Tuning the Frequencies: Robust Training for Sinusoidal Neural Networks - Supplementary Material -

Tiago Novello[1,2,*] Diana Aldana[1,*] Andre Araujo[2] Luiz Velho[1]
[1] IMPA [2] Google DeepMind

Contents

1. Theoretical analysis

This section presents the proofs of Theorems 1 and 2 enunciated in the main paper, and a generalization of Theorem 1 for deep sinusoidal MLPs. Then, we use the sine-cosine expansion of a sinusoidal INR (Equation (4) in the main paper) to obtain a closed formula for the coefficients of its Fourier series. Finally, we analyze some representation problems (see Sec 4.1 in the main paper) that may arise due to arbitrary initialization, and present a simple but effective solution to avoid them.

1.1. Proofs of Theorems 1 and 2

To prove Theorems 1 and 2 we consider the INR input to be 1D; the general case is analogous. Let $f(x) = \mathbf{C} \circ \mathbf{S} \circ \mathbf{D}(x) + e$ be a sinusoidal INR. Here, the input layer $\mathbf{D}(x) = \sin(\omega x + \varphi)$ projects the input x into a list of harmonics with frequencies $\omega = (\omega_1, \dots, \omega_m) \in \mathbb{R}^m$ and shifts $\varphi \in \mathbb{R}^m$. Then, the sinusoidal layer $\mathbf{S}(\cdot) = \sin(\mathbf{W} \cdot + \mathbf{b})$ with hidden matrix $\mathbf{W} \in \mathbb{R}^{n \times m}$ and bias $\mathbf{b} \in \mathbb{R}^n$ modulates those harmonics. Finally, those neurons are combined using $\mathbf{C} \cdot + e$, an affine transformation. The ith hidden neuron can be written as follows:

$$h_i(x) = \sin\left(\sum_{j=1}^{m} W_{ij} \sin(\omega_j x + \varphi_j) + b_i\right), \tag{1}$$

where W_{ij} are the ij coefficients of $\mathbf{W}$. Note that h_i receives a list of m input neurons $\sin(\omega_j x + \varphi_j)$ that are combined with the weights W_{ij} and activated by $\sin$. Before presenting the expansion of h_i, let us recall the Fourier series of a neuron with width 1 and no bias [1, Page 361]:

$$\sin\big(W_{11}\sin(x)\big) = \sum_{k \in \mathbb{Z}\text{ odd}} J_k(W_{11}) \sin(kx), \text{ with } J_k(W_{11}) = \frac{1}{\pi}\int_0^\pi \cos\big(kt - W_{11}\sin(t)\big)\,dt. \tag{2}$$

The functions J_k are the *Bessel functions* of the first kind. Theorem 1 provides an expansion for h_i which generalizes (2).

*These authors contributed equally to this work.

Theorem 1. *Each hidden neuron h_i of a 3-layer sinusoidal MLP has an amplitude-phase expansion of the form*

$$h_i(x) = \sum_{\mathbf{k}\in\mathbb{Z}^m} \alpha_{\mathbf{k}} \sin\big(\beta_{\mathbf{k}}\, x + \lambda_{\mathbf{k}}\big), \tag{3}$$

where $\beta_{\mathbf{k}} = \langle \mathbf{k}, \omega\rangle$, $\lambda_{\mathbf{k}} = \langle \mathbf{k}, \varphi\rangle + b_i$, and $\alpha_{\mathbf{k}} = \prod_j J_{k_j}(W_{ij})$ is the product of the Bessel functions of the first kind.

To prove Theorem 1 we use the following lemma which will also be helpful for the generalization presented in next section.

Lemma 1. *Given $\mathbf{a} = (a_1, \ldots, a_m), \mathbf{y} = (y_1, \ldots, y_m) \in \mathbb{R}^m$ and $b \in \mathbb{R}$, we have that*

$$\sin\left(\sum_{i=1}^{m} a_i \sin(y_i) + b\right) = \sum_{\mathbf{k}\in\mathbb{Z}^m} \alpha_{\mathbf{k}} \sin\Big(\langle \mathbf{k}, \mathbf{y}\rangle + b\Big), \quad \text{with} \quad \alpha_{\mathbf{k}} = \prod_{l=1}^{m} J_{k_l}(a_l). \tag{4}$$

Proof. The proof consists of verifying (4) as well as a similar formula using the cosine as the activation function:

$$\cos\left(\sum_{i=1}^{m} a_i \sin(y_i) + b\right) = \sum_{\mathbf{k}\in\mathbb{Z}^m} \alpha_{\mathbf{k}} \cos\Big(\langle \mathbf{k}, \mathbf{y}\rangle + b\Big). \tag{5}$$

The proof is by induction in m. For the base case $m = 1$, we prove $\sin\big(a_1 \sin(y_1) + b\big) = \sum_{k\in\mathbb{Z}} J_k(a_1) \sin(k y_1 + b)$. For this, we use (2) and its cosine analogous expansion $\cos\big(a_1 \sin(y_1)\big) = \sum J_l(a_1) \cos(l y_1)$, here the sum is over the even numbers. Thus, by the angle sum identity we obtain:

$$\begin{aligned}
\sin\big(a_1 \sin(y_1) + b\big) &= \sin\big(a_1 \sin(y_1)\big)\cos(b) + \cos\big(a_1 \sin(y_1)\big)\sin(b) \\
&= \sum_{k\in\mathbb{Z}\text{ odd}} J_k(a_1) \sin(k y_1)\cos(b) + \sum_{l\in\mathbb{Z}\text{ even}} J_l(a_1)\cos(l y_1)\sin(b) \\
&= \sum_{k\in\mathbb{Z}\text{ odd}} J_k(a_1) \sin(k y_1 + b) + \sum_{l\in\mathbb{Z}\text{ even}} J_l(a_1)\sin(l y_1 + b) \\
&= \sum_{k\in\mathbb{Z}} J_k(a_1) \sin(k y_1 + b).
\end{aligned}$$

In the third equality we combined the formula $\sin(u)\cos(v) = \frac{\sin(u+v)+\sin(u-v)}{2}$ and the fact that $J_{-k}(u) = (-1)^k J_k(u)$ to rewrite the summations. The proof of the formula using the cosine as an activation function is similar.

Assume that the formulas hold for $m-1$, with $m > 1$, we prove that (4) holds for m (the **induction step**).

$$\begin{aligned}
\sin\left(\sum_{j=1}^{m} a_j \sin(y_j) + b\right) &= \sin\left(\sum_{j=1}^{m-1} a_j \sin(y_j) + b\right)\cos\big(a_m \sin(y_m)\big) && (6)\\
&\quad + \cos\left(\sum_{j=1}^{m-1} a_j \sin(y_j) + b\right)\sin\big(a_m \sin(y_m)\big) && (7)\\
&= \sum_{\mathbf{l}\in\mathbb{Z}^{m-1},\, k\in\mathbb{Z}\text{ even}} \alpha_{\mathbf{l}} J_k(a_m) \sin\Big(\langle \mathbf{l}, \mathbf{y}\rangle + b\Big)\cos(k y_m) && (8)\\
&\quad + \sum_{\mathbf{l}\in\mathbb{Z}^{m-1},\, k\in\mathbb{Z}\text{ odd}} \alpha_{\mathbf{l}} J_k(a_m) \cos\Big(\langle \mathbf{l}, \mathbf{y}\rangle + b\Big)\sin(k y_m) && (9)\\
&= \sum_{\mathbf{k}\in\mathbb{Z}^m} \alpha_{\mathbf{k}} \sin\Big(\langle \mathbf{k}, \mathbf{y}\rangle + b\Big) && (10)
\end{aligned}$$

We use the induction hypothesis in the second equality and an argument similar to the one used in the base case to rewrite the harmonic sum. Again, the cosine activation function case is analogous. □

Note that the proof of Theorem 1 is a particular case of Lemma 1 with $\mathbf{a} = \mathbf{W}_i$, $\mathbf{y} = \omega\mathbf{x} + \varphi$, and $b = b_i$; where $\mathbf{W}_i$ is the ith row of $\mathbf{W}$. Yüce et al. [5] presented a similar formula for MLPs activated by polynomial functions. While it is evident that the sine function can be approximated by a polynomial using Taylor series, our formula requires no approximations. In addition to providing a simple proof, we also derive the analytical expressions for the amplitudes. These expressions enable us to compute upper bounds (Theorem 2) for the new frequencies $\alpha_{\mathbf{k}}$ in terms of $\mathbf{k}$ and $\mathbf{W}$.

Theorem 2. *The magnitude of the amplitudes $\alpha_{\mathbf{k}}$ in the expansion (3) is bounded by $\prod_{j=1}^{m} \left(\frac{|W_{ij}|}{2}\right)^{|k_j|} \frac{1}{|k_j|!}$.*

Proof. Theorem 1 says that $\alpha_{\mathbf{k}} = \prod_{j=1}^{m} J_{k_j}(W_{ij})$. To estimate an upper bound for this number, we use the following inequality [3], which gives an upper bound for the Bessel functions J_k.

$$|J_k(W_{ij})| < \frac{\left(\frac{|W_{ij}|}{2}\right)^k}{k!}, \quad k > 0, \quad W_{ij} > 0. \tag{11}$$

Observe that this inequality also holds for $W_{ij} < 0$ since $|J_k(-u)| = |J_k(u)|$. Therefore, replacing (11) in $\prod_{j=1}^{m} J_{k_j}(W_{ij})$ and using $|J_{-k}(W_{ij})| = |J_k(W_{ij})|$ results in the desired inequality. □

1.2. Extension to deeper networks

We extend Theorem 1 to deeper neurons using a recursive argument. For this, let f be a sinusoidal MLP with $d > 1$ hidden layers defined as $f(\mathbf{x}) = \mathbf{C} \circ \mathbf{S}_d \circ \cdots \circ \mathbf{S}_1 \circ \mathbf{S}_0(\mathbf{x}) + e$ where $\mathbf{S}_i(\mathbf{x}) = \sin\left(\mathbf{W}^i\mathbf{x} + \mathbf{b}^i\right)$ is a sinusoidal layer with hidden weights $\mathbf{W}^i \in \mathbb{R}^{n_{i+1} \times n_i}$ and bias $\mathbf{b}^i \in \mathbb{R}^{n_{i+1}}$. For simplicity, we denote $\mathbf{W}^0 := \omega$ and $\mathbf{b}^0 := \varphi$.

Now, the jth neuron of the ith hidden layer h_j^i can be written as

$$h_j^i(\mathbf{x}) = \sin\left(\sum_k \mathbf{W}_{jk}^i \underbrace{\sin(\mathbf{y}^{i-1})}_{\mathbf{h}^{i-1}(\mathbf{x})} + b_j^i\right), \tag{12}$$

where $\mathbf{y}^{i-1}$ is the linear part of $\mathbf{h}^{i-1}(\mathbf{x})$. Again, we prove that (12) has an expansion which generalizes (3) to deeper networks.

Theorem 3. *Each hidden neuron $h_j^i(\mathbf{x})$ of a sinusoidal INR has the following expansion,*

$$h_j^i(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^{n_1 + \cdots + n_i}} \alpha \, \sin\left(\left\langle \mathbf{k}^1, \omega \right\rangle \mathbf{x} + \lambda\right), \tag{13}$$

with $\mathbf{k} = \left(\mathbf{k}^1, \cdots, \mathbf{k}^i\right)$ being a tuple of integer vectors, $\lambda = b_j^i + \sum_{l=1}^{i} \left\langle \mathbf{k}^l, \mathbf{b}^{l-1} \right\rangle$, and $\alpha = \alpha_i(\mathbf{W}_j^i) \prod_{l=1}^{i-1} \alpha_l(\mathbf{k}^{l+1}\mathbf{W}^l)$ where $\alpha_l(\mathbf{a}) := J_{k_1^l}(a_1) \cdots J_{k_{n_l}^l}(a_{n_l})$ is the product of the Bessel functions of the first kind and $\mathbf{W}_j^i$ is the jth row of $\mathbf{W}^i$.

Proof. The proof is by induction on the depth i of the neuron $h_j^i(\mathbf{x})$. First, note that the **base case** ($i = 1$) is Theorem 1. For the **induction step**, we suppose the hypothesis holds for depth $i - 1 > 0$ and prove that it holds for i. In this case, we have that the jth neuron of the ith layer is given by,

$$h_j^i(\mathbf{x}) = \sin\left(\mathbf{W}_j^i \sin\left(\mathbf{W}^{i-1}\mathbf{h}^{i-2}(\mathbf{x}) + \mathbf{b}^{i-1}\right) + b_j^i\right).$$

Considering $\mathbf{a} := \mathbf{W}_j^i$, $\mathbf{y} := \mathbf{W}^{i-1}\mathbf{h}^{i-2}(\mathbf{x}) + \mathbf{b}^{i-1}$ and $b := b_j^i$, Lemma 1 implies that

$$\begin{aligned} h_j^i(\mathbf{x}) &= \sum_{\mathbf{k}^i \in \mathbb{Z}^{n_i}} \alpha_i(\mathbf{a}) \, \sin\left(\left\langle \mathbf{k}^i, \mathbf{y} \right\rangle + b\right) \\ &= \sum_{\mathbf{k}^i \in \mathbb{Z}^{n_i}} \alpha_i\left(\mathbf{W}_j^i\right) \underbrace{\sin\left(\left\langle \mathbf{k}^i, \mathbf{W}^{i-1}\mathbf{h}^{i-2}(\mathbf{x}) + \mathbf{b}^{i-1} \right\rangle + b_j^i\right)}_{\widetilde{h}^{i-1}(\mathbf{x})}, \end{aligned} \tag{$*$}$$

where $\widetilde{h}^{i-1}(\mathbf{x})$ is a list of hidden neurons with weights $\widetilde{\mathbf{W}} := \mathbf{k}^i \mathbf{W}^{i-1}$ and bias $\widetilde{b} := b^i_j + \langle \mathbf{k}^i, \mathbf{b}^{i-1} \rangle$:

$$\widetilde{h}^{i-1}(\mathbf{x}) = \sin\left(\widetilde{\mathbf{W}} \sin\left(\mathbf{W}^{i-2}\mathbf{h}^{i-3}(\mathbf{x}) + \mathbf{b}^{i-2}\right) + \widetilde{b}\right).^{1}$$

Since $\widetilde{h}^{i-1}$ has depth $i-1$, the induction hypothesis implies

$$\widetilde{h}^{i-1}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^{n_1+\cdots+n_{i-1}}} \alpha \sin\Big(\langle \mathbf{k}^1, \mathbf{W}^0 \rangle \mathbf{x} + \lambda\Big), \tag{14}$$

where α and λ are defined as follows:

$$\alpha = \alpha_{i-1}(\widetilde{\mathbf{W}}) \prod_{l=1}^{i-2} \alpha_l(\mathbf{k}^{l+1}\mathbf{W}^l) = \prod_{l=1}^{i-1} \alpha_l(\mathbf{k}^{l+1}\mathbf{W}^l), \quad \lambda = \widetilde{b} + \sum_{l=1}^{i-1} \langle \mathbf{k}^l, \mathbf{b}^{l-1} \rangle = b^i_j + \sum_{l=1}^{i} \langle \mathbf{k}^l, \mathbf{b}^{l-1} \rangle .$$

Replacing (14) in Equation $(*)$, we obtain the desired expression:

$$\begin{aligned} h^i_j(\mathbf{x}) &= \sum_{\mathbf{k}^i \in \mathbb{Z}^{n_i}} \alpha_i(\mathbf{W}^i_j) \sum_{\mathbf{k} \in \mathbb{Z}^{n_1+\cdots+n_{i-1}}} \alpha \, \sin\left(\langle \mathbf{k}^0, \mathbf{W}^0 \rangle \mathbf{x} + \lambda\right) \\ &= \sum_{\mathbf{k} \in \mathbb{Z}^{n_1+\cdots+n_i}} \left(\alpha_i(\mathbf{W}^i_j)\, \alpha\right) \, \sin\left(\langle \mathbf{k}^0, \mathbf{W}^0 \rangle \mathbf{x} + \lambda\right). \end{aligned}$$

□

Note that several properties observed in INRs with a single hidden layer also extend to the general case. First, the amplitudes α depend only on the hidden matrices and remain products of Bessel functions. Second, the biases determine the shifts λ, suggesting that not all of them may be necessary during training. Furthermore, we emphasize that adding more hidden layers does not introduce new frequencies, as these are still given by $\langle \mathbf{k}, \omega \rangle$. Instead, deep sinusoidal networks primarily refine the amplitudes of the generated frequencies, which are entirely determined by the choice of ω.

1.3. Towards computing the Fourier series of a sinusoidal MLP

Without loss of generality, we will assume the 3-layer sinusoidal MLP f (defined in Section 1.1) has $\mathbb{R}^2$ as its domain. To compute its Fourier series, we recall that Equation (4) from the main paper states that f can be rewritten as

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^m} \langle \mathbf{C}, A_{\mathbf{k}} \rangle \sin\left(\beta_{\mathbf{k}} \mathbf{x}\right) + \langle \mathbf{C}, B_{\mathbf{k}} \rangle \cos\left(\beta_{\mathbf{k}} \mathbf{x}\right) + e, \tag{15}$$

where $\beta_{\mathbf{k}} = \langle \mathbf{k}, \omega \rangle = \mathbf{k}^\top \omega$. This expression resembles a Fourier series, however, a given frequency $\mathcal{F} \in \mathbb{Z}^2$ may appear several times as generated frequencies $\beta_{\mathbf{k}}$ associated with different coefficients $\mathbf{k}$. Here, we provide a characterization of all $\mathbf{k} \in \mathbb{Z}^m$ such that $\mathbf{k}^\top \omega = \frac{2\pi}{p}\mathcal{F}$, where p is the INR period, leading to an algorithm to compute the Fourier series of f.

To guarantee that f, initialized with $\omega = \frac{2\pi}{p}[\mathbf{f}^x, \mathbf{f}^y] \in \frac{2\pi}{p}\mathbb{Z}^{m\times 2}$, can represent an arbitrary signal, we must determine when it can reconstruct any given frequency $\mathcal{F} \in \mathbb{Z}^2$. This is equivalent of finding a solution $\mathbf{k}$ to the following system of Diophantine equations:

$$[\mathbf{f}^x, \mathbf{f}^y]^\top \mathbf{k} = \mathcal{F} \quad \text{for any} \quad \mathcal{F} \in \mathbb{Z}^2. \tag{16}$$

To solve (16) we use the approach in [6, p. 50]. Specifically, we consider the Smith normal form of $[\mathbf{f}^x, \mathbf{f}^y]$, that is, $\mathbf{B} = \mathbf{U}[\mathbf{f}^x, \mathbf{f}^y]^\top \mathbf{V}$, where $\mathbf{U} \in \mathbb{Z}^{2\times 2}$ and $\mathbf{V} \in \mathbb{Z}^{m\times m}$ are inversible matrices. Defining $\mathbf{e} = \mathbf{U}\mathcal{F}$, there exist solution for (16) only if B_{ii} divides e_i for all i. Thus, to have integer solutions for (16) we need $e_i/B_{ii} \in \mathbb{Z}$. Finally, the solutions have the form $\mathbf{V}\left[\begin{smallmatrix} \frac{e_1}{B_{11}} & \frac{e_2}{B_{22}} & l_1 & \cdots & l_{m-2} \end{smallmatrix}\right]^T$ with $l_1, \ldots, l_{m-2}$ arbitrary integers.

Now, recall that we are considering $\omega_1 = \frac{2\pi}{p}(1,0)$ and $\omega_2 = \frac{2\pi}{p}(0,1)$. Then, the Smith normal form and unimodular matrices $\mathbf{U}$, $\mathbf{V}$ are reduced to,

$$\mathbf{B} = [\mathbb{I}_2 \,|\, \mathbf{0}_{2\times m-2}], \quad \mathbf{U} = \mathbb{I}_2, \quad \text{and} \quad \mathbf{V} = \left[\begin{array}{c|ccc} \mathbb{I}_2 & -\mathrm{f}^x_3 & \cdots & -\mathrm{f}^x_m \\ & -\mathrm{f}^y_3 & \cdots & -\mathrm{f}^y_m \\ \hline \mathbf{0}_{m-2\times 2} & \multicolumn{3}{c}{\mathbb{I}_{m-2}} \end{array}\right].$$

[1] For $i = 2$, we define $\mathbf{h}^{i-3}(\mathbf{x}) = \mathbf{x}$.

Since $B_{ii} = 1$, the divisibility condition is satisfied. Additionally, we have $\mathbf{e} = \mathcal{F}$ which implies that the set of integer solutions of (16) is given by

$$\mathcal{C}_{\mathcal{F}} = \left\{ \begin{bmatrix} \mathcal{F}_1 - l_1 \mathrm{f}_3^x - \cdots - l_{m-2} \mathrm{f}_m^x \\ \mathcal{F}_2 - l_1 \mathrm{f}_3^y - \cdots - l_{m-2} \mathrm{f}_m^y \\ l_1 \\ \vdots \\ l_{m-2} \end{bmatrix} : l_1, ..., l_{m-2} \in \mathbb{Z} \right\}.$$

We obtain the Fourier series of f by aggregating all coefficients associated with each frequency $\mathcal{F} \in \mathbb{Z}^2$ in (15). That is, the Fourier coefficients are given by $\widehat{A}_{\mathcal{F}} = \sum_{\mathbf{k} \in \mathcal{C}_{\mathcal{F}}} \langle \mathbf{C}, A_{\mathbf{k}} \rangle$ and $\widehat{B}_{\mathcal{F}} = \sum_{\mathbf{k} \in \mathcal{C}_{\mathcal{F}}} \langle \mathbf{C}, B_{\mathbf{k}} \rangle$.

1.4. On the sub-periodicity of sinusoidal MLPs

Initializing a sinusoidal MLP f using input neurons with period p implies that f is also periodic with period p, however, the initialization could generate sub-periods implying that we can not fit a signal with fundamental period p. Next, we derive a condition to avoid such a problem. First, recall that Theorem 1 says that a neuron $h(\mathbf{x}) = \sin\left(\mathbf{W}_i \sin(\omega \mathbf{x} + \varphi) + b_i\right)$, with $\omega = \frac{2\pi}{p}[\mathbf{f}^x, \mathbf{f}^y]$ for some $\mathbf{f}^x, \mathbf{f}^y \in \mathbb{Z}^m$, can be expressed as:

$$h(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^m} \alpha_{\mathbf{k}} \sin\left(\beta_{\mathbf{k}} \mathbf{x} + \lambda_{\mathbf{k}}\right) \tag{17}$$

with $\beta_{\mathbf{k}} = \langle \mathbf{k}, \omega \rangle$ and $\lambda_{\mathbf{k}} = \langle \mathbf{k}, \varphi \rangle + b_i$. Therefore, $\beta_{\mathbf{k}} \mathbf{x} = \frac{2\pi}{p}\Big(\langle \mathbf{k}, \mathbf{f}^x \rangle\, x + \langle \mathbf{k}, \mathbf{f}^y \rangle\, y \Big)$. Now, suppose that h has sub-period $\frac{p}{q}$ in x-axis and $\frac{p}{s}$ in y-axis ($q, s \in \mathbb{Z}^+$), i.e. $h(x, y) = h\left(x + \frac{p}{q}, y + \frac{p}{s}\right)$. Then, (17) implies

$$h\left(x + \frac{p}{q}, y + \frac{p}{s}\right) = \sum_{\mathbf{k} \in \mathbb{Z}^m} \alpha_{\mathbf{k}} \sin\left(\beta_{\mathbf{k}} \mathbf{x} + \lambda_{\mathbf{k}} + 2\pi \left\langle \mathbf{k}, \frac{\mathbf{f}^x}{q} + \frac{\mathbf{f}^y}{s} \right\rangle\right). \tag{18}$$

Thus, since $\sin(x + 2\pi k) = \sin(x)$ for $k \in \mathbb{Z}$, we have that (17) and (18) are equal only if $\left\langle \mathbf{k}, \frac{\mathbf{f}^x}{q} + \frac{\mathbf{f}^y}{s} \right\rangle \in \mathbb{Z}$ for all $\mathbf{k} \in \mathbb{Z}^m$. Therefore, h (and consequently f) has sub-periods only if there exists $q > 1$ or $s > 1$ such that $\frac{\mathbf{f}^x}{q} + \frac{\mathbf{f}^y}{s} \in \mathbb{Z}^m$. To avoid this problem, we define the first elements of $[\mathbf{f}^x, \mathbf{f}^y]$ as $\mathrm{f}_1^x = 1$, $\mathrm{f}_1^y = 0$, $\mathrm{f}_2^x = 0$, and $\mathrm{f}_2^y = 1$. Note that this implies $\frac{\mathrm{f}_1^x}{q} + \frac{\mathrm{f}_1^y}{s} = \frac{1}{q} \in \mathbb{Z}$ if and only if $q = 1$, and $\frac{\mathrm{f}_2^x}{q} + \frac{\mathrm{f}_2^y}{s} = \frac{1}{s} \in \mathbb{Z}$ if and only if $s = 1$.

Fig 1 shows cases where the initialization of ω does not hold the above condition resulting in poor reconstructions. We train networks with architecture $m = 32$, $n = 1500$ (number of hidden neurons), and bandlimit $\mathcal{b} = 100$. We networks are trained for 3000 epochs on a 256^2 resolution image. In Fig 1(a) and (b), we initialize $[\mathbf{f}^x, \mathbf{f}^y]$ using even frequencies and the cartesian product of $\{1, 10, 100\}$, with period $p = 2$, respectively. In Fig 1(c), we use the same initialization as in Fig1(a) but increase the period to $p = 3$ and visualize an extrapolation of the training domain. This highlights the overlap of copies caused by the sub-periodicity due the poor initialization. We fix this by adding $(0, 1), (1, 0)$ to ω, see Fig 3 in the main text.

(a) ω defined with even frequencies. (b) ω=$\{(u, v) | u, v \in \pm[1, 10, 100]\}$ (c) Extrapolation to $[-2, 2]^2$ of f ($p = 3$) with ω as in (a) .

Figure 1. Bad reconstructions given by specific (wrong) initializations of the input frequencies.

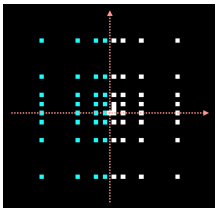
2. Additional experiments

2.1. Initialization

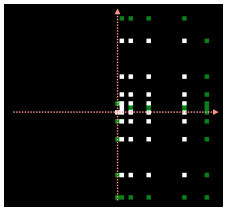
Note that initializing a frequency ω_i implies that its negative $-\omega_i$ also appears in the spectrum. This is a consequence of $\sin(\omega_j \mathbf{x} + \varphi_j) = \cos(\varphi_j)\sin(\omega_j \mathbf{x}) - \sin(\varphi_j)\sin(-\omega_j \mathbf{x} + \pi/2)$. Then, we only need to sample in half of $[-b, b]^2$, with b being the bandlimit of the input frequencies, allowing the sampling of more frequencies. We use this fact to avoid sampling duplicated frequencies to obtain a better reconstruction. We test the capacity of two INRs with the same architecture $m = 102$, $n = 512$, but with different initializations for ω (blue and white dots in Fig 2a) and ω' (white/green dots in Fig 2b):

$$\omega = \left[(k,l)|\ k,l \in \{1,3,4,7,10,20\}\right] \text{ and } \omega' = \left[(k,l)|\ (k,l) \in \omega \text{ with } k > 0\right] \sqcup \eta$$

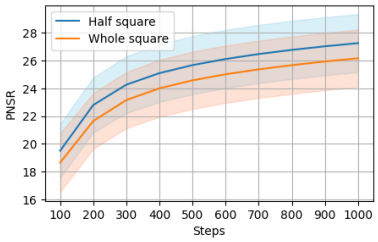
where η are additional frequencies (in green) sampled in the Cartesian product of $[0,1,2,3,4,7,10,20]$.



(a) Initialization of ω. Blue dots are the negative frequencies of those in white.



(b) Initialization of ω'. The frequencies in green replace those in blue.



(c) Comparison of PSNR during training.

Figure 2. Comparison of reconstructions with different input frequency initializations.

Fig 2(c) shows the PSNR of the resulting networks with respect to the iterations. The results show that adding the new frequencies η results in a slight but consistent improvement in the PSNR during training. Also, from (15) the final bias e represents the amplitude of the frequency $(0,0)$, hence we do not initialize it in ω.

In Sec 4.2 of the main paper we proposed an initialization for the hidden matrix $\mathbf{W}$ based on the bound values. The following experiment shows that such an initialization may grant faster convergence. In Fig 3, we initialize the INRs with size $m = 416$ and $n = 1024$, bandlimit $b = 82$, and bounds $c_{\mathbf{L}} = 1.0$, $c_{\mathbf{H}} = 0.2$. We train the networks during 10 epochs and fit an image with resolution 1024^2. Observe that our initialization for $\mathbf{W}$ (below) offers a better reconstruction than the uniform initialization (above). This can also be observed in the zoom-ins of images, where our method presents more details.

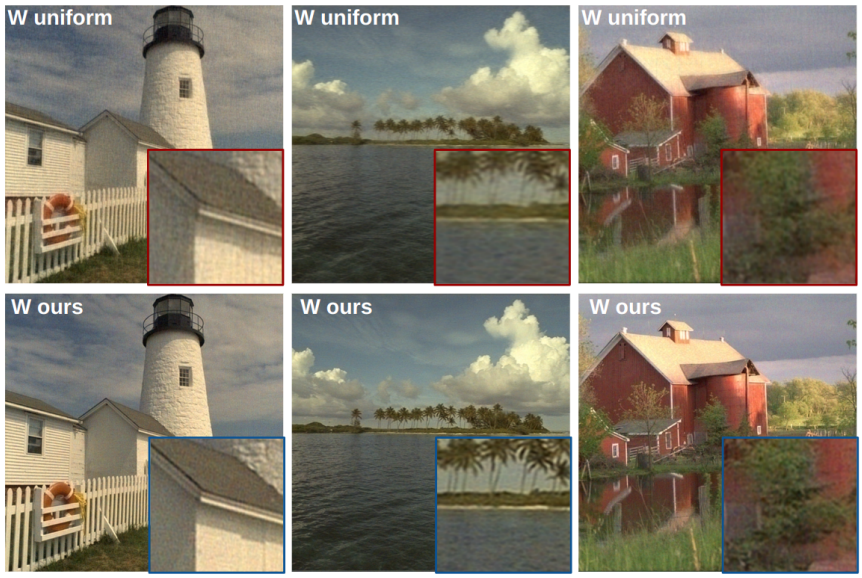


Figure 3. Comparison between INRs with different initializations for $\mathbf{W}$. The row above/below shows the reconstruction of a network with $\mathbf{W}$ initialized as in [4]/Sec 4.2. We trained during 10 epochs. The blue and red squares present a zoom-in of the image center.

2.2. Learned bounds

Sec 4.2 of the main paper introduced an architecture that enables learning the bounds during training. Here, we study its behavior as the sampling of input frequencies vary. We compare sinusoidal INRs of size $m = n = 416$, initializing the learned bounds as 0.5 and varying the bandlimit between $b = 41, 85, 171, 256$. The lower frequency limit is set to $\ell = b/4$. Fig 4 shows the bounds trained over 400 epochs on a 512^2 resolution image. Each plot includes a green dashed vertical line that splits the low (**L**) and high (**H**) frequencies. Figs 4(a)-(b) show the learned bounds when initializing using small b, leading to similar bounds across all frequencies. On the other hand, Figs 4(c)-(d) consider higher values of b. Here, low/high frequency bounds increase/decrease, reducing noise generation as bigger bounds may imply in higher multiples of the input frequencies. This behavior aligns with our claims that such bounds serve as a mechanism of spectral control.

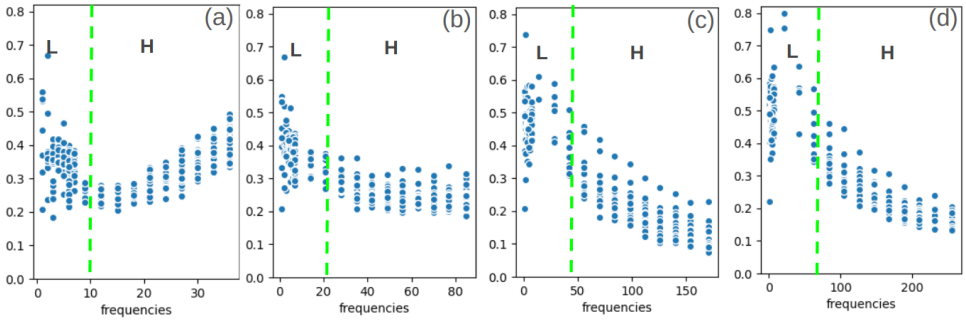


Figure 4. Ablation of the learned bounds of each column. The x-axis corresponds to the maximum coordinate (in absolute) of the frequency ω_j, while the y-axis shows the trained bound c_j of $\mathbf{W}$'s j-th column. Thus, the blue points in the diagram represent $(\max(|\omega_j|), c_j)$.

2.3. Representational capacity of layer composition

Theorem 1 states that a sinusoidal INR with a single hidden layer and input frequencies ω can represent a signal using an infinite number of frequencies $\beta_{\mathbf{k}} = \langle \mathbf{k}, \omega \rangle$. Here, we illustrate how the composition of layers enables a more compact and expressive representation by generating additional frequencies. Specifically, Fig 5 compares the representation capacity of a wide but shallow network and a narrower, deeper network. For the shallow case, we use a INR with no hidden layers $f(\mathbf{x}) = \mathbf{C}\sin(\omega\mathbf{x} + \varphi) + e$ where $\omega \in \frac{2\pi}{p}\mathbb{Z}^{11300\times 2}$. Training in this case optimizes the amplitudes $\mathbf{C}$ of the frequencies ω. For the deeper case, we train a network with a hidden layer composition, using a configuration of $m = 120$ and $n = 239$.

Theorem 1 then implies that f has as an expansion with amplitudes determined by the hidden weights. Thus, training a deeper sinusoidal INR also fits the amplitudes of a sum of sines, but with a much wider number of frequencies. Indeed, Fig 5 shows that this network not only converges faster but also achieved better quality with half the number of parameters.

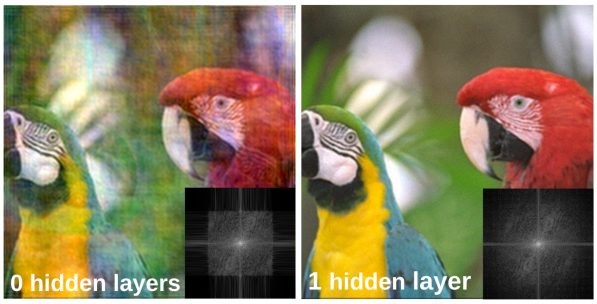


Figure 5. Comparison between INRs for fitting images, with zero and one hidden layers. Training with one hidden layer is significantly faster (50s vs 13m), uses half the parameters (33842 vs 67803), and produces higher quality results (34.4dB vs 21.1dB).

Next, we extend our ablations to include deeper sinusoidal MLPs. Specifically, we consider MLPs of two hidden layers with 256 neurons each and train them on Kodak dataset images with a resolution of 512^2. We set the bounds as $c_{\mathbf{L}} = 1.0$, $c_{\mathbf{H}} = 0.3$, and $c_{\mathbf{2}} = 0.05$, where the last one corresponds to the bound of the hidden matrix $\mathbf{W}^2$. Table 1 compares our initialization (2rd column) of the input frequencies with uniform initialization (1rd column), using b=128, ℓ=10, 70% of ω set as lower frequencies, and 200 training steps. Additionally, we include a comparison with our full method (3rd column). This shows that TUNER is also effective for deeper MLPs.

	rand init	our init	TUNER
RGB	16.86	28.47	**28.58**
grad	17.67	24.95	**25.42**

Table 1. Experiments with deeper MLPs. We compare our initialization of the input freq. with uniform and our full method (3rd column).

2.4. Additional comparisons

This section provides additional comparisons between different approaches to signal fitting using sinusoidal INRs and their variations. Figure 6 presents the gradient and error maps of SIREN and TUNER reconstructions after 3000 training epochs. We use networks of size $m = n = 416$ with $b = 171$, training them on images of resolution 512^2 while supervising with 90% of the available samples.

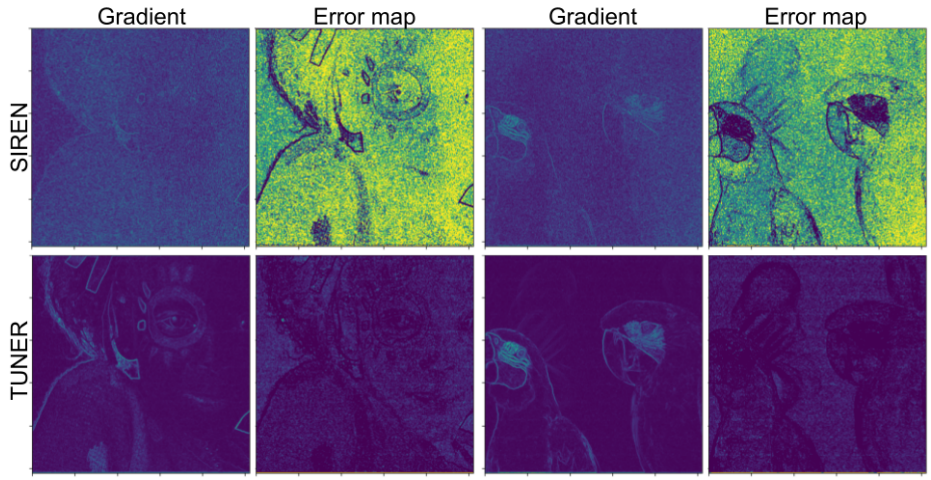


Figure 6. Gradient visualization of experiments in Table 3. The error map shows that while SIREN fits well only the higher frequencies, ours fit both.

Note that compared to SIREN, our gradient has no notable noise in both images. In particular, note that the error maps of SIREN have low error around regions of high detail (such as the silhouette of the girl, or near the eyes of the macaws) while regions of low detail have high error (like the background of the macaws or the face of the girl). This indicates that SIREN represented well the areas with high frequency content but those as propagated as noise in regions of low frequency content. Conversely, our error maps show that TUNER was able to represent well most of the image, tuning the frequencies to the spectral content of each region.

Now, we show an extended version of the bandlimit control experiment from Fig 1 (main paper). Specifically, we compare BACON (red) and TUNER (Ours, in blue), trained using different bandlimits for the network $(85, 171, 256)$. In Fig 7, we observe that for the low bandlimit (leftmost two), BACON generates visible ringing artifacts. In contrast, our method resembles a soft filter preserving the smoothness of the reconstruction. Conversely, when increasing the bandlimit of the spectrum, the reconstruction of BACON distorts color, while ours improves reconstruction.

TUNER and BANF. We present a numerical comparison with BANF on the DIV2K dataset [2], rescaling the images to a resolution of 256^2. We compare different bands ($b = 64$ and $b = 256$) for the spectrum and measure the PSNR in a resolution of 512^2. To assess the bandlimiting capacity of each network, we consider the amplitudes $\alpha_{\mathbf{k}}$ of frequencies $\langle \mathbf{k}, \omega \rangle$ outside of the band $\mathcal{B}$ to be noise. Then, we define the bandlimit error as:

$$\text{Error}_{\text{FFT}} = \sum_{\langle \mathbf{k}, \omega \rangle \notin \mathcal{B}} |\alpha_{\mathbf{k}}|.$$

For each level of detail in BANF, we initialized and trained a new TUNER INR from scratch. The epochs used were adjusted so that our training time was equivalent to BANF's, since their epochs took longer to train. The quantitative results are

Figure 7. Comparison between BACON (red) and our method (blue), trained with different bandlimits $\mathfrak{b} = 85, 171, 256$. We observe that BACON uses a box filter, generating ringing artifacts (first image). In contrast, our method resembles a soft filter, improving quality.

summarized in Table 2. Observe that TUNER outperforms BANF on both metrics for all bandlimits, even when it is not trained over the residuals of previous resolutions.

	PSNR↑		$\text{Error}_{\text{FFT}}\downarrow$	
Band	64	256	64	256
BANF	22.96	29.59	30.71	50.17
TUNER	**26.31**	**32.65**	**18.97**	**36.16**

Table 2. Comparison of TUNER and BANF in signal quality and bandlimiting error $\text{Error}_{\text{FFT}}$ with bands 64 and 256. Our method has better quality ($\approx$ 3dB) and reduces the appearance of frequencies outside the band.

We also present a qualitative comparison with BANF, showing two reconstructions with a resolution of 512^2 trained with a bandlimit of 128, under the same conditions as in Table 2. Figure 8 shows that our method preserves more details (with at least a 3dB of improvement) while enforcing a smooth filtering outside the band (red square). In contrast, BANF exhibits many frequencies outside the specified bandlimit, which is reflected in the higher $\text{Error}_{\text{FFT}}$ values.

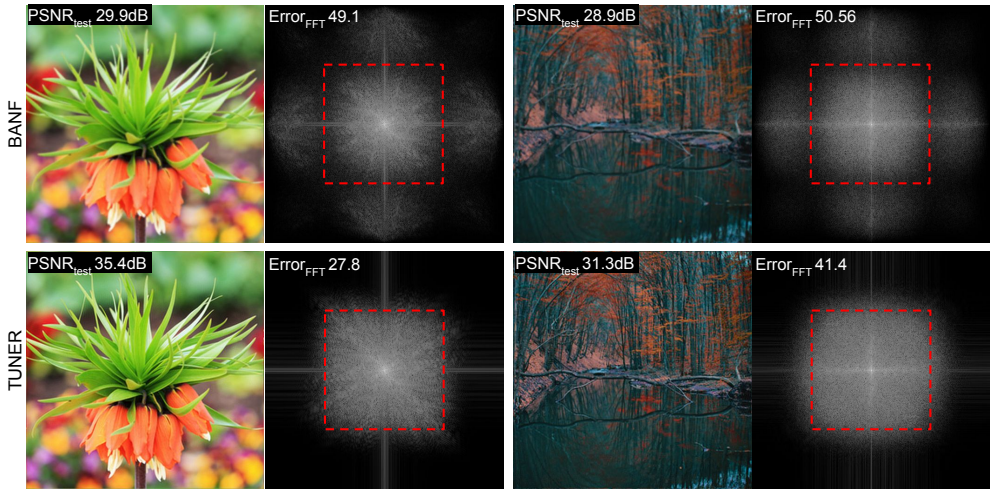


Figure 8. Comparison between TUNER and BANF in the signal (1st and 3rd column) and spectral (2nd and 4th columns) domains. Note that our reconstruction offers more details while restricting the appearance of frequencies to the band (red square) akin to a soft filter. Conversely, BANF exhibits artifacts in the spectrum, even though its representation appears blurrier.

Compare with FINER & Fourier Feature Mapping (FFM). We present a comparison between Fourier Feature Mapping (FFM), FINER, and TUNER on the DIV2K dataset with images of resolution 512^2. All networks are configured with a single hidden layer of size $m = n = 256$ and their corresponding initializations. For FINER, we use $\mathfrak{b} = 85$ and $\mathbf{b} \sim \mathcal{U}(-1, 1)$. The TUNER INR has a period of 3 and bounds $c_{\mathbf{L}} = 1.0$, $c_{\mathbf{H}} = 0.6$. All networks are trained for 5000 epochs using the Adam optimizer with a learning rate of 5×10^{-4}. As shown in Table 3, TUNER outperforms both methods with at least a

2 dB improvement. We also observe a qualitative improvement in Figure 9, where our method achieves comparable quality to previous works while enhancing the reconstruction of higher-order information.

epochs	FFM	FINER	TUNER
1000	29.42	30.23	**32.14**
5000	31.19	31.00	**33.16**

Table 3. Comparison between Fourier Feature Mapping (FFM), FINER and TUNER when training during 1000 and 5000 epochs.

Figure 9. Comparison between Fourier Feature Mapping (FFM), FINER and TUNER in the RGB and gradient (grayscale band) domains after training for 1000 epochs. Observe that FFM reconstruction has a signal quality comparable to ours, but their gradients are noisier, while FINER presents a smoother gradient but fails to reconstruct more detailed regions.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1964. 1

[2] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, July 2017. 8

[3] R. Paris. An inequality for the bessel function $j_\nu(\nu x)$. *SIAM journal on mathematical analysis*, 15(1):203–205, 1984. 3

[4] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 6

[5] G. Yüce, G. Ortiz-Jiménez, B. Besbinar, and P. Frossard. A structured dictionary perspective on implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19228–19238, 2022. 3

[6] R. Zippel. *Effective polynomial computation*, volume 241. Springer Science & Business Media, 2012. 4