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 \( x = [x_1, x_2, \cdots, x_N]^T \), corresponding outputs \( y = [y_1, y_2, \cdots, y_N]^T \), 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 \( x \) and \( y \) is for \( X = [\Psi(x_1), \Psi(x_2), \ldots, \Psi(x_N)] \) to have full rank.

**Proof:** Let us refer to the row vectors of \( X \) as \( [p_1, \ldots, p_d]^T \). In order to perfectly reconstruct \( y \) using a linear learner with weights \( w = [w_1, w_2, \ldots, w_d] \) as

\[
y = \sum_{i=1}^{d} w_i p_i + b,
\]

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

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 \( D(x_1, x_2) = \exp\left(-\frac{\|x_1-x_2\|^2}{4\sigma^2}\right) \).

**Proof:** Let us define the Gaussian embedder as \( \psi(t, x) = \exp\left(-\frac{\|t-x\|^2}{2\sigma^2}\right) \), where \( \sigma \) 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

\[
\Psi_{i,j} = \psi(t_i, x_j).
\]  

* 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, $\Psi$ 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

$$C = \frac{1}{n} F_n^{-1} \text{diag} (F_n c) F_n,$$

(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 $\Psi$ as a continuous signal, which is $\psi(x, t=0) = \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)$, so

the singular values are

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

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

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

(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:

$$D(x_1, x_2) = \int_{-\infty}^{+\infty} \psi(t, x_1) \psi(t, x_2) dt$$

\begin{align*}
= \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^2 - 2tx_1 + x_1^2 + t^2 - 2tx_2 + x_2^2}{2\sigma^2}} dt \\
= \int_{-\infty}^{+\infty} e^{-\frac{(t-x_1+x_2)^2 + (x_1-x_2)^2}{2\sigma^2}} dt \\
= \int_{-\infty}^{+\infty} e^{-\frac{x_1^2 - 2x_1 + x_2^2 + x_1^2 - 2x_1 + x_2^2}{2\sigma^2}} dt \\
= \int_{-\infty}^{+\infty} e^{-\frac{(x_1-x_2)^2}{4\sigma^2}} \int_{-\infty}^{+\infty} e^{-\frac{(t-x_1+x_2)^2}{2\sigma^2}} dt \\
= \sqrt{\frac{\pi}{2\sigma}} e^{-\frac{(x_1-x_2)^2}{4\sigma^2}}.
\end{align*}

(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 $\sigma$ for an arbitrary $l$ using the relationship $\sigma = \frac{l}{2\sqrt{\ln 10}}$. \qed
Proposition 3 Let the RFF embedding be denoted as \( \gamma(x) = [\cos 2\pi b_1 x, \sin 2\pi b_2 x] \), where \( b \) are sampled from a Gaussian distribution. When the embedding dimension is large enough, the stable rank of RFF will be \( \min (N, \sqrt{2\pi \sigma}) \), where \( N \) is the number of embedded coordinates. Under the same conditions, the embedded distance between two coordinates \( x_1 \) and \( x_2 \) is \( D(x_1, x_2) = \sum_j \cos 2\pi b_j (x_1 - x_2) \).

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

To make sure the stable rank is saturated, we assume that the \( d \) and \( N \) are 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

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

Similarly,

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

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

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

\[
D(x_1, x_2) = \int_{-\infty}^{+\infty} e^{-\frac{x^2}{2\sigma^2}} \cos 2\pi \xi (x_1 - x_2) d\xi \\
= 2 \int_{0}^{+\infty} e^{-\frac{x^2}{2\sigma^2}} \cos 2\pi \xi (x_1 - x_2) d\xi \\
= 2 \int_{0}^{+\infty} e^{-\frac{x^2}{2\sigma^2}} \frac{1}{2} (e^{2\pi (x_1 - x_2) \xi} + e^{-i2\pi (x_1 - x_2) \xi}) d\xi \\
= \int_{0}^{+\infty} e^{-\frac{x^2}{2\sigma^2} + i2\pi (x_1 - x_2) \xi} + e^{-\frac{x^2}{2\sigma^2} - i2\pi (x_1 - x_2) \xi} d\xi 
\]

Further,

\[
\int_{0}^{\infty} e^{-ax^2 + bx} dx = e^{-\frac{b^2}{4a}} \int_{0}^{\infty} e^{-a(x - \frac{b}{2a})^2} dx = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{b}{2\sqrt{a}} \right) \right) \sqrt{\frac{\pi}{a}} e^{-\frac{b^2}{4a}}.
\]

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

\[
D(x_1, x_2) = \sqrt{2\pi \sigma} e^{-2\pi^2 \sigma^2 (x_1 - x_2)^2}.
\]
Proposition 4 Let the Rectangular embedder be denoted as \( \psi(t, x) = \text{rect} \left( \frac{x - t}{\sigma} \right) = (1 - \frac{|x - t|}{0.5\sigma}) > 0 \). 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 \( D(x_1, x_2) = \sigma \text{tri} \left( \frac{|x_1 - x_2|}{\sigma} \right) = \sigma \max(1 - \frac{|x_1 - x_2|}{\sigma}, 0) \).

Proof: Let us define the Rectangular embedder as \( \psi(t, x) = \text{rect} \left( \frac{x - t}{\sigma} \right) = (1 - \frac{|x - t|}{0.5\sigma}) > 0 \), where \( \sigma \) 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

\[
\Psi_{i,j} = \psi(t_i, x_j).
\] (12)

To make sure the stable rank is saturated, we assume that \( d \) and \( N \) are large enough. Then, \( \Psi \) 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

\[
C = \frac{1}{n} F_n^{-1} \text{diag} \left( F_n c \right) F_n,
\] (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 \( \Psi \) as a continuous signal, which is \( \psi(x, t=0) = \text{rect}(\frac{x}{\sigma}) \), so the singular values are

\[
s(\xi) = F(\psi(x; t=0)) = \sigma \text{sinc}(\sigma \xi),
\] (14)

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

\[
\text{Stable Rank}(\Psi) = \sum_{i=1}^{N} \frac{s_i^2}{s_i^2} = \int_{-\infty}^{+\infty} s(\xi)^2 d\xi = \int_{-\infty}^{+\infty} \text{sinc}^2(\sigma \xi) d\xi = \frac{1}{\sigma}.
\] (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:

\[
D(x_1, x_2) = \int_{-\infty}^{+\infty} \psi(t, x_1)\psi(t, x_2) dt
= \int_{-\infty}^{+\infty} \text{rect} \left( \frac{x_1 - t}{\sigma} \right) \text{rect} \left( \frac{x_2 - t}{\sigma} \right) dt
= \sigma \text{tri} \left( \frac{x_1 - x_2}{\sigma} \right).
\] (16)
Proposition 5 Let the Triangular embedder be $\psi(t, x) = \text{tri} \left( \frac{x - t}{0.5\sigma} \right) = \max(1 - \frac{|x - t|}{0.5\sigma}, 0)$.
With a sufficient embedding dimension, the stable rank of the embedding matrix obtained using the Triangular embedder is $\min(N, \frac{1}{\sigma})$ where $N$ is the number of embedded coordinates. Under the same conditions, the embedded distance between two coordinates $x_1$ and $x_2$ is $D(x_1, x_2) = \frac{1}{4} \sigma^2 \text{tri}^2 \left( \frac{|x_1 - x_2|}{\sigma} \right) = \frac{1}{4} \sigma^2 \max(1 - \frac{|x_1 - x_2|}{\sigma}, 0)^2$.

Proof: Let us define the Triangle embedder as $\psi(t, x) = \text{tri} \left( \frac{x - t}{0.5\sigma} \right) = \max \left(1 - \frac{|x - t|}{0.5\sigma}, 0\right)$, where $\sigma$ 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

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

To make sure the stable rank is saturated, we assume that $d$ and $N$ are large enough. Then, $\Psi$ 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

$$C = \frac{1}{n} F_n^{-1} \text{diag}(F_n c) F_n, \quad (18)$$

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 $\Psi$ as a continuous signal, which is $\psi(x, t=0) = \text{tri} \left( \frac{x}{\sigma} \right)$, so the singular values are

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

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

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

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:

$$D(x_1, x_2) = \int_{-\infty}^{+\infty} \psi(t, x_1) \psi(t, x_2) dt$$

$$= \int_{-\infty}^{+\infty} \text{tri} \left( \frac{x - t}{0.5\sigma} \right) \text{tri} \left( \frac{x - t}{0.5\sigma} \right) dt$$

$$= \frac{1}{4} \sigma^2 \max \left(1 - \frac{|x_1 - x_2|}{\sigma}, 0\right)^2. \quad (21)$$

□
B 2D complex encoding

B.1 Closed form solution for separable coordinates

If pixels are sampled on a regular grid formed by samples 
\[ x = [x_1, x_2, \cdots, x_N]^T \]
and samples 
\[ y = [y_1, y_2, \cdots, y_M]^T \],
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 \( \Psi: \mathbb{R} \rightarrow \mathbb{R}^K \) 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,

\[
\begin{align*}
\arg \min_{\mathbf{W}} \| \text{vec} (\mathbf{S}) - (\Psi (y) \otimes \Psi (x)) \text{vec} (\mathbf{W}) \|^2_2,
\end{align*}
\]

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

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

which means,

\[
\begin{align*}
\mathbf{W}^* &= (\Psi (x)^T \Psi (x))^{-1} \Psi (x) \mathbf{S} (\Psi (y)^T \Psi (y))^{-1}.
\end{align*}
\]

B.2 Blending matrix for non-separable coordinates

First, we focus on 1D encoders. Given a 1D encoder \( \Psi: \mathbb{R} \rightarrow \mathbb{R}^K \) 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 \( x_0 \leq x \leq x_1 \). This problem can be solved by

\[
\begin{align*}
\arg \min_\alpha \| \Psi (x) - \left[ \Psi (x_0) \ \Psi (x_1) \right] \alpha \|^2_2,
\end{align*}
\]

where \( \alpha = [\alpha_0 \ \alpha_1]^T \). 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 \( \alpha^* \) can be solved by

\[
\begin{align*}
\alpha^* &= \arg \min_\alpha \| \Psi (x) - \left[ \Psi (x_0) \ \Psi (x_1) \right] \alpha \|^2_2 \\
&= \left( \left[ \Psi (x_0) \ \Psi (x_1) \right]^T \left[ \Psi (x_0) \ \Psi (x_1) \right] \right)^{-1} \left[ \Psi (x_0) \ \Psi (x_1) \right]^T \Psi (x) \\
&= \left( \left[ \Psi (x_0)^T \Psi (x_0) \ \Psi (x_1)^T \Psi (x_1) \right]^{-1} \left[ \Psi (x_0)^T \Psi (x) \ \Psi (x_1)^T \Psi (x) \right] \right) \Psi (x)
\end{align*}
\]
With the definition $D(x_1, x_2)$ in Appendix A, this can be written as,

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

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 $0 \leq \beta \leq 1$. Then, the solution becomes,

$$\alpha^* = \begin{bmatrix} D(0, 0) & D(0, 1) \\ D(1, 0) & D(1, 1) \end{bmatrix}^{-1} \begin{bmatrix} D(\beta d) \\ D((1 - \beta) d) \end{bmatrix}$$

$$= \frac{1}{D^2(0) - D^2(d)} \begin{bmatrix} D(0) & -D(d) \\ -D(d) & D(0) \end{bmatrix} \begin{bmatrix} D(\beta d) \\ D((1 - \beta) d) \end{bmatrix}. \quad (28)$$

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 $x = [x_1, x_2, \cdots, x_N]^T$ and samples $y = [y_1, y_2, \cdots, y_M]^T$. The query points are randomly sampled in the space as $Q = [q_1, q_2, \cdots, q_P]^T$, where $P$ is the number of points and each $q_i \in \mathbb{R}^{2 \times 1}$ is a random 2D coordinate. Let $s \in \mathbb{R}^{P \times 1}$ be the signal, and $\Psi: \mathbb{R} \to \mathbb{R}^K$ be the 1D encoder. We want to find the weights $W \in \mathbb{R}^{K \times K}$ of the linear layer by optimizing the following equation,

$$\arg \min_W \|s - B(Q)(\Psi(y) \otimes \Psi(x)) \text{ vec}(W)\|^2,$$  

(29)

where $B: \mathbb{R}^2 \to \mathbb{R}^{MN}$ is the non-linear interpolation coefficients function, i.e., $B(Q) \in \mathbb{R}^{P \times MN}$ 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 $q_i$, 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 $x_i \leq q_{i0} \leq x_{i+1}$ and $y_j \leq q_{p1} \leq y_{j+1}$. Then we can obtain the encoding for $q_{i0}$ and $q_{p1}$ as follows,

$$\Psi(q_{i0}) \approx \alpha_0 \Psi(x_i) + \alpha_1 \Psi(x_{i+1}),$$

$$\Psi(q_{p1}) \approx \beta_0 \Psi(y_j) + \beta_1 \Psi(y_{j+1}). \quad (30)$$

Then, the 2D encoding for $q_i$ is,

$$\Psi(q_i) = \Psi(q_{p0}, q_{p1})$$

$$= \Psi(q_{p1}) \otimes \Psi(q_{p0})$$

$$= (\beta_0 \Psi(y_j) + \beta_1 \Psi(y_{j+1})) \otimes (\alpha_0 \Psi(x_i) + \alpha_1 \Psi(x_{i+1}))$$

$$= \alpha_0 \beta_0 \Psi(x_i, y_j) + \alpha_0 \beta_1 \Psi(x_i, y_{j+1}) + \alpha_1 \beta_0 \Psi(x_{i+1}, y_j) + \alpha_1 \beta_1 \Psi(x_{i+1}, y_{j+1}) \quad (31)$$
which means $B(q_{\phi}) \in \mathbb{R}^{1 \times MN}$ 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 $X \in \mathbb{R}^{ND \times D}$ be $N^D$ points in $D$ dimensional space, $\Psi : \mathbb{R} \rightarrow \mathbb{R}^K$ be the 1D encoder, and we want to know the memory and computational complexity when the encoding multiply a linear layer $W$.

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

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

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

**Complex encoding (non-separable coordinates).** The embedding $\Psi(X) \in \mathbb{R}^{NK \times D}$, the weights $W \in \mathbb{R}^{K^D \times 1}$ and the Blending matrix $B(X) \in \mathbb{R}^{ND \times DP}$ (sparse matrix with only $N^D \times 2^D$ non-zeros values), so the memory complexity is $O(K^DN^D + 2^DN^D)$, the computational complexity is $2^DN^D + \sum_{i=1}^{D} N^i K^{D+1-i} = O(D^N + K^{-2}N^D)$. A special case of $N=K$ will be discussed later.

**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).**

$$
\phi(x) = \left[ \cdots, \cos \left( 2\pi \cdot \left( \frac{K-1}{K} 2^0 + \frac{i}{K} 2^\sigma \right) x \right), 
\sin \left( 2\pi \cdot \left( \frac{K-1}{K} 2^0 + \frac{i}{K} 2^\sigma \right) x \right), \cdots \right]^T,
$$

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

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

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

where $i=0, \ldots, K-1$ and $\sigma$ is the hyperparameter for frequency range. The frequency are sampled log-linearly from base frequency ($2^0$) to max frequency ($2^\sigma$).
RFF (Fourier feature-based encoding using randomly sampled frequency) [2].

\[ \phi(x) = \begin{bmatrix} \cos(2\pi b x)^T, \sin(2\pi b x)^T \end{bmatrix}^T, \]

(34)

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

Tri (shifted triangle encoding).

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

(35)

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

Gau (shifted Gaussian encoding).

\[ \phi(x) = \begin{bmatrix} \cdots, e^{-\frac{x-i/K}{2\sigma^2}}, \cdots \end{bmatrix}^T, \]

(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 64×64×64 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). \( \bullet \) are simple positional encodings. \( \bullet \) 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  | 21.54 ± 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   | 21.54 ± 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.

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)
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)
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.