReLU as activation
- So they approximate functions with piecewise linear functions
- Bad idea for high-frequency signals
  - Common in images, textured 3D surfaces etc
  - Need lots and lots of pieces!

Representing high frequency details



Tancik, Matthew, et al. "Fourier features let networks learn high frequency functions in low dimensional domains." *arXiv preprint arXiv:2006.10739* (2020).

Representing high frequency details



Sitzmann, Vincent, et al. "Implicit neural representations with periodic activation functions." *Advances in Neural Information Processing Systems* 33 (2020).

Challenges with neural fields

- Shape information is stored in neural network weights
  - Difficult to edit
- Appearance information entangled with shape and pose
- Generalization across complex scenes unclear.