Supplementary Material for Trading Positional Complexity vs Deepness in Coordinate Networks

Jianqiao Zheng ⋆, Sameera Ramasinghe⋆, Xueqian Li, and Simon Lucey

Australian Institute for Machine Learning
University of Adelaide
jianqiao.zheng@adelaide.edu.au

A Theoretical results

Proposition 1 *Consider a set of coordinates* $\mathbf{x}{=}[x_1, x_2, \cdots, x_N]^T$*, corresponding outputs* $\mathbf{y}{=}[y_1, y_2, \cdots, y_N]^T$*, and a* d *dimensional embedding* $\Psi{:}\mathbb{R}{\to}\mathbb{R}^d$*. Assuming perfect convergence, the necessary and sufficient condition for a linear model to perfect memorize of the mapping between* $\mathbf{x}$ *and* $\mathbf{y}$ *is for* $\mathbf{X}{=}[\Psi(x_1), \Psi(x_2), \ldots, \Psi(x_N)]$ *to have full rank.*

Proof: Let us refer to the row vectors of $\mathbf{X}$ as $[\mathbf{p}_1, \ldots, \mathbf{p}_d]^T$. In order to perfectly reconstruct $\mathbf{y}$ using a linear learner with weights $\mathbf{w}{=}[w_1, w_2, \ldots, w_d]$ as

$$\mathbf{y} = \sum_{i=1}^{d} w_i \mathbf{p}_i + b \,, \tag{1}$$

one needs $\mathbf{X}$ to be of rank N (since $\mathbf{y}$ needs to completely span $\{\mathbf{p}_i\}_{i=1}^d$). If $d > N$ then there is no unique solution to $\{\mathbf{w}, b\}$ without some regularization. In the unlikely scenario that the row vectors of $\mathbf{X}$ have zero mean, then $\mathbf{X}$ needs to be of rank $N-1$ since the bias term b can account for that missing linear basis. □

Proposition 2 *Let the Gaussian embedder be denoted as* $\psi(t, x){=}\exp\left(-\frac{\|t-x\|^2}{2\sigma^2}\right)$*. With a sufficient embedding dimension, the stable rank of the embedding matrix obtained using the Gaussian embedder is* $\min\left(N, \frac{1}{2\sqrt{\pi}\sigma}\right)$ *where* N *is the number of embedded coordinates. Under the same conditions, the embedded distance between two coordinates* x_1 *and* x_2 *is* $\mathrm{D}(x_1, x_2){=}\exp\left(-\frac{\|x_1-x_2\|^2}{4\sigma^2}\right)$*.*

Proof: Let us define the Gaussian embedder as $\psi(t, x){=}\exp\left(-\frac{\|t-x\|^2}{2\sigma^2}\right)$, where σ is the standard deviation. Given d samples points $[t_1, \ldots, t_d]$ and N input coordinates $[x_1, \ldots, x_N]$, the elements of the embedding matrix are

$$\Psi_{i,j} = \psi(t_i, x_j) \,. \tag{2}$$

⋆ Project page at https://osiriszjq.github.io/complex_encoding

To make sure the stable rank is saturated, we assume that d and N is large enough. Then, Ψ is approximately a circulant matrix. We know that the singular value decomposition of a circulant matrix C, whose first row is c, can be written as

$$C = \frac{1}{n} F_n^{-1} diag\left(F_n c\right) F_n \ , \tag{3}$$

where F_n is the Fourier transform matrix. This means the singular values of a circulant matrix is the Fourier transform of first row. When N is large enough, we can approximate the first row of Ψ as a continuous signal, which is $\psi(x, t{=}0){=}\exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)$, so the singular values are

$$s(\xi) = \mathcal{F}\left(\psi(x; t=0)\right) = \sqrt{2\pi}\sigma \exp\left(-2\sigma^2 \|\pi\xi\|^2\right) \ . \tag{4}$$

Therefore, we can calculate stable rank directly from the definition,

$$\text{Stable Rank}(\Psi){=}\sum_{i=1}^{N} \frac{s_i^2}{s_1^2}{=}\int_{-\infty}^{+\infty} \frac{s^2(\xi)}{s^2(0)} d\xi = \int_{-\infty}^{+\infty} \exp\left(-4\sigma^2\|\pi\xi\|^2\right) d\xi{=}\frac{1}{2\sqrt{\pi}\sigma} \ . \tag{5}$$

Considering the general case, where N might not be large enough, the stable rank will be $\min\left(N, \frac{1}{2\sqrt{\pi}\sigma}\right)$.

The distance (or similarity) between two embedded coordinates can be obtained via the inner product:

$$\begin{aligned}
\mathrm{D}(x_1, x_2) &= \int_{-\infty}^{+\infty} \psi(t, x_1)\psi(t, x_2) dt \\
&= \int_{-\infty}^{+\infty} e^{-\frac{(t-x_1)^2}{2\sigma^2}} e^{-\frac{(t-x_2)^2}{2\sigma^2}} dt \\
&= \int_{-\infty}^{+\infty} e^{-\frac{(t-x_1)^2+(t-x_2)^2}{2\sigma^2}} dt \\
&= \int_{-\infty}^{+\infty} e^{-\frac{t^2-2x_1t+x_1^2+t^2-2x_2t+x_2^2}{2\sigma^2}} dt \\
&= \int_{-\infty}^{+\infty} e^{-\frac{2t^2-2(x_1+x_2)t+\frac{(x_1+x_2)^2}{2}+\frac{(x_1-x_2)^2}{2}}{2\sigma^2}} dt \\
&= \int_{-\infty}^{+\infty} e^{-\frac{(t-\frac{x_1+x_2}{2})^2}{\sigma^2}} e^{-\frac{(x_1-x_2)^2}{4\sigma^2}} dt \\
&= e^{-\frac{(x_1-x_2)^2}{4\sigma^2}} \int_{-\infty}^{+\infty} e^{-\frac{(t-\frac{x_1+x_2}{2})^2}{\sigma^2}} dt \\
&= \sqrt{\pi}\sigma e^{-\frac{(x_1-x_2)^2}{4\sigma^2}} \ .
\end{aligned} \tag{6}$$

which is also a Gaussian with a standard deviation $\sqrt{2}\sigma$. We can empirically define that the distance between two embedded coordinates x_1 and x_2 is preserved if $D(x_1, x_2) \geq 10^{-k}$, for an interval $x_1{-}x_2{\leq}l$, where k is a threshold. In the Gaussian embedder, we can analytically obtain a σ for an arbitrary l using the relationship $\sigma{=}\frac{l}{2\sqrt{k\ln 10}}$. □

Proposition 3 *Let the RFF embedding be denoted as* $\gamma(x){=}[\cos 2\pi \mathbf{b}x, \sin 2\pi \mathbf{b}x]$*, where* $\mathbf{b}$ *are sampled from a Gaussian distribution. When the embedding dimension is large enough, the stable rank of RFF will be* $\min\left(N, \sqrt{2\pi}\sigma\right)$*, where* N *is the numnber of embedded coordinates. Under the same conditions, the embedded distance between two coordinates* x_1 *and* x_2 *is* $\mathrm{D}(x_1, x_2){=}\sum_j \cos 2\pi b_j(x_1{-}x_2)$*.*

Proof: Given $\frac{d}{2}$ samples for $\mathbf{b}$ as $[b_1, \ldots, b_{\frac{d}{2}}]$ from a Gaussian distribution with a standard deviation σ and N input coordinates $[x_1, \ldots, x_N]$, RFF embedding is defined as $\gamma(x){=}[\cos 2\pi \mathbf{b}x_i, \sin 2\pi \mathbf{b}x_i]$.

To make sure the stable rank is saturated, we assume that the d and N is large enough. Although RFF embedding matrix is not circulant, it is naturally frequency based so we already know its spectrum, which is its singular value distribution

$$s(\xi) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\xi^2}{2\sigma^2}\right) . \tag{7}$$

Similarly,

$$\text{Stable Rank}(\gamma) = \sum_{i=1}^{N} \frac{s_i^2}{s_1^2} = \int_{-\infty}^{+\infty} \frac{s^2(\xi)}{s^2(0)} d\xi = \int_{-\infty}^{+\infty} \exp\left(-\frac{\xi^2}{2\sigma^2}\right) d\xi = \sqrt{2\pi}\sigma \ , \tag{8}$$

Considering the general case, the stable rank is $\min\left(N, \sqrt{2\pi}\sigma\right)$.

From the basic trigonometry, it can be easily deduced the distance function that $\mathrm{D}(x_1, x_2){=}\sum_j \cos 2\pi b_j(x_1{-}x_2)$. When d is extremely large it can be considered as $f(\xi){=}\cos 2\pi\xi(x_1{-}x_2)$ where ξ is a Gaussian random variable with standard deviation σ. Then the above sum can be replaced with the integral,

$$\begin{aligned}
\mathrm{D}(x_1, x_2) &= \int_{-\infty}^{+\infty} e^{-\frac{\xi^2}{2\sigma^2}} \cos 2\pi\xi(x_1 - x_2) d\xi \\
&= 2\int_{0}^{+\infty} e^{-\frac{\xi^2}{2\sigma^2}} \cos 2\pi\xi(x_1 - x_2) d\xi \\
&= 2\int_{0}^{+\infty} e^{-\frac{\xi^2}{2\sigma^2}} \frac{1}{2}(e^{i2\pi(x_1-x_2)\xi} + e^{-i2\pi(x_1-x_2)\xi}) d\xi \\
&= \int_{0}^{+\infty} e^{-\frac{\xi^2}{2\sigma^2}+i2\pi(x_1-x_2)\xi} + e^{-\frac{\xi^2}{2\sigma^2}-i2\pi(x_1-x_2)\xi} d\xi \ .
\end{aligned} \tag{9}$$

Further,

$$\int_{0}^{+\infty} e^{-ax^2+bx} dx = e^{-\frac{b^2}{4a}} \int_{0}^{+\infty} e^{-a(x-i\frac{b}{2a})^2} dx = \frac{1}{2}\left(1{+}\mathrm{erfi}\left(\frac{b}{2\sqrt{a}}\right)\right)\sqrt{\frac{\pi}{a}} e^{-\frac{b^2}{4a}} \ . \tag{10}$$

Let $a{=}\frac{1}{2\sigma^2}$ and $b{=}\pm 2\pi(x_1{-}x_2)$. Then, we have

$$\mathrm{D}(x_1, x_2) = \sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2(x_1-x_2)^2} \ . \tag{11}$$

□

Proposition 4 *Let the Rectangular embedder be denoted as* $\psi(t,x)$=*rect* $\left(\frac{x-t}{\sigma}\right)=(1-\frac{|x-t|}{0.5\sigma})>0$. *With a sufficient embedding dimension, the stable rank of the embedding matrix obtained using the Rectangular embedder is* $\min\left(N,\frac{1}{\sigma}\right)$ *where* N *is the number of embedded coordinates. Under the same conditions, the embedded distance between two coordinates* x_1 *and* x_2 *is* $\mathrm{D}\left(x_1,x_2\right)=\sigma$*tri* $\left(\frac{|x_1-x_2|}{\sigma}\right)=\sigma\max(1-\frac{|x_1-x_2|}{\sigma},0)$.

Proof: Let us define the Rectabgular embedder as $\psi(t,x)=\mathrm{rect}\left(\frac{x-t}{\sigma}\right)=\left(1-\frac{|x-t|}{0.5\sigma}\right)>0$, where σ is the width of the rectangle impulse. Given d samples points $[t_1,\ldots,t_d]$ and N input coordinates $[x_1,\ldots,x_N]$, the elements of the embedding matrix are

$$\Psi_{i,j}=\psi(t_i,x_j)\;. \tag{12}$$

To make sure the stable rank is saturated, we assume that d and N are large enough. Then, Ψ is approximately a circulant matrix. We know that the singular value decomposition of a circulant matrix C, whose first row is c, can be written as

$$C=\frac{1}{n}F_n^{-1}diag\left(F_nc\right)F_n\;, \tag{13}$$

where F_n is the Fourier transform matrix. This means the singular values of a circulant matrix are the Fourier transform of the first row. When N is large enough, we can approximate the first row of Ψ as a continuous signal, which is $\psi(x,t{=}0)=\mathrm{rect}(\frac{x}{\sigma})$, so the singular values are

$$s(\xi)=\mathcal{F}\left(\psi(x;t=0)\right)=\sigma\mathrm{sinc}(\sigma\xi)\;, \tag{14}$$

where $\mathrm{sinc}(\xi)=\frac{\sin(\pi x)}{\pi x}$. Therefore, we can compute the stable rank directly from the definition,

$$\text{Stable Rank}(\Psi)=\sum_{i=1}^{N}\frac{s_i^2}{s_1^2}=\int_{-\infty}^{+\infty}\frac{s(\xi)^2}{s(0)^2}d\xi=\int_{-\infty}^{+\infty}\mathrm{sinc}^2(\sigma\xi)d\xi=\frac{1}{\sigma}\;. \tag{15}$$

Considering the general case, where N might not be large enough, the stable rank will be $\min\left(N,\frac{1}{\sigma}\right)$.

The distance (or similarity) between two embedded coordinates can be obtained via the inner product:

$$\begin{aligned}\mathrm{D}(x_1,x_2)&=\int_{-\infty}^{+\infty}\psi(t,x_1)\psi(t,x_2)dt\\&=\int_{-\infty}^{+\infty}\mathrm{rect}\left(\frac{x_1-t}{\sigma}\right)\mathrm{rect}\left(\frac{x_2-t}{\sigma}\right)dt\\&=\sigma\mathrm{tri}\left(\frac{x_1-x_2}{\sigma}\right)\;.\end{aligned} \tag{16}$$

$\square$

Proposition 5 *Let the Triangular embedder be* $\psi(t,x)$=*tri* $\left(\frac{x-t}{0.5\sigma}\right)=\max(1-\frac{|x-t|}{0.5\sigma},0)$. *With a sufficient embedding dimension, the stable rank of the embedding matrix obtained using the Triangular embedder is* $\min(N,\frac{4}{3\sigma})$ *where* N *is the number of embedded coordinates. Under the same conditions, the embedded distance between two coordinates* x_1 *and* x_2 *is* $\mathrm{D}(x_1,x_2)=\frac{1}{4}\sigma^2 \textit{tri}^2(\frac{|x_1-x_2|}{\sigma})=\frac{1}{4}\sigma^2\max(1-\frac{|x_1-x_2|}{\sigma},0)^2$.

Proof: Let us define the Triangle embedder as $\psi(t,x)$=tri $\left(\frac{x-t}{0.5\sigma}\right)=\max\left(1-\frac{|x-t|}{0.5\sigma},0\right)$, where σ is the width of the Triangular impulse. Given d samples points $[t_1,\ldots,t_d]$ and N input coordinates $[x_1,\ldots,x_N]$, the elements of the embedding matrix are

$$\Psi_{i,j}=\psi(t_i,x_j)\;. \tag{17}$$

To make sure the stable rank is saturated, we assume that d and N are large enough. Then, Ψ is approximately a circulant matrix. We know that the singular value decomposition of a circulant matrix C, whose first row is c, can be written as

$$C=\frac{1}{n}F_n^{-1}diag\left(F_n c\right)F_n\;, \tag{18}$$

where F_n is the Fourier transform matrix. This means the singular values of a circulant matrix are the Fourier transform of the first row. When N is large enough, we can approximate the first row of Ψ as a continuous signal, which is $\psi(x,t{=}0)$=tri $\left(\frac{x}{\sigma}\right)$, so the singular values are

$$s(\xi)=\mathcal{F}\left(\psi(x;t=0)\right)=\frac{\sigma}{2}\mathrm{sinc}^2\left(\frac{\sigma}{2}\xi\right)\;, \tag{19}$$

where sinc(ξ)=$\frac{\sin(\pi x)}{\pi x}$. Therefore, we can compute stable rank directly from the definition as,

$$\text{Stable Rank}(\Psi)=\sum_{i=1}^{N}\frac{s_i^2}{s_1^2}=\int_{-\infty}^{+\infty}\frac{s(\xi)}{s(0)}^2 d\xi=\int_{-\infty}^{+\infty}\mathrm{sinc}^4\left(\frac{\sigma}{2}\xi\right)d\xi=\frac{4}{3\sigma}\;. \tag{20}$$

Considering the general case, where N might not be large enough, the stable rank will be $\min\left(N,\frac{1}{\sigma}\right)$.

The distance (or similarity) between two embedded coordinates can be obtained via the inner product:

$$\begin{aligned}\mathrm{D}(x_1,x_2)&=\int_{-\infty}^{+\infty}\psi(t,x_1)\psi(t,x_2)dt\\&=\int_{-\infty}^{+\infty}\mathrm{tri}\left(\frac{x-t}{0.5\sigma}\right)\mathrm{tri}\left(\frac{x-t}{0.5\sigma}\right)dt\\&=\frac{1}{4}\sigma^2\max\left(1-\frac{|x_1-x_2|}{\sigma},0\right)^2\;.\end{aligned} \tag{21}$$

□

B 2D complex encoding

B.1 Closed form solution for separable coordinates

If pixels are sampled on a regular grid formed by samples $\mathbf{x}{=}[x_1, x_2, \cdots, x_N]^T$ and samples $\mathbf{y}{=}[y_1, y_2, \cdots, y_M]^T$, then the coordinates of these pixels are separable. Let $\mathbf{S}{\in}\mathbb{R}^{M\times N}$ be the signal defined as $\mathbf{S}_{i,j}{=}I(x_i, y_j)$, where $i{=}1, 2, \cdots, N$, $j{=}1, 2, \cdots, M$, and $\Psi{:}\mathbb{R}{\to}\mathbb{R}^K$ be the 1D encoder. We want to find the weights $\mathbf{W}{\in}\mathbb{R}^{K\times K}$ of the linear layer by optimizing the following equation,

$$\arg\min_{\mathbf{W}} \|\text{vec}(\mathbf{S}) - (\Psi(\mathbf{y}) \otimes \Psi(\mathbf{x}))\, \text{vec}\,(\mathbf{W})\|_2^2 \ , \tag{22}$$

where $\Psi(\mathbf{x}){\in}\mathbb{R}^{N\times K}$ is the encoding for $\mathbf{x}$, $\Psi(\mathbf{y}){\in}\mathbb{R}^{M\times K}$ is the encoding for $\mathbf{y}$. This is a linear least squares problem. Based on the properties of the Kronecker product, we find the optimal solution $\mathbf{W}^*$ as,

$$\begin{aligned}
\text{vec}\,(\mathbf{W}^*) &= \arg\min_{\mathbf{W}} \|\text{vec}\,(\mathbf{S}) - (\Psi(\mathbf{y}) \otimes \Psi(\mathbf{x}))\, \text{vec}\,(\mathbf{W})\|_2^2 \\
&= \left((\Psi(\mathbf{y}) \otimes \Psi(\mathbf{x}))^T (\Psi(\mathbf{y}) \otimes \Psi(\mathbf{x})) \right)^{-1} (\Psi(\mathbf{y}) \otimes \Psi(\mathbf{x}))^T\, \text{vec}\,(\mathbf{S}) \\
&= \left(\left(\left(\Psi(\mathbf{y})^T \Psi(\mathbf{y})\right)^{-1} \Psi(\mathbf{y}) \right) \otimes \left(\left(\Psi(\mathbf{x})^T \Psi(\mathbf{x})\right)^{-1} \Psi(\mathbf{x}) \right) \right) \text{vec}\,(\mathbf{S}) \\
&= \text{vec}\left(\left(\left(\Psi(\mathbf{x})^T \Psi(\mathbf{x})\right)^{-1} \Psi(\mathbf{x}) \right) \mathbf{S} \left(\left(\Psi(\mathbf{y})^T \Psi(\mathbf{y})\right)^{-1} \Psi(\mathbf{y}) \right)^T \right) \\
&= \text{vec}\left(\left(\Psi(\mathbf{x})^T \Psi(\mathbf{x})\right)^{-1} \Psi(\mathbf{x}) \mathbf{S} \Psi(\mathbf{y})^T \left(\Psi(\mathbf{y})^T \Psi(\mathbf{y})\right)^{-1} \right) \ ,
\end{aligned} \tag{23}$$

which means,

$$\mathbf{W}^* = \left(\Psi(\mathbf{x})^T \Psi(\mathbf{x})\right)^{-1} \Psi(\mathbf{x}) \mathbf{S} \Psi(\mathbf{y})^T \left(\Psi(\mathbf{y})^T \Psi(\mathbf{y})\right)^{-1} \ . \tag{24}$$

B.2 Blending matrix for non-separable coordinates

First, we focus on 1D encoders. Given a 1D encoder $\Psi{:}\mathbb{R}{\to}\mathbb{R}^K$ and two points x_0, x_1, we want to express $\Psi(x){\approx}\alpha_0\Psi(x_0){+}\alpha_1\Psi(x_1)$ for $x_0{\le}x{\le}x_1$. This problem can be solved by

$$\arg\min_{\alpha} \left\|\Psi(x) - \left[\Psi(x_0)\ \Psi(x_1)\right] \alpha\right\|_2^2 \ , \tag{25}$$

where $\alpha{=}\left[\alpha_0\ \alpha_1\right]^T$. Note here that $\Psi(x)$, $\Psi(x_0)$, and $\Psi(x_1)$ are $K{\times}1$ vectors. This is equivalent to a least squared problem, thus, the optimal solution α^* can be solved by,

$$\begin{aligned}
\alpha^* &= \arg\min_{\alpha} \left\|\Psi(x) - \left[\Psi(x_0)\ \Psi(x_1)\right] \alpha\right\|_2^2 \\
&= \left(\left[\Psi(x_0)\ \Psi(x_1)\right]^T \left[\Psi(x_0)\ \Psi(x_1)\right] \right)^{-1} \left[\Psi(x_0)\ \Psi(x_1)\right]^T \Psi(x) \\
&= \left(\begin{bmatrix} \Psi(x_0)^T \\ \Psi(x_1)^T \end{bmatrix} \left[\Psi(x_0)\ \Psi(x_1)\right] \right)^{-1} \begin{bmatrix} \Psi(x_0)^T \\ \Psi(x_1)^T \end{bmatrix} \Psi(x) \\
&= \begin{bmatrix} \Psi(x_0)^T\Psi(x_0) & \Psi(x_0)^T\Psi(x_1) \\ \Psi(x_1)^T\Psi(x_0) & \Psi(x_1)^T\Psi(x_1) \end{bmatrix}^{-1} \begin{bmatrix} \Psi(x_0)^T\Psi(x) \\ \Psi(x_1)^T\Psi(x) \end{bmatrix} \ .
\end{aligned} \tag{26}$$

With the definition $D(x_1, x_2)$ in Appendix A, this can be written as,

$$\alpha^* = \begin{bmatrix} D(x_0, x_0) & D(x_0, x_1) \\ D(x_1, x_0) & D(x_1, x_1) \end{bmatrix}^{-1} \begin{bmatrix} D(x_0, x) \\ D(x_1, x) \end{bmatrix} . \tag{27}$$

Typically, this distance function only depends on the difference of the inputs, as examples shown in Appendix A. Therefore, we can have a close form solution for D:$\mathbb{R}\to\mathbb{R}$. Let $d{=}x_1{-}x_0$, and $x{=}x_0{+}\beta d$, where $0{\le}\beta{\le}1$. Then, the solution becomes,

$$\begin{aligned} \alpha^* &= \begin{bmatrix} D(x_0, x_0) & D(x_0, x_1) \\ D(x_1, x_0) & D(x_1, x_1) \end{bmatrix}^{-1} \begin{bmatrix} D(x_0, x) \\ D(x_1, x) \end{bmatrix} \\ &= \begin{bmatrix} D(0) & D(d) \\ D(d) & D(0) \end{bmatrix}^{-1} \begin{bmatrix} D(\beta d) \\ D\left((1-\beta)\, d\right) \end{bmatrix} \\ &= \frac{1}{D^2(0) - D^2(d)} \begin{bmatrix} D(0) & -D(d) \\ -D(d) & D(0) \end{bmatrix} \begin{bmatrix} D(\beta d) \\ D\left((1-\beta)\, d\right) \end{bmatrix} . \end{aligned} \tag{28}$$

Based on the 1D analysis, encoding 2D non-separable points can also be expressed as non-linear interpolation of 2D separable coordinates. Suppose that the settings are the same as in Appendix B.1. The virtual pixels are sampled on a regular grid formed by samples $\mathbf{x}{=}[x_1, x_2, \cdots, x_N]^T$ and samples $\mathbf{y}{=}[y_1, y_2, \cdots, y_M]^T$. The query points are randomly sampled in the space as $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_P]^T$, where P is the number of points and each $\mathbf{q}_i{\in}\mathbb{R}^{2\times 1}$ is a random 2D coordinate. Let $\mathbf{s}{\in}\mathbb{R}^{P\times 1}$ be the signal, and Ψ:$\mathbb{R}\to\mathbb{R}^K$ be the 1D encoder. We want to find the weights $\mathbf{W}{\in}\mathbb{R}^{K\times K}$ of the linear layer by optimizing the following equation,

$$\arg\min_{\mathbf{W}} \left\| \mathbf{s} {-} B(\mathbf{Q}) \left(\Psi(\mathbf{y}) \otimes \Psi(\mathbf{x})\right) \operatorname{vec}\left(\mathbf{W}\right) \right\|_2^2 , \tag{29}$$

where B:$\mathbb{R}^2\to\mathbb{R}^{MN}$ is the non-linear interpolation coefficients function, *i.e.*, $B(\mathbf{Q}) \in \mathbb{R}^{P\times MN}$ is the blending matrix. Note that although B is large, it is extremely sparse and only have 4 non-zero values on each row of MN elements. Consider a certain point $\mathbf{q}_p$ is in the grid whose corner points are (x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}), and (x_{i+1}, y_{j+1}), which means $\mathbf{x}_i{\le}\mathbf{q}_{p0}{\le}\mathbf{x}_{i+1}$ and $\mathbf{y}_j{\le}\mathbf{q}_{p1}{\le}\mathbf{y}_{j+1}$. Then we can obtain the encoding for $\mathbf{q}_{p0}$ and $\mathbf{q}_{p1}$ as follows,

$$\begin{aligned} \Psi(\mathbf{q}_{p0}) &\approx \alpha_0 \Psi(x_i) + \alpha_1 \Psi(x_{i+1}), \\ \Psi(\mathbf{q}_{p1}) &\approx \beta_0 \Psi(y_j) + \beta_1 \Psi(y_{j+1}) . \end{aligned} \tag{30}$$

Then, the 2D encoding for $\mathbf{q}_p$ is,

$$\begin{aligned} \Psi(\mathbf{q}) =& \Psi(\mathbf{q}_{p0}, \mathbf{q}_{p1}) \\ =& \Psi(\mathbf{q}_{p1}) \otimes \Psi(\mathbf{q}_{p0}) \\ \approx& \left(\beta_0 \Psi\left(y_j\right) + \beta_1 \Psi\left(y_{j+1}\right)\right) \otimes \left(\alpha_0 \Psi\left(x_i\right) + \alpha_1 \Psi\left(x_{i+1}\right)\right) \\ =& \alpha_0\beta_0 \Psi\left(y_j\right) \otimes \Psi\left(x_i\right) + \alpha_0\beta_1 \Psi\left(y_{j+1}\right) \otimes \Psi\left(x_i\right) \\ &+ \alpha_1\beta_0 \Psi\left(y_j\right) \otimes \Psi\left(x_{i+1}\right) + \alpha_1\beta_1 \Psi\left(y_{i+1}\right) \otimes \Psi\left(x_{i+1}\right) \\ =& \alpha_0\beta_0 \Psi\left(x_i, y_j\right) + \alpha_0\beta_1 \Psi\left(x_i, y_{j+1}\right) \\ &+ \alpha_1\beta_0 \Psi\left(x_{i+1}, y_j\right) + \alpha_1\beta_1 \Psi\left(x_{i+1}, y_{i+1}\right) , \end{aligned} \tag{31}$$

which means $B(\mathbf{q}_p){\in}\mathbb{R}^{1\times MN}$ are all zeros except $\alpha_0\beta_0$ at index $jN{+}i$, $\alpha_0\beta_1$ at index $(j{+}1)N{+}i$, $\alpha_0\beta_0$ at index $jN{+}i{+}1$ and $\alpha_0\beta_0$ at index $(j{+}1)N{+}i{+}1$.

C HD complexity

Let $\mathbf{X}{\in}\mathbb{R}^{N^D\times D}$ be N^D points in D dimensional space, $\Psi{:}\mathbb{R}{\to}\mathbb{R}^K$ be the 1D encoder, and we want to know the memory and computational complexity when the encoding multiply a linear layer $\mathbf{W}$.

Simple encoding. The embedding $\Psi(\mathbf{X}){\in}\mathbb{R}^{N^D\times DK}$ and the weights $\mathbf{W}{\in}\mathbb{R}^{DK\times 1}$, so the memory complexity is $O(DKN^D)$ and the computational complexity is $O(DKN^D)$.

Complex encoding (naive implementation). The embedding $\Psi(\mathbf{X}){\in}\mathbb{R}^{N^D\times K^D}$ and the weights $\mathbf{W}{\in}\mathbb{R}^{K^D\times 1}$, so the memory complexity is $O(K^DN^D)$ and the computational complexity is $O(K^DN^D)$.

Complex encoding (separable coordinates). The embedding $\Psi(\mathbf{X}){\in}\mathbb{R}^{N\times K}$ and the weights $\mathbf{W}{\in}\mathbb{R}^{K^D}$, so the memory complexity is $O(K^D{+}NK)$ and the computational complexity is $\sum_{i=1}^{D} N^iK^{D+1-i}{=}O(NK\frac{N^D-K^D}{N-K})$. A special case of $N{=}K$ will be discussed later.

Complex encoding (non-separable coordinates). The embedding $\Psi(\mathbf{X}){\in}\mathbb{R}^{N\times K}$, the weights $\mathbf{W}{\in}\mathbb{R}^{K^D}$ and the Blending matrix $B(\mathbf{X}){\in}\mathbb{R}^{N^D\times N^D}$(sparse matrix with only $N^D{\times}2^D$ non-zeros values), so the memory complexity is $O(K^D{+}NK{+}2^DN^D)$, the computational complexity is $2^DN^D{+}\sum_{i=1}^{D} N^iK^{D+1-i}{=}O(2^DN^D{+}NK\frac{N^D-K^D}{N-K})$.

Special case $N{=}K$. Both simple encoding and separable complex encoding have $O(DN^{D+1})$ computational encoding. Memory complexity is $O(DN^{D+1})$ for simple encoding while it is $O(N^D{+}2N)$ for separable encoding. However, the rank of the latter one is power of D to the first one.

D Experiments

D.1 Method Notations

For 1D encoding experiments, we used Fourier-feature-based encodings with linearly, log-linearly, or randomly sampled frequencies, and shifted encodings whose bases are Gaussian or triangle. We give a brief introduction to these methods below.

LinF (Fourier feature-based encoding using linearly sampled frequency).

$$\phi(x) = \left[\cdots, \cos\left(2\pi\cdot\left(\tfrac{K-i}{K}2^0 + \tfrac{i}{K}2^\sigma\right)x\right), \sin\left(2\pi\cdot\left(\tfrac{K-i}{K}2^0 + \tfrac{i}{K}2^\sigma\right)x\right), \cdots\right]^T, \quad (32)$$

where $i{=}0,\ldots,K{-}1$ and σ is the hyperparameter for the frequency range that sampled linearly from base frequency (2^0) to max frequency (2^σ).

LogF (Fourier feature-based encoding using log-linearly sampled frequency).

$$\phi(x) = \left[\cdots, \cos\left(2\pi\cdot 2^{\sigma i/K}x\right), \sin\left(2\pi\cdot 2^{\sigma i/K}x\right), \cdots\right]^T, \quad (33)$$

where $i{=}0,\ldots,K{-}1$ and σ is the hyperparameter for frequency range. The frequency are sampled log-linearly from base frequency (2^0) to max frequency (2^σ).

RFF (Fourier feature-based encoding using randomly sampled frequency) [2].

$$\phi(x) = \left[\cos\left(2\pi \mathbf{b}x\right)^T, \sin\left(2\pi \mathbf{b}x\right)^T\right]^T, \tag{34}$$

where $\mathbf{b} \in \mathbb{R}^{K\times 1}$ is random frequencies sampled from $\mathcal{N}(0, \sigma^2)$, where σ is the hyperparameter for frequency range.

Tri (shifted triangle encoding).

$$\phi(x) = \left[\cdots, \max\left(1 - \left|\frac{x - i/K}{d}\right|, 0\right), \cdots\right]^T, \tag{35}$$

where i=$0, \ldots, K$$-$$1$ and d is the hyperparameter for the width of triangle wave.

Gau (shifted Gaussian encoding).

$$\phi(x) = \left[\cdots, e^{-\frac{x - i/K}{2d^2}}, \cdots\right]^T, \tag{36}$$

where i=$0, \ldots, K$$-$$1$ and d is the hyperparameter for the width of Gaussian wave.

D.2 Non-separable 3D video reconstruction

We used the same Youtube video dataset [1] as described in the main paper. The only difference is that the training points were *randomly* sampled (12.5% from the total number of points) of a $64\times64\times64$ grid, and the rest of the points were used for testing. The results are shown in Table 1. Similar to our observations in the main paper, complex encodings combined with a single linear layer have comparable performance to simple encodings combined with deep (4 layer MLPs) networks while being 10x faster. Complex frequency-based encodings (LinF, LogF, RFF) have inferior results than complex shifted-based encodings (Tri, Gau) due to deficient rank.

Table 1: Performance of video reconstruction with randomly sampled inputs (non-separable coordinates). ● are simple positional encodings. ● are complex positional encodings with stochastic gradient descent using smart indexing. Complex encodings with a single linear network are 10x faster than simple encodings with deep networks.

	PSNR	No. of params (memory)	Time (s)
● LinF	21.38 ± 3.32	$1,445,891$ $(5.78M)$	76.87
● LogF	$\mathbf{21.54 \pm 3.32}$	$1,445,891$ $(5.78M)$	76.76
● RFF [2]	21.35 ± 3.32	$1,445,891$ $(5.78M)$	76.22
● Tri	20.90 ± 3.09	$1,445,891$ $(5.78M)$	77.82
● Gau	21.16 ± 3.11	$1,445,891$ $(5.78M)$	77.98
● LinF	10.08 ± 3.63	$786,432$ $(3.15M)$	55.34
● LogF	18.79 ± 2.55	$786,432$ $(3.15M)$	53.48
● RFF [2]	20.26 ± 2.82	$786,432$ $(3.15M)$	1.82
● Tri	$\mathbf{21.54 \pm 3.01}$	$786,432$ $(3.15M)$	1.83
● Gau	21.29 ± 3.04	$786,432$ $(3.15M)$	1.86

D.3 Visual results for 2D images

Here we show 2D image visual results for separable coordinates in Figs. 2 and 3, and non-separable coordinates in Figs. 4 and 5. For simple encoding, five aforementioned encoders were tested with 256 width MLP of 0 and 4 hidden ReLU layers (0 means only a linear layer). For complex encoding, the same five encoders were tested with 0 and 1 hidden ReLU MLPs.

As shown in column 1 of these figures, when we used simple encodings and the network only had a single linear layer (0 hidden layers), the reconstructed images are of low quality, showing low-resolution color grids (LinF, LogF), cross strip colors (Tri, Gau), or random color blobs (RFF). The results clearly support our claim that a linear network can only reconstruct a 2D image signal with at most rank 2. When we introduced non-linear layers and increased the hidden layer depth (depth 4, column 2), the reconstruction quality improves, leading to a better PSNR.

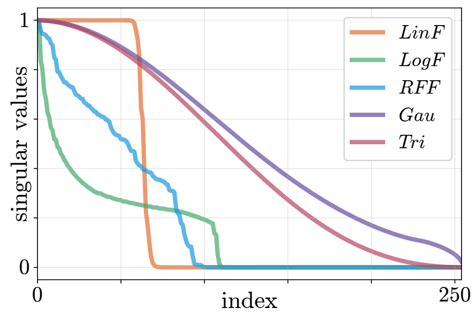

Fig. 1: The normalized singular values of different 1D embeddings $\Psi(\mathbf{x}) \in \mathbb{R}^{N\times K}$. Here N=K=256 and $\mathbf{x}$ is sampled equally spaced from 0 to 1. Fourier feature-based encodings (LinF, LogF, RFF) tend to have much fewerr non-zero singular values, which results in low rank. While shifted encodings (Tri, Gau) usually have sufficient non-zero singular values, which leads to a high rank. When $\mathbf{x}$ is randomly sampled, the rank deficiency in Fourier feature-based encodings becomes worse.

On the contrary, even with a single linear layer (depth 0, column 3), our complex encoding methods can achieve comparable results with methods that used a simple encoding combined with deeper non-linear networks. Note that Fourier feature-based (frequency-based) complex encodings (LinF, LogF, RFF) performed worse than shifted-based complex encodings (Tri, Gau) when there was only one single linear layer due to the deficiency of the embedding rank (shown in Fig. 1). Adding an extra non-linear layer (depth 1, column 4) did not substantially improve the performance of shifted-based complex encodings while adding more details for frequency-based complex encodings.

References

1. Real, E., Shlens, J., Mazzocchi, S., Pan, X., Vanhoucke, V.: Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5296–5305 (2017) 9
2. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. arXiv preprint arXiv:2006.10739 (2020) 9

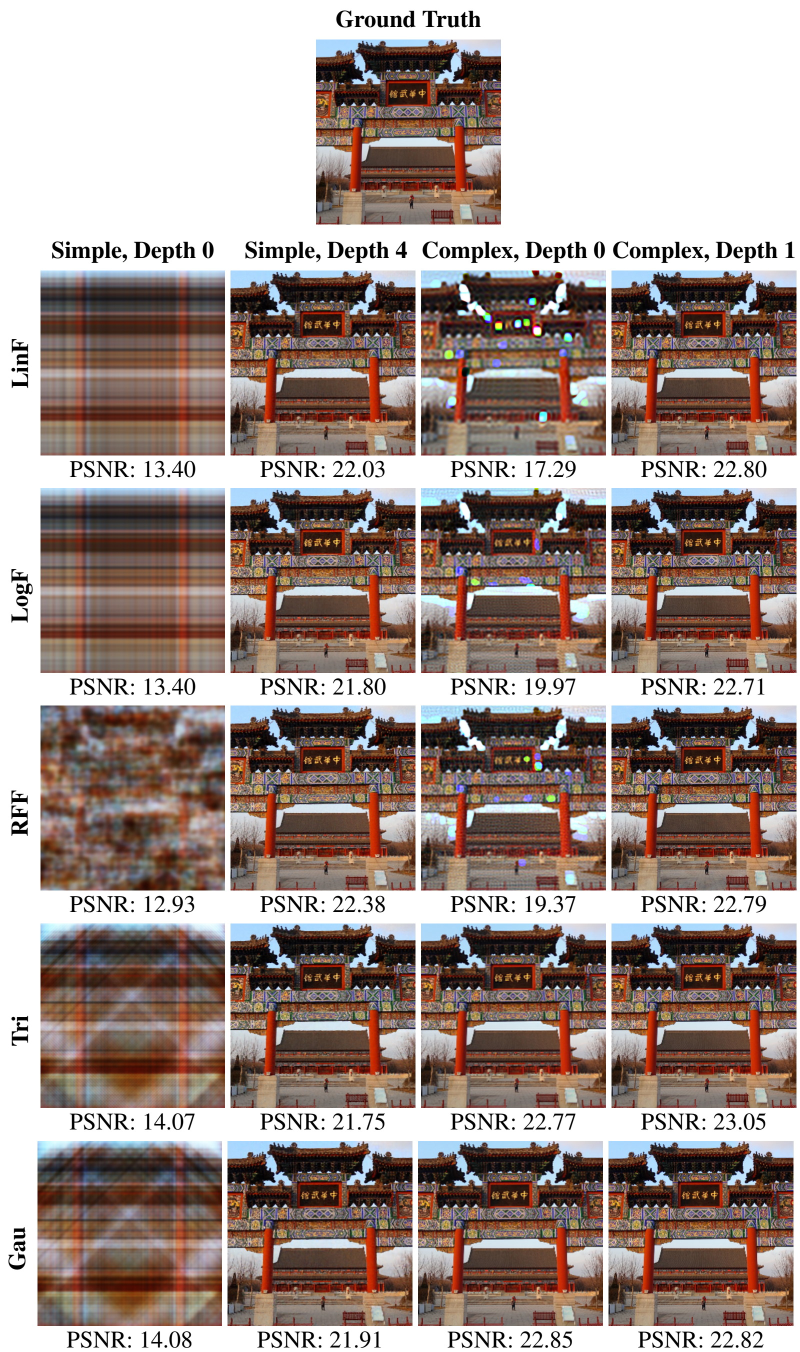

Fig. 2: Reconstruction results of an archway using separable coordinates (regular-grid sampled training points) with different combinations of simple or complex encodings and network depths.

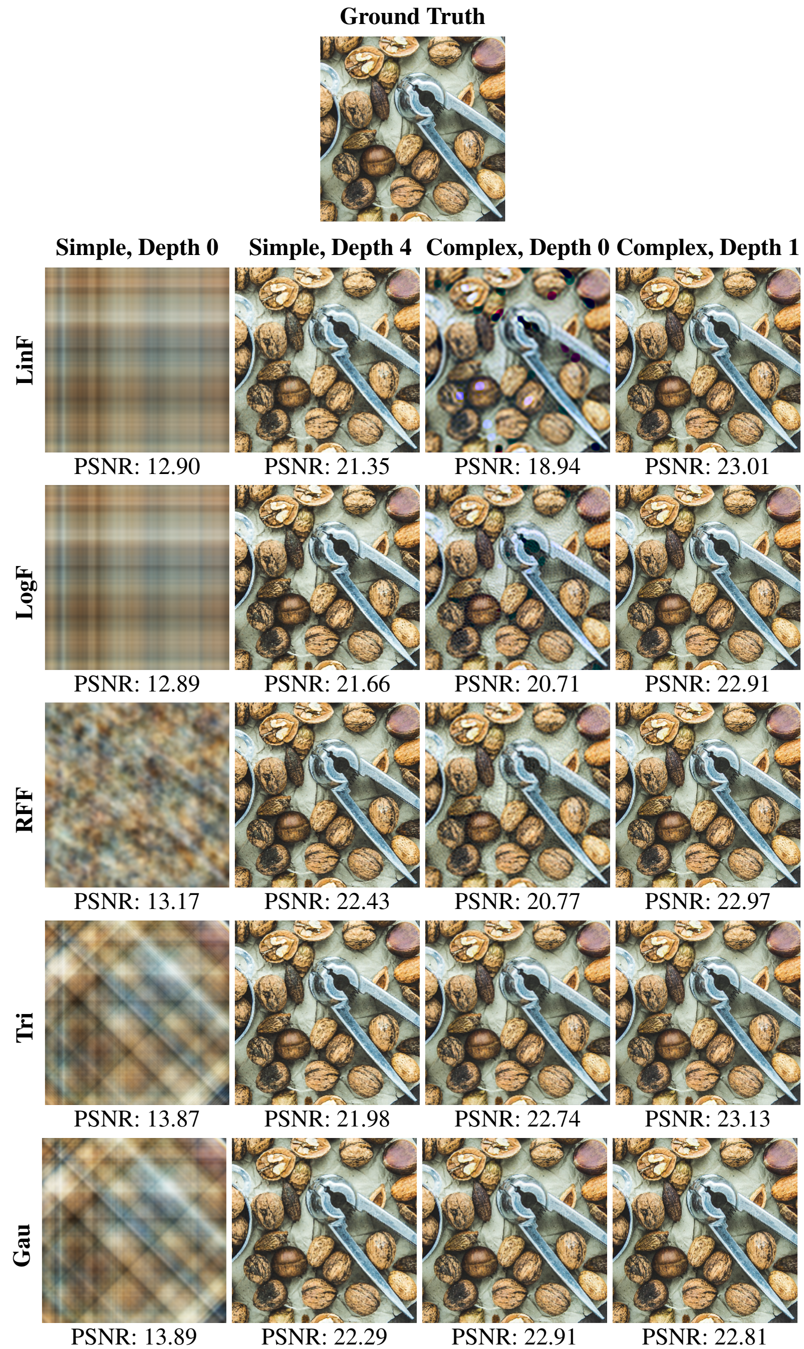

Fig. 3: Reconstruction results of a heap of walnuts using separable coordinates (regular-grid sampled training points) with different combinations of simple or complex encodings and network depths.

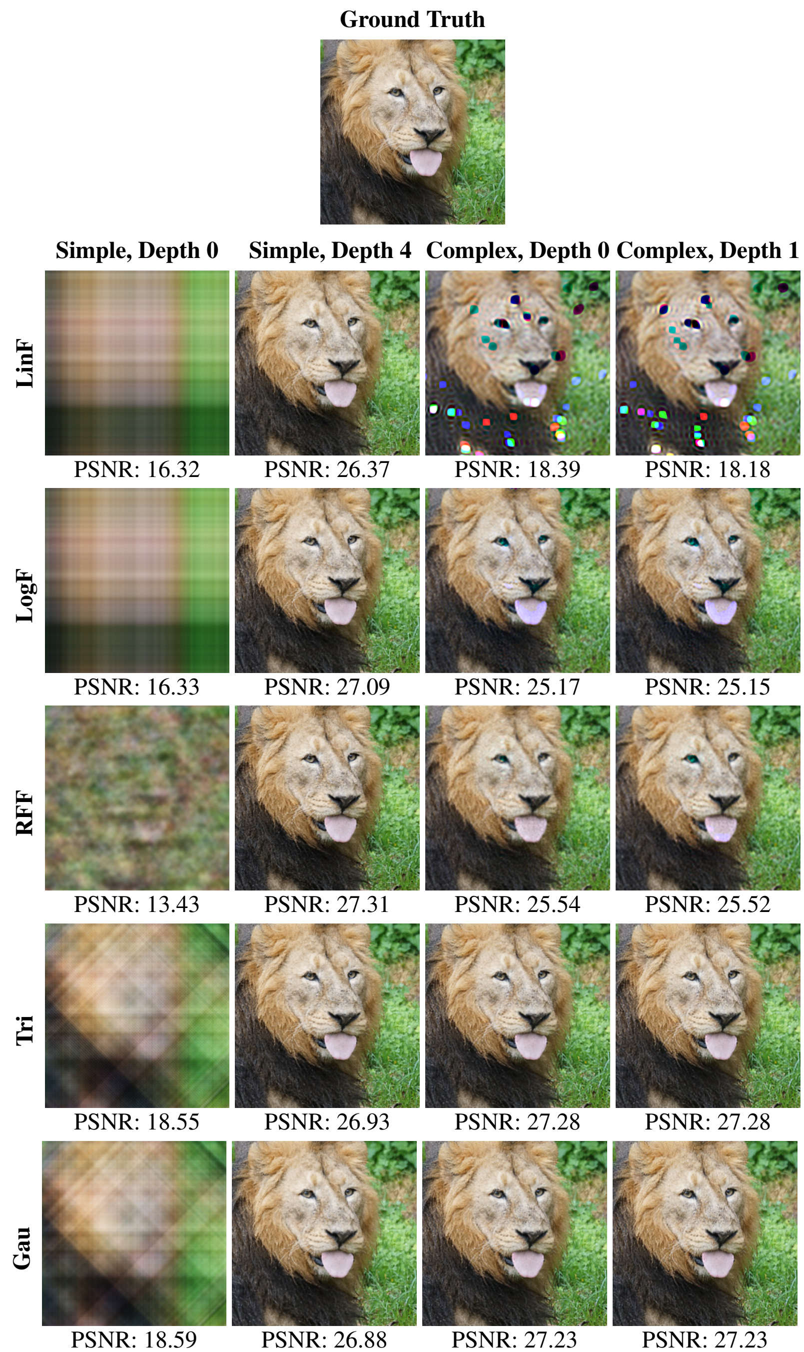

Fig. 4: Reconstruction results of a lion using non-separable coordinates (randomly sampled training points) with different combinations of simple or complex encodings and network depths.

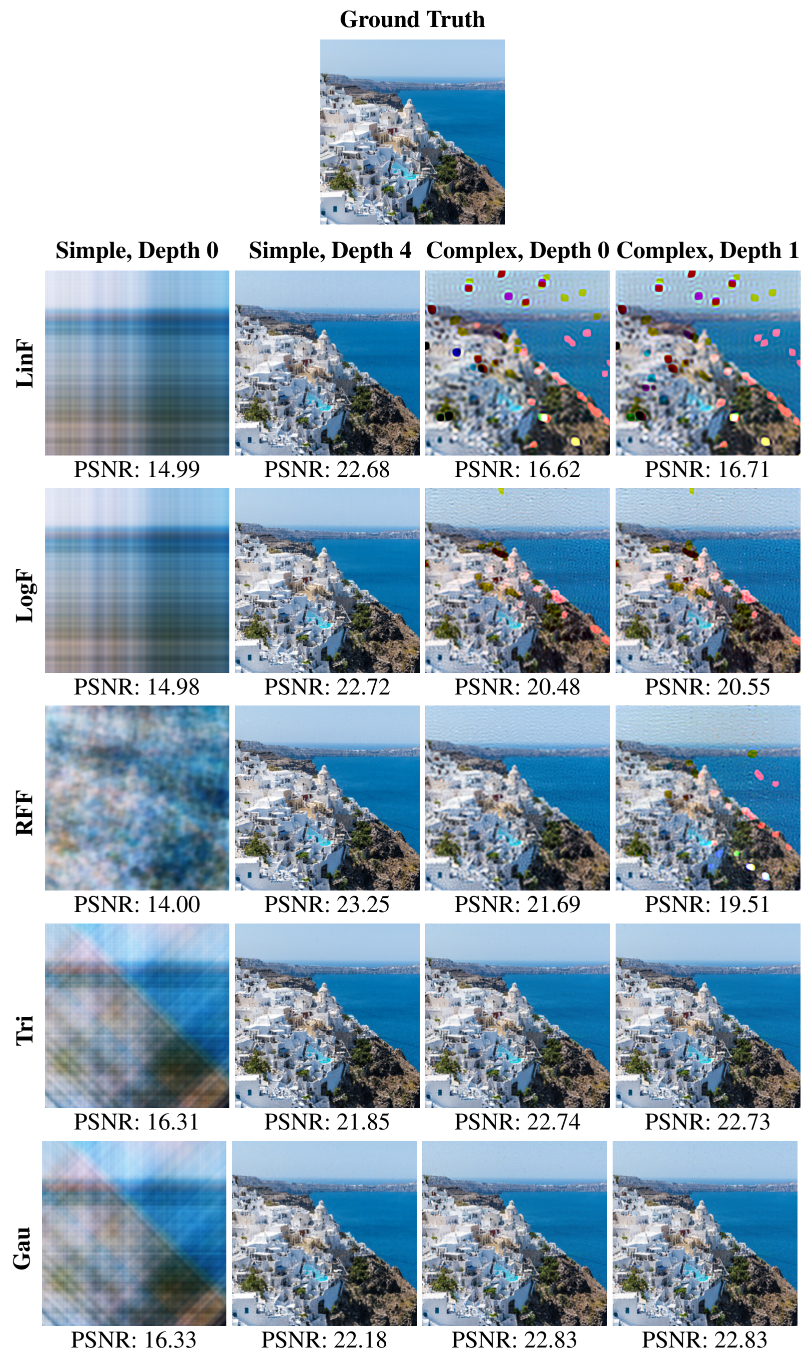

Fig. 5: Reconstruction results of a seaside residential area using non-separable coordinates (randomly sampled training points) with different combinations of simple or complex encodings and network depths.