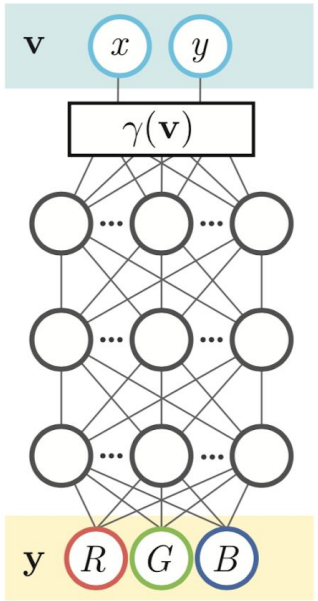
Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains

Sadrtdinov Ildus

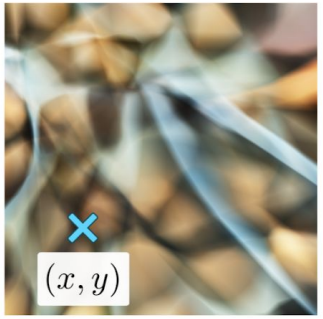
Spectral bias



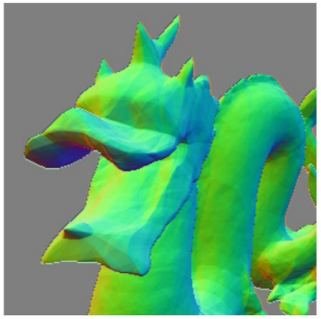
(a) Coordinate-based MLP

No Fourier features $\gamma(\mathbf{v}) = \mathbf{v}$

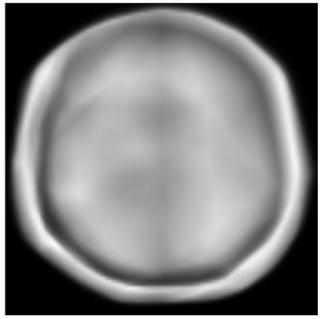
With Fourier features $\gamma(\mathbf{v}) = \mathrm{FF}(\mathbf{v})$



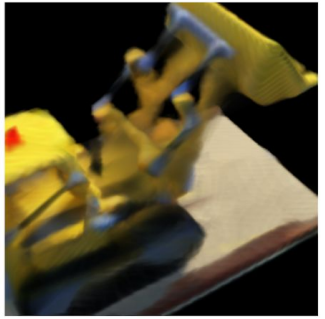
(b) Image regression
$(x,y) \rightarrow$ RGB



(c) 3D shape regression
$(x,y,z) \rightarrow$ occupancy



(d) MRI reconstruction
$(x,y,z) \rightarrow$ density

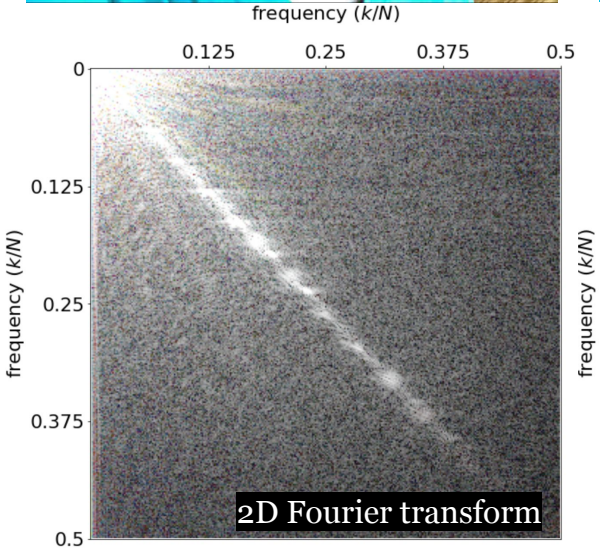


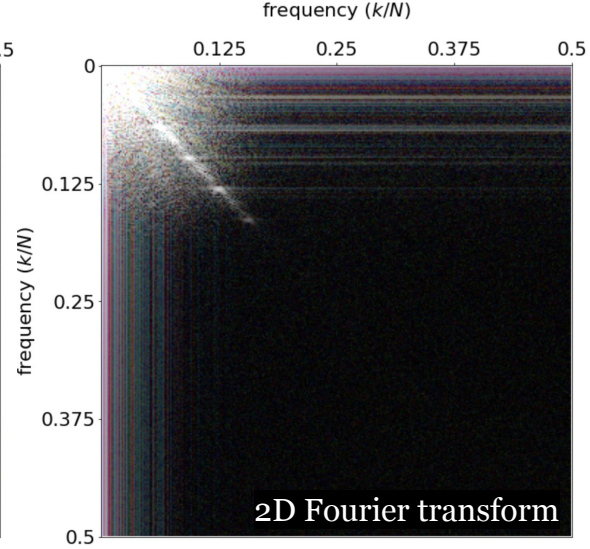
(e) Inverse rendering
$(x,y,z) \rightarrow$ RGB, density

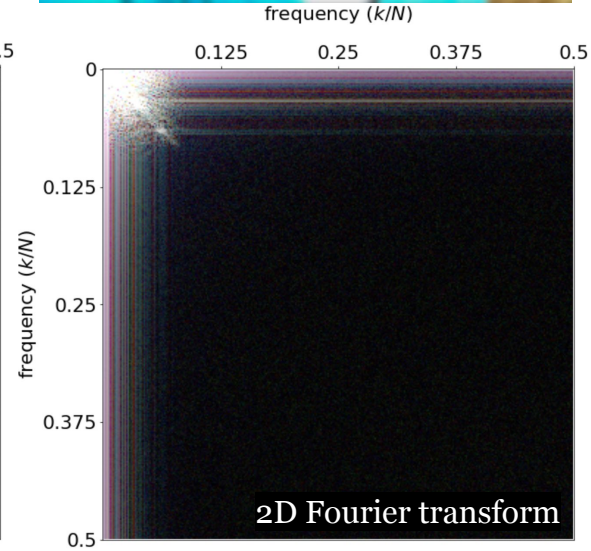

Original image


Gaussian blur (R=2)


Gaussian blur (R=4)


frequency (k/N)
0.125
0.25
0.375
0.5
0
0.125
0.25
0.375
0.5
frequency (k/N)
2D Fourier transform


frequency (k/N)
0.125
0.25
0.375
0.5
0
0.125
0.25
0.375
0.5
frequency (k/N)
2D Fourier transform


frequency (k/N)
0.125
0.25
0.375
0.5
0
0.125
0.25
0.375
0.5
frequency (k/N)
2D Fourier transform

Kernel regression

$$\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

$$f(\mathbf{x}) \approx \hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \left(\mathbf{K}^{-1} y\right)_i k(\mathbf{x}_i, \mathbf{x})$$

$$\hat{\mathbf{y}} = \hat{f}(\mathbf{X}_{\text{test}}) = \mathbf{K}_{\text{test}} \mathbf{K}^{-1} \mathbf{y}$$

$\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ - train
$f(\cdot)$ - ground truth, $\hat{f}(\cdot)$ - model
$k(\cdot, \cdot)$ - kernel
$\mathbf{K}$ - kernel matrix between train points
$\mathbf{K}_{\text{test}}$ - kernel matrix between train and test points
$\hat{\mathbf{y}}$ - test predictions

NTK theory

$$k_{\mathrm{NTK}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{E}_{\theta \sim \mathcal{N}} \left\langle \frac{\partial f(\mathbf{x}_i; \theta)}{\partial \theta}, \frac{\partial f(\mathbf{x}_j; \theta)}{\partial \theta} \right\rangle$$

$$\hat{\mathbf{y}}^{(t)} = f(\mathbf{X}_{\mathrm{test}}; \theta)$$

$$\hat{\mathbf{y}}^{(t)} \approx \mathbf{K}_{\mathrm{test}} \mathbf{K}^{-1} \left(\mathbf{I} - e^{-\eta \mathbf{K} t} \right) \mathbf{y}$$

$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ - train
$f(\cdot)$ - MLP, θ - parameters
$k_{\mathrm{NTK}}(\cdot, \cdot)$ - NTK
$\mathbf{K}$ - NTK matrix between train points
$\mathbf{K}_{\mathrm{test}}$ - NTK matrix between train and test points
η - learning rate, t - training time
$\hat{\mathbf{y}}^{(t)}$ - test predictions

MLP convergence

$$\hat{\mathbf{y}}_{\text{train}}^{(t)} = \mathbf{K}\mathbf{K}^{-1}\left(\mathbf{I} - e^{-\eta\mathbf{K}t}\right)\mathbf{y} = \left(\mathbf{I} - e^{-\eta\mathbf{K}t}\right)\mathbf{y}$$

$$\mathbf{K} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{T}} \qquad e^{-\eta\mathbf{K}t} = \mathbf{Q}e^{-\eta\mathbf{\Lambda}t}\mathbf{Q}^{\mathrm{T}}$$

$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ - train
$\mathbf{K}$ - NTK matrix between train points
η - learning rate, t - training time
$\hat{\mathbf{y}}_{\text{train}}^{(t)}$ - train predictions
$\mathbf{K} = \mathbf{Q}^T\mathbf{\Lambda}\mathbf{Q}$ - eigendecomposition

MLP convergence

$$\hat{\mathbf{y}}_{\text{train}}^{(t)} = \mathbf{K}\mathbf{K}^{-1}\left(\mathbf{I} - e^{-\eta\mathbf{K}t}\right)\mathbf{y} = \left(\mathbf{I} - e^{-\eta\mathbf{K}t}\right)\mathbf{y}$$

$$\mathbf{K} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{T}} \qquad e^{-\eta\mathbf{K}t} = \mathbf{Q}e^{-\eta\mathbf{\Lambda}t}\mathbf{Q}^{\mathrm{T}}$$

$$\mathbf{Q}^{\mathrm{T}}(\hat{\mathbf{y}}_{\text{train}}^{(t)} - \mathbf{y}) \approx \mathbf{Q}^{\mathrm{T}}\left(\left(\mathbf{I} - e^{-\eta\mathbf{K}t}\right)\mathbf{y} - \mathbf{y}\right) = -e^{-\eta\mathbf{\Lambda}t}\mathbf{Q}^{\mathrm{T}}\mathbf{y}$$

$$\left|\mathbf{Q}^{T}\left(\hat{\mathbf{y}}_{\text{train}}^{(t)} - \mathbf{y}\right)\right|_{i} = \mathcal{O}\left(e^{-\eta\lambda_i t}\right)$$

$\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ - train
$\mathbf{K}$ - NTK matrix between train points
η - learning rate, t - training time
$\hat{\mathbf{y}}_{\text{train}}^{(t)}$ - train predictions
$\mathbf{K} = \mathbf{Q}^{T}\mathbf{\Lambda}\mathbf{Q}$ - eigendecomposition

Eigenvalues and frequencies

$$\left|\mathbf{Q}^T\left(\hat{\mathbf{y}}^{(t)}_{\text{train}} - \mathbf{y}\right)\right|_i = \mathcal{O}\left(e^{-\eta\lambda_i t}\right)$$

Physical intuition: $\lambda = \dfrac{c}{\nu}$

$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ - train
$\mathbf{K}$ - NTK matrix between train points
η - learning rate, t - training time
$\hat{\mathbf{y}}^{(t)}_{\text{train}}$ - train predictions
$\mathbf{K} = \mathbf{Q}^T\mathbf{\Lambda}\mathbf{Q}$ - eigendecomposition
λ - eigenvalue, ν - frequency

Eigenvalues and frequencies

$$\left|\mathbf{Q}^T\left(\hat{\mathbf{y}}^{(t)}_{\text{train}} - \mathbf{y}\right)\right|_i = \mathcal{O}\left(e^{-\eta\lambda_i t}\right)$$

Physical intuition: $\lambda = \dfrac{c}{\nu}$

Our case (1-D only): $\nu = k \Rightarrow \lambda = \mathcal{O}\left(\dfrac{1}{k^2}\right)$

$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ - train
$\mathbf{K}$ - NTK matrix between train points
η - learning rate, t - training time
$\hat{\mathbf{y}}^{(t)}_{\text{train}}$ - train predictions
$\mathbf{K} = \mathbf{Q}^T\mathbf{\Lambda}\mathbf{Q}$ - eigendecomposition
λ - eigenvalue, ν - frequency

Fourier Features

$$\gamma(\mathbf{v}) = \left[a_1 \cos(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{v}), a_1 \sin(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{v}), \ldots, a_m \cos(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{v}), a_m \sin(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{v})\right]^{\mathrm{T}}$$

$\mathbf{v} = [x, y, z]^T$ - coordinates input
$\gamma(\mathbf{v})$ - Fourier Features
a_j - amplitudes, $\mathbf{b}_j$ - frequencies
n - size of train
p - hyperparameter

Fourier Features

$$\gamma(\mathbf{v}) = \left[a_1 \cos(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{v}), a_1 \sin(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{v}), \ldots, a_m \cos(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{v}), a_m \sin(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{v})\right]^{\mathrm{T}}$$

1-D experiment: $a_j = \frac{1}{j^p}, b_j = j, j = 1, \ldots, n/2$

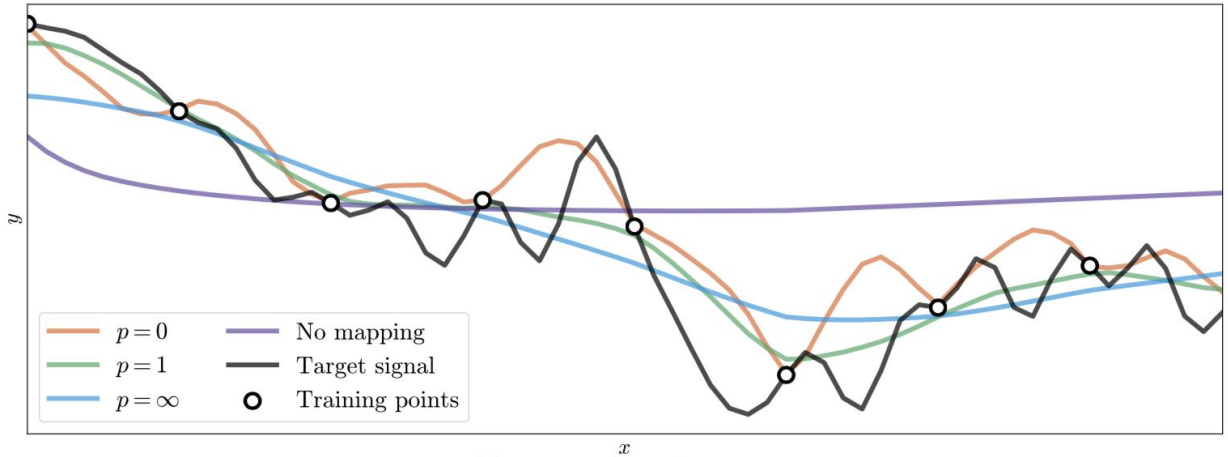
$$p = 0 \Rightarrow \gamma(v) = [\cos(2\pi v), \sin(2\pi v), \cos(4\pi v), \sin(4\pi v), \ldots, \cos(\pi n v), \sin(\pi n v)]^T$$

$$p = 1 \Rightarrow \gamma(v) = [\cos(2\pi v), \sin(2\pi v), \frac{1}{2}\cos(4\pi v), \frac{1}{2}\sin(4\pi v), \ldots, \frac{2}{n}\cos(\pi n v), \frac{2}{n}\sin(\pi n v)]^T$$

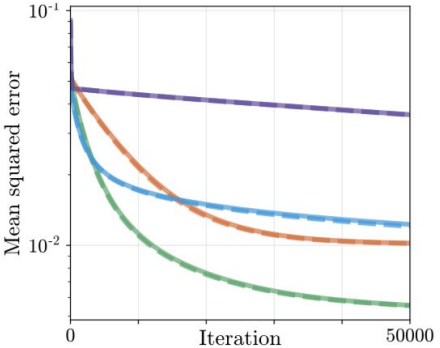
$$p = \infty \Rightarrow \gamma(v) = [\cos(2\pi v), \sin(2\pi v)]^T$$

$\mathbf{v} = [x, y, z]^T$ - coordinates input
$\gamma(\mathbf{v})$ - Fourier Features
a_j - amplitudes, $\mathbf{b}_j$ - frequencies
n - size of train
p - hyperparameter

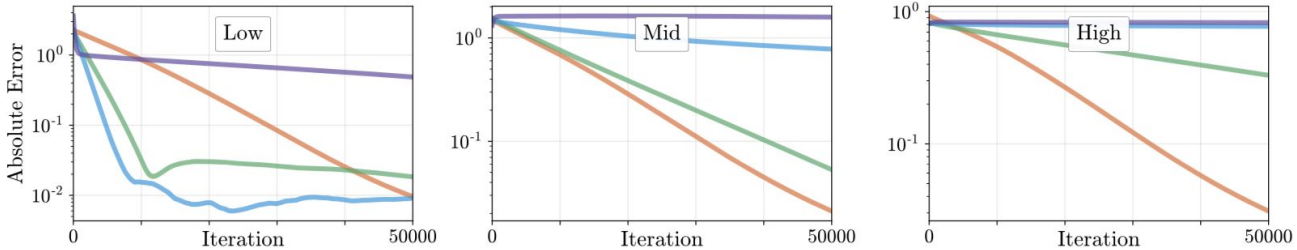
Frequency components convergence



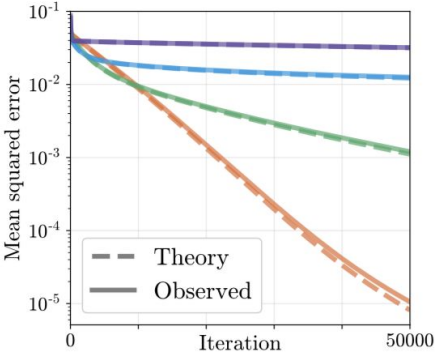
(a) Final learned functions



(b) Test loss



(c) Train loss frequency components



(d) Train loss

Scalability problem

$$v \in \mathbb{R}^1, \gamma(v) = [\ldots, a_j \cos(2\pi b_j v), a_j \sin(2\pi b_j v), \ldots]^T$$

$$b_j \in \{1, ..., n/2\} : \frac{n}{2} \text{ frequencies}$$

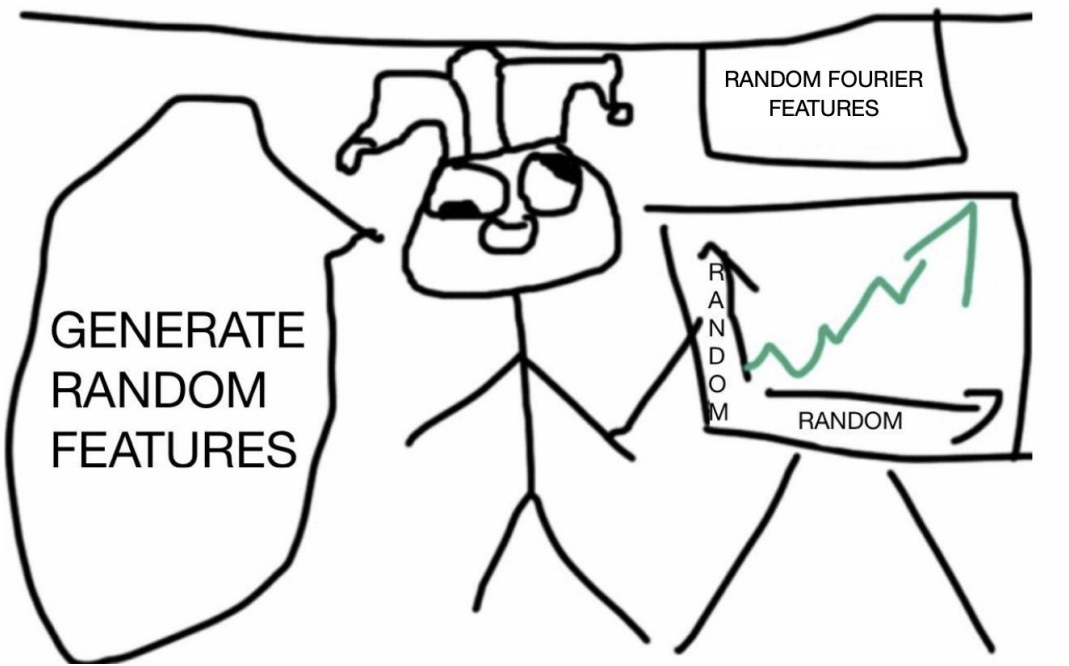
$$\mathbf{v} \in \mathbb{R}^d, \gamma(\mathbf{v}) = [\ldots, a_j \cos(2\pi \mathbf{b}_j^T \mathbf{v}), a_j \sin(2\pi \mathbf{b}_j^T \mathbf{v}), \ldots]^T$$

$$\mathbf{b}_j \in \{1, ..., n/2\}^d : \left(\frac{n}{2}\right)^d \text{ frequencies}$$

$\mathbf{v} \in \mathbb{R}^d$ - coordinates input
$\gamma(\mathbf{v}) \in \mathbb{R}^n$ - Fourier Features
a_j - amplitudes, $\mathbf{b}_j$ - frequencies

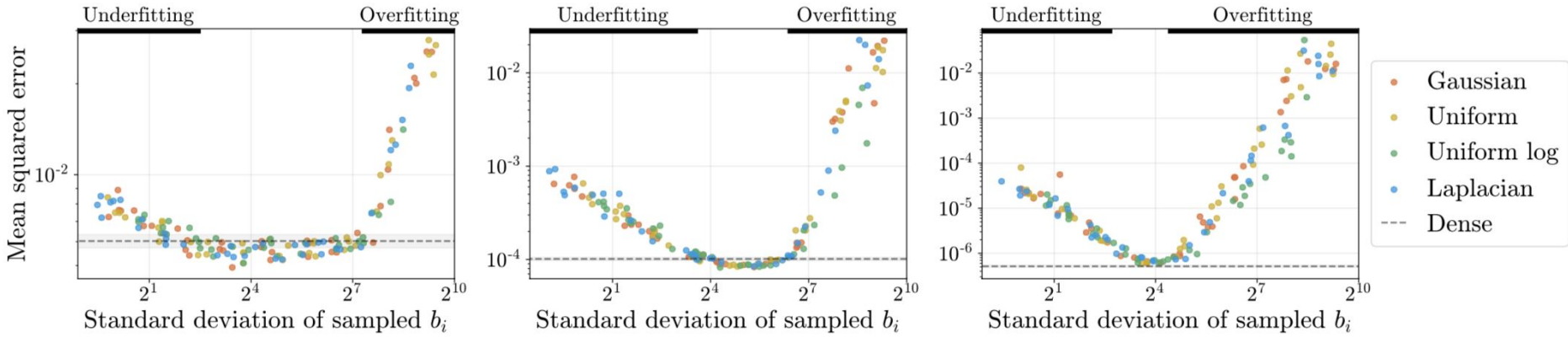
Random Fourier Features

$$\gamma(\mathbf{v}) = [\cos(2\pi\mathbf{B}\mathbf{v}), \sin(2\pi\mathbf{B}\mathbf{v})]^T, \mathbf{B} \in \mathbb{R}^{m \times d}, \mathbf{B}_{ij} \sim \mathcal{P}(\theta)$$



$\mathbf{v} \in \mathbb{R}^d$ - coordinates input
$\gamma(\mathbf{v}) \in \mathbb{R}^{2m}$ - Fourier Features
$\mathbf{B}$ - frequencies
$\mathcal{P}(\theta)$ - distributions family

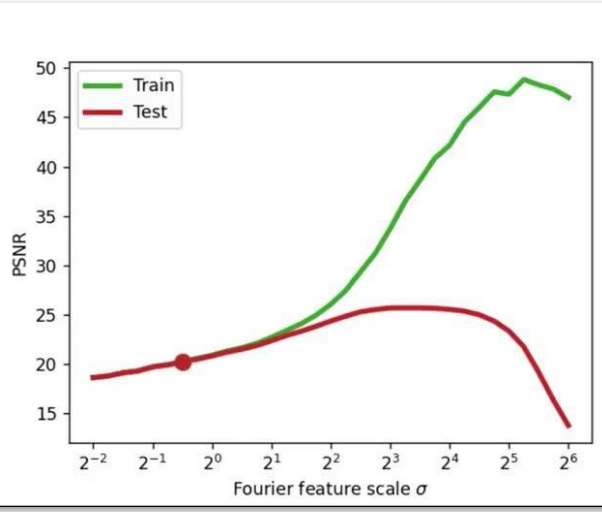
Distribution scale and overfitting

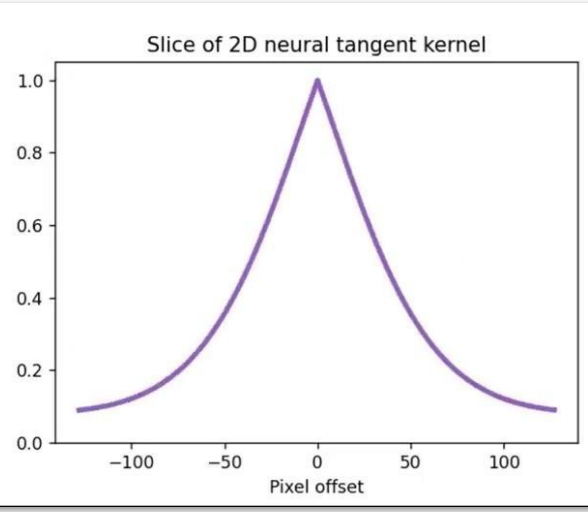


(a) Data sampled from $1/f^{0.5}$ (b) Data sampled from $1/f^{1.0}$ (c) Data sampled from $1/f^{1.5}$

Name	Sampled b_j values
Gaussian	$\sigma_g X$ for $X \sim \mathcal{N}(0,1)$
Uniform	$\sigma_u X$ for $X \sim \mathcal{U}[0,1)$
Uniform log	σ_{ul}^X for $X \sim \mathcal{U}[0,1)$
Laplacian	$\sigma_l X$ for $X \sim \text{Laplace}(0,1)$

Distribution scale and overfitting





https://people.eecs.berkeley.edu/~bmild/fourfeat/index.html

Comparing mappings

1. **Basic**: $\gamma(\mathbf{v}) = [\cos(2\pi\mathbf{v}), \sin(2\pi\mathbf{v})]^T$
2. **Positional encoding**: $\gamma(\mathbf{v}) = [\ldots, \cos(2\pi\sigma^{j/m}\mathbf{v}), \sin(2\pi\sigma^{j/m}\mathbf{v}), \ldots]^T, j = 0, \ldots, m-1$
3. **Gaussian**: $\gamma(\mathbf{v}) = [\cos(2\pi\mathbf{B}\mathbf{v}), \sin(2\pi\mathbf{B}\mathbf{v})]^T, \mathbf{B} \in \mathbb{R}^{m \times d}, \mathbf{B}_{ij} \sim \mathcal{N}(0, \sigma^2)$

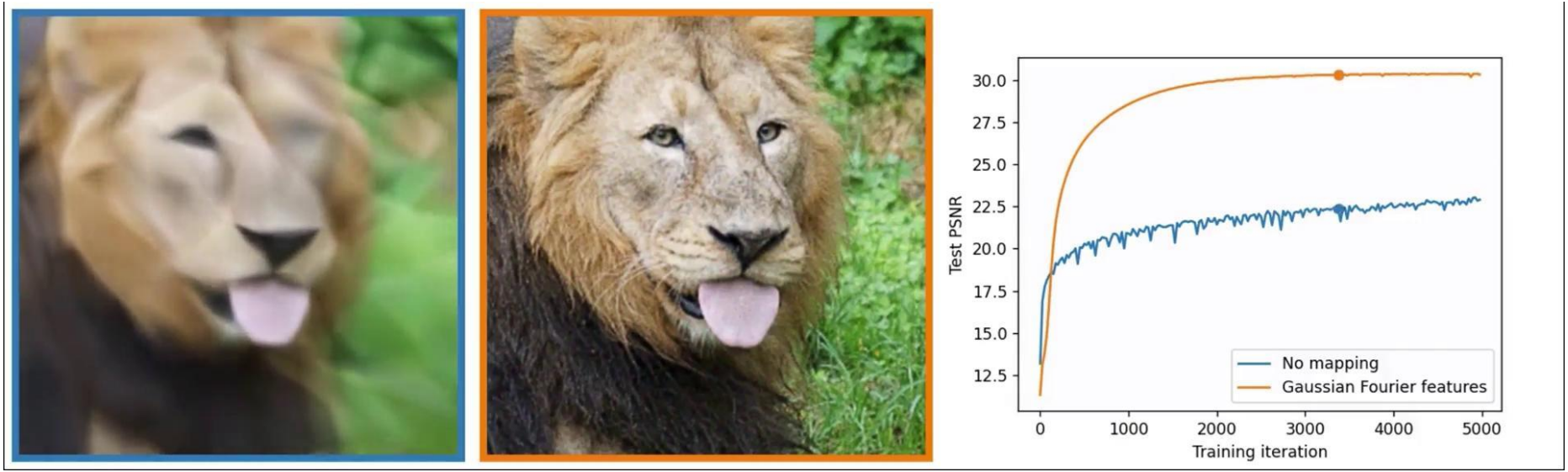
$\mathbf{v} \in \mathbb{R}^d$ - coordinates input
$\gamma(\mathbf{v}) \in \mathbb{R}^{2m}$ - Fourier Features

Results

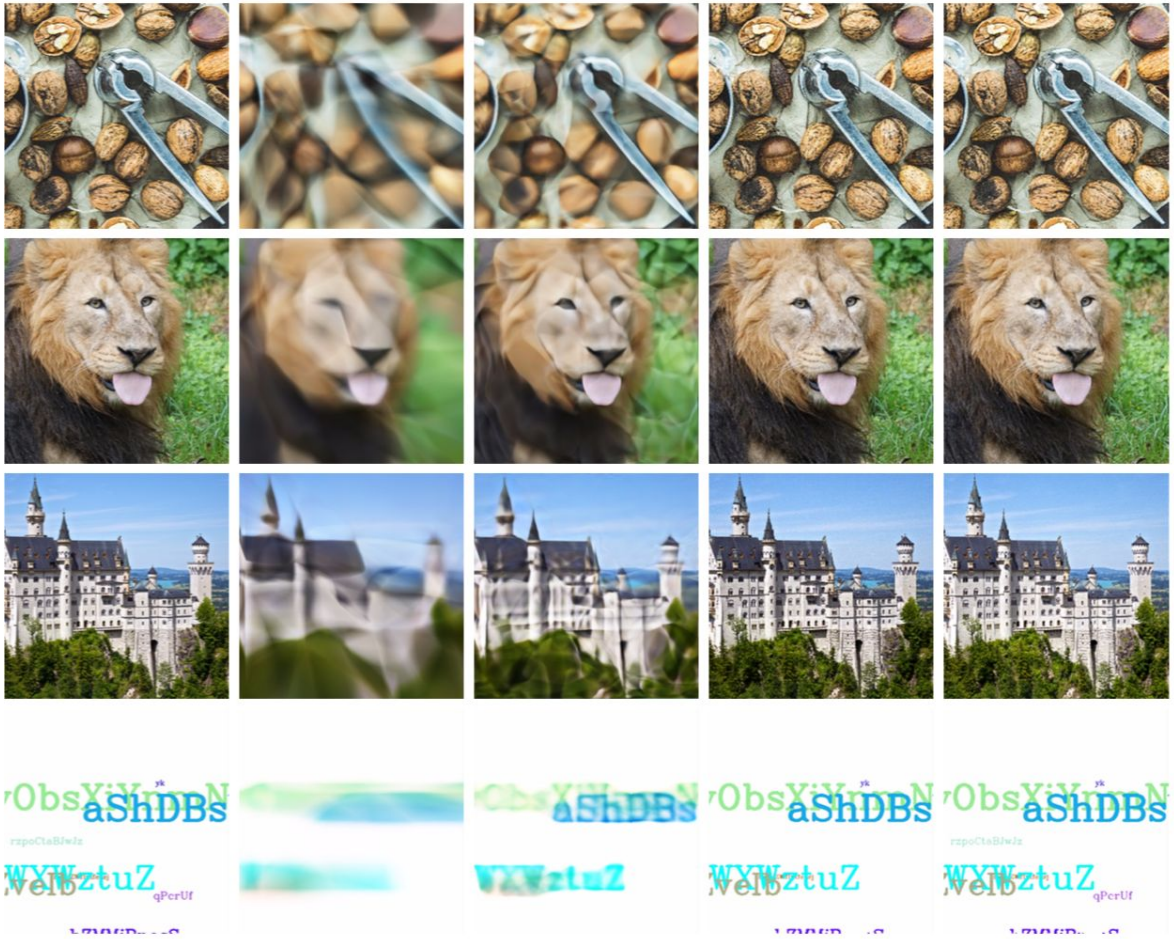
	Direct supervision			Indirect supervision			
	2D image		3D shape [24]	2D CT		3D MRI	3D NeRF [27]
	Natural	Text		Shepp	ATLAS	ATLAS	
No mapping	19.32	18.40	0.864	16.75	15.44	26.14	22.41
Basic	21.71	20.48	0.892	23.31	16.95	28.58	23.16
Positional enc.	24.95	27.57	0.960	26.89	19.55	32.23	25.28
Gaussian	**25.57**	**30.47**	**0.973**	**28.33**	**19.88**	**34.51**	**25.48**

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\mathrm{MAX}_i^2}{\mathrm{MSE}} \right) \qquad \mathrm{IoU} = \frac{|A \cap B|}{|A \cup B|}$$

Results

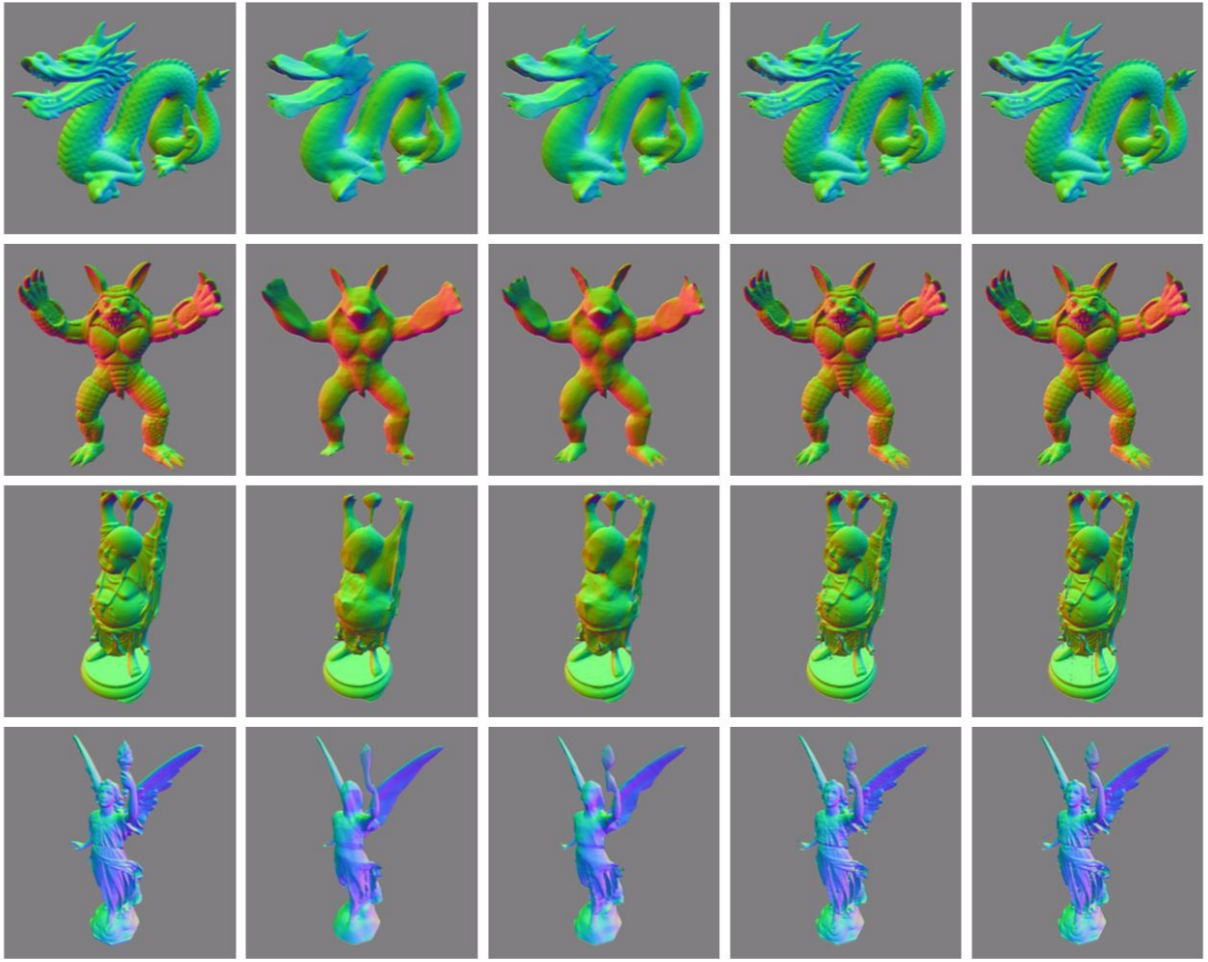


https://people.eecs.berkeley.edu/~bmild/fourfeat/index.html



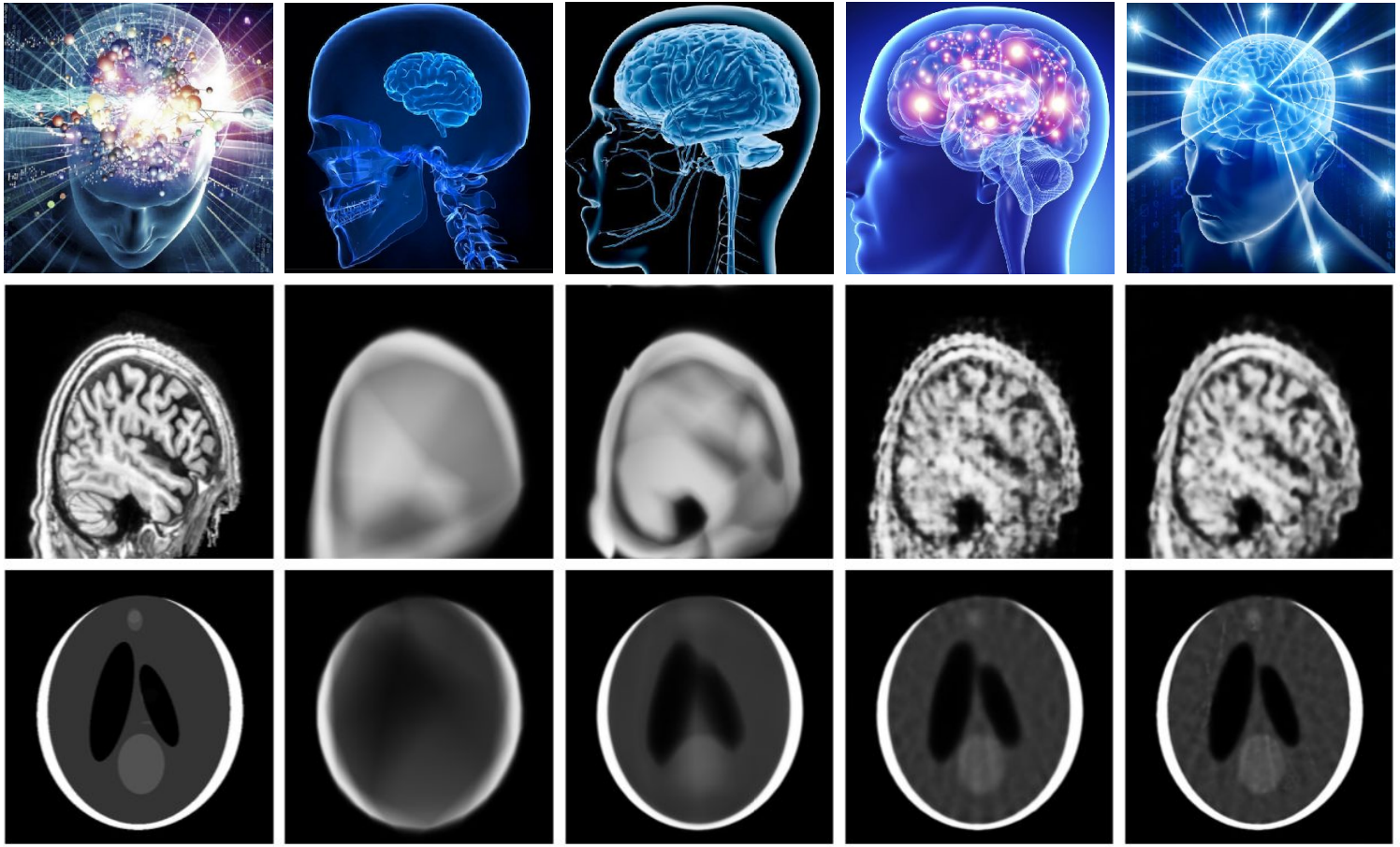
(a) Ground Truth (b) No mapping (c) Basic (d) Positional enc. (e) Gaussian

Figure 15: Additional results for the 2D image regression task, for three images from our *Natural* dataset (top) and two images from our *Text* dataset (bottom).



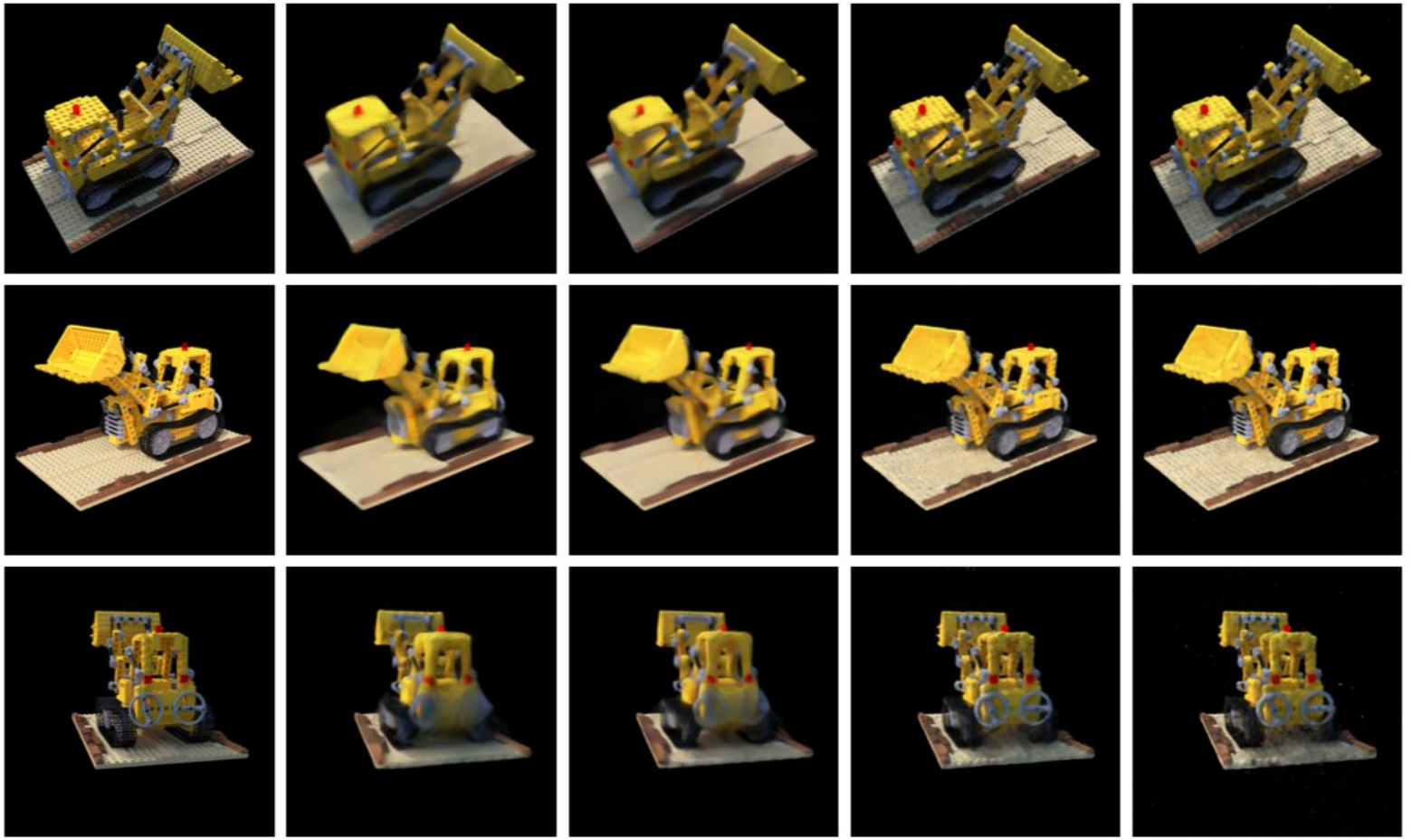
(a) Ground Truth (b) No mapping (c) Basic (d) Positional enc. (e) Gaussian

Figure 16: Additional results for the 3D shape occupancy task [24].



(a) Ground Truth (b) No mapping (c) Basic (d) Positional enc. (e) Gaussian

Figure 17: Results for the 2D CT task.



(a) Ground Truth (b) No mapping (c) Basic (d) Positional enc. (e) Gaussian

Figure 19: Additional results for the inverse rendering task [27].

Summary

1. Coordinate-based MLP плохо выучивают высокочастотные функции.
2. Решение проблемы - использование признаков Фурье для координат.
3. Переобучение можно рассматривать как заучивание моделью высокочастотных компонент функции.
4. Проблема масштабируемости для координат больших размерностей, решение - использование RFF.
5. Дисперсия распределения RFF регулирует недообучение/переобучение.

Questions

1. Опишите задачу регрессии картинки по координатам (вход/выход сети). Какую проблему предлагается решать с помощью признаков Фурье?
2. Напишите приближенную формулу для целевой переменной на тестовой выборке через NTK. Выведите из нее скорость убывания абсолютной ошибки на трэйне через спектр матрицы NTK.
3. Зачем нужны случайные признаки Фурье вместо равномерной сетки частот? Как меняется модель при варьировании дисперсии используемых распределений?

The bibliography

Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains

1. https://arxiv.org/abs/2006.10739
2. https://people.eecs.berkeley.edu/~bmild/fourfeat/index.html

The Convergence Rate of Neural Networks for Learned Functions of Different Frequencies

1. https://arxiv.org/pdf/1906.00425.pdf