



ÉCOLE NORMALE SUPÉRIEURE DE PARIS-SACLAY, CENTRE  
BORELLI

PLATEAU DE SACLAY, PARIS

---

Research visit - M1 Hadamard

IMPLICIT NEURAL PRESENTATIONS

Report  
AMER ESSAKINE  
supervised by  
DR ANGELICA AVILES RIVERO

---

May 2024-August 2024

# Abstract

Implicit neural representations have recently emerged as a novel paradigm for addressing computer vision tasks. This approach is based on representing data as continuous functions, in contrast to traditional grid-based methods. Implicit neural representations (INRs) utilize the expressive power of neural networks to approximate these implicitly defined, continuous signals. Specifically, the signal is parameterized by a multilayer perceptron (MLP) with alternating linear layers and element-wise nonlinear activation functions. The MLP takes coordinates as input and approximates the signal at those coordinates.

The continuous parametrization provided by INRs offers several advantages over traditional methods. For instance, INRs are significantly more memory-efficient than grid-based approaches, making them well-suited for high-dimensional problems such as 2D and 3D reconstruction. Moreover, INRs are highly versatile and can be applied to a variety of inverse problems using the same underlying network architecture.

In [1], the authors demonstrated that classical MLPs with ReLU activation functions tend to prioritize learning low frequencies, which can limit their ability to accurately capture fine details. To address this issue, several strategies have been proposed in the literature. These include mapping the input to a higher-dimensional space using Fourier mappings or employing alternative activation functions, such as periodic sinusoidal functions or Gabor wavelets.

This work is divided into two main parts. The first part surveys existing methods, providing descriptions and mathematical analyses of recent approaches. The second part involves a comparative evaluation of these methods across various applications in 1D, 2D, and 3D, optimizing parameters for each architecture due to the sensitivity of INRs to parameter variations. This comparative analysis is motivated by the need for a comprehensive survey that evaluates and contrasts recent methods in this field.

The second part of this work also introduces a novel activation function inspired by generalized sampling theory. We propose the Walsh Implicit Neural Network (WIREN), which utilizes the inverse Fourier transform of Walsh functions as its activation function. This approach seeks to overcome the limitations of traditional activation functions by leveraging the unique properties of Walsh functions. We then compare the performance of WIREN with other state-of-the-art methods to assess its effectiveness.

Keywords: Neural network, activation function, signal processing, computer vision

Mathematics Subject Classification: 68T07

# Résumé

Les représentations implicites par réseaux de neurones ont récemment émergé comme un nouveau paradigme pour aborder les tâches de vision par ordinateur. Cette approche est basée sur la représentation des données comme des fonctions continues, contrairement aux méthodes traditionnelles basées sur des grilles. Les représentations implicites par réseaux de neurones (INRs) utilisent le pouvoir expressif des réseaux de neurones pour approximer ces signaux définis implicitement et continus. Plus précisément, le signal est paramétré par un perceptron multicouche (MLP) avec des couches linéaires alternées et des fonctions d'activation non linéaires élémentaires. Le MLP prend les coordonnées en entrée et approxime le signal à ces coordonnées.

La paramétrisation continue fournie par les INRs offre plusieurs avantages par rapport aux méthodes traditionnelles. Par exemple, les INRs sont nettement plus économies en mémoire que les approches basées sur des grilles, ce qui les rend bien adaptés aux problèmes de haute dimension, tels que la reconstruction en 2D et 3D. De plus, les INRs sont extrêmement polyvalents et peuvent être appliqués à une variété de problèmes inverses en utilisant la même architecture de réseau sous-jacente.

Dans [1], les auteurs ont démontré que les MLP classiques avec des fonctions d'activation ReLU tendent à privilégier l'apprentissage des basses fréquences, ce qui peut limiter leur capacité à capturer les détails fins. Pour remédier à ce problème, plusieurs stratégies ont été proposées dans la littérature. Celles-ci incluent la projection de l'entrée dans un espace de dimension plus élevée en utilisant des projections de Fourier ou l'emploi de fonctions d'activation alternatives, telles que des fonctions sinusoïdales périodiques ou des ondelettes de Gabor.

Ce travail est divisé en deux parties principales. La première partie passe en revue les méthodes existantes, fournissant des descriptions et des analyses mathématiques des approches récentes. La deuxième partie implique une évaluation comparative de ces méthodes à travers diverses applications en 1D, 2D et 3D, en optimisant les paramètres pour chaque architecture en raison de la sensibilité des INRs aux variations de paramètres. Cette analyse comparative est motivée par le besoin d'une enquête complète qui évalue et contraste les méthodes récentes dans ce domaine.

La deuxième partie de ce travail introduit également une nouvelle fonction d'activation inspirée par la théorie de l'échantillonnage généralisé. Nous proposons le Réseau de Neurones Implicite Walsh (WIREN), qui utilise la transformée de Fourier inverse des fonctions Walsh comme fonction d'activation. Cette approche cherche à surmonter les limites des fonctions d'activation traditionnelles en exploitant les propriétés uniques des fonctions Walsh. Nous comparons ensuite la performance de WIREN avec celle des autres méthodes de pointe pour évaluer son efficacité.

Mots-clés : Réseau de neurones, fonction d'activation, traitement du signal, vision par ordinateur

Classification des sujets mathématiques : 68T07

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>A survey of existing methods</b>	<b>7</b>
2.1	Notations . . . . .	7
2.2	Inverse problem . . . . .	7
2.3	Neural networks . . . . .	9
2.4	Problem formulation . . . . .	10
2.5	Positional encodings . . . . .	11
2.6	Activation functions . . . . .	12
2.6.1	SIREN . . . . .	12
2.6.2	Gaussian Network . . . . .	13
2.7	HOSC (Harmonic Oscillator Activation) . . . . .	13
2.8	Sinus Cardinal . . . . .	13
2.8.1	Multiplicative Filter Networks (MFNs) . . . . .	14
2.8.2	WIRE (Wavelet-Domain Interpolation ReLU) . . . . .	14
2.8.3	FINER(Flexible implicit neural representations) . . . . .	14
2.9	INCODE . . . . .	15
2.9.1	Trident . . . . .	15
2.10	Reparametrized training . . . . .	16
<b>3</b>	<b>WIREN(Walsh implicit representation networks)</b>	<b>16</b>
3.1	Failure of SIREN . . . . .	16
3.2	Walsh functions . . . . .	18
3.2.1	General sampling theory . . . . .	18
3.2.2	WIREN . . . . .	22
<b>4</b>	<b>Experiments</b>	<b>22</b>
4.1	Tasks . . . . .	22
4.1.1	CT reconstruction . . . . .	22
4.1.2	Image denoising . . . . .	24
4.1.3	Image super resolution . . . . .	24
4.1.4	Audio reconstruction . . . . .	25
4.1.5	Occupancy volume . . . . .	25
4.2	Results . . . . .	25
4.2.1	Audio reconstruction . . . . .	25
4.2.2	CT reconstruction . . . . .	26
4.2.3	Image denoising . . . . .	27
4.2.4	3D shape representation . . . . .	28
<b>5</b>	<b>Scientific conclusion</b>	<b>29</b>
<b>6</b>	<b>Personal Conclusion</b>	<b>29</b>



# 1 Introduction

Traditional methods for processing images, videos, or 3D shapes often rely on discretizing the data. While this approach can yield satisfactory results, it has several drawbacks

- Traditional discretization methods are not well-suited for high-dimensional data, as the computational cost increases significantly with the dimensionality of the data. Even in 3D, discretizing a slightly irregular space is not straightforward.
- These methods require a substantial amount of memory, which can be problematic for large-scale applications.

The introduction and ongoing development of Implicit Neural Representations (INR) have recently emerged as a new way to represent data. Unlike traditional methods, an INR uses a continuous (and generally differentiable) function to represent the data (or generally a quantity of interest based on the data). For example, instead of using a matrix  $A = (\rho_{i,j})$  to provide the RGB values for each pixel of an image, the values can be represented by a continuous function  $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ .

More concretely, we assume that there is a relationship between  $a$ , a function of interest (which we observe from the data), and a continuous INR function  $\psi$ , and possibly its derivatives:

$$\text{there exists a function } F : F(a(x), \psi(x), \nabla_x \psi(x), \dots) = 0$$

$\psi$  is called the INR function and is defined implicitly by this equation.

In practice, to find  $\psi$ , it is assumed to be parameterized by a neural network called a multi-layer perceptron (**F**).

The MLP is given by:

$$\mathbf{F}(x; \theta) = \phi_N \circ \phi_{N-1} \circ \dots \circ \phi_0(x; \theta)$$

where

$$\phi_i = \sigma(W_i x_i + b_i)$$

and  $\sigma$  is the activation function.

Several studies have demonstrated the capability of Implicit Neural Representations (INRs) to accurately represent signals while capturing fine details. Initially, INRs were introduced for novel view synthesis in graphics [A][2] and for shape representation [3]. The classical use of the ReLU activation function often resulted in suboptimal performance across many applications. To address this issue, various approaches have been proposed to enhance performance.

One notable advancement was introduced by [4], who proposed a Fourier mapping technique that projects the network input into a higher-dimensional space. Building on this, [5] presented a new architecture where the output of each layer is multiplied by a Gabor wavelet. Further researchers have introduced various activation functions. These include periodic sinusoidal functions [6], time-frequency localized Gabor wavelets [7], Gaussian functions [8], and the FINER network [9].

Additionally, Trident [10] is a network that integrates both positional encoding and a carefully chosen activation function. Furthermore, reparametrized learning techniques[12] have been employed to adjust the weights and mitigate bias, further enhancing the network’s performance. This work is divided into three main sections.

1. Survey of Existing Methods: The first section provides a comprehensive survey of various methods in Implicit Neural Representations (INRs). It aims to represent these approaches, discussing their mathematical properties and the motivations behind them.
2. Introduction of WIREN: In the second section, motivated by general sampling theory, we introduce WIREN (Walsh Implicit Neural Network), which utilizes the inverse Fourier transform as an activation function.
3. Comparative Analysis: The final section presents a comparative analysis of different approaches across various tasks, evaluating their performance and effectiveness.

## 2 A survey of existing methods

### 2.1 Notations

Throughout this report, we work in a Hilbert space  $E$  equipped with a scalar product  $\langle \cdot, \cdot \rangle$ . Additionally, we consider the Hilbert space  $L^2(E)$ , which consists of square-integrable functions over  $E$ , with the scalar product defined by

$$\langle f, g \rangle = \int_E f(x)g^*(x) dx,$$

where  $f$  and  $g$  are functions in  $L^2(E)$ , and  $g^*(x)$  denotes the complex conjugate of  $g(x)$ .

We define the Fourier transform of a function  $f \in L^2(E)$  as

$$\hat{f}(\omega) = \int_E f(x) \exp(-j\omega \cdot x) dx,$$

where  $\omega \in E$  and  $j$  is the imaginary unit.

The inverse Fourier transform is given by

$$\check{f}(x) = \int_E \hat{f}(\omega) \exp(j\omega \cdot x) d\omega.$$

### 2.2 Inverse problem

Mathematical models and data are essential in contemporary science and engineering, serving as the foundation for understanding complex systems and making informed decisions. The study of inverse problems bridges these two domains, focusing on how model parameters

can be inferred from observed data and exploring the feasibility of such inferences. Mathematically, an inverse problem can be formulated as the problem of finding a solution  $x \in \mathbb{E}$  to the equation:

$$Ax = Y \quad (1)$$

Here,  $A : \mathbb{E} \rightarrow \mathbb{F}$  is an operator modeling the system's interaction, and  $y$  represents the observed data.

However, solving inverse problems is often highly challenging, primarily because many real-world problems are ill-posed, meaning they do not satisfy the Hadamard conditions.

**Definition 1** *A problem is well-posed if the following three properties hold:*

1. *There exists at least one solution  $x$ .*
2. *The solution is unique.*
3. *The solution depends continuously on the data, i.e., the operator  $A$  is continuous.*

A problem that does not satisfy one or more of these conditions is called an ill-posed problem. In such cases, small perturbations in the data can lead to large errors in the solution, or the solution may not exist or be unique.

For example, consider the image denoising problem, where the operator is given by  $A(x) = x + \epsilon$ , with  $\epsilon$  representing noise, typically modeled as a random variable. While this might appear to be an ill-posed problem due to the random nature of the noise, it is actually well-posed because a solution exists, is unique, and depends continuously on the data, even in the presence of noise. Many approaches have been proposed in the literature to solve inverse problems, or at least to approximate a solution. One of the prominent methods is the variational approach:

- **Variational Approach:** Instead of directly solving the equation  $Ax = y$ , we consider solving the associated minimization problem:

$$x = \arg \min\{x \in F \mid \|Ax - y\| + \lambda R(x)\} \quad (2)$$

where  $F$  is a subspace of  $E$ ,  $\|\cdot\|$  is a norm on  $E$ , typically the  $L_2$  norm, and  $R(x)$  is a regularization term that promotes stability and prevents overfitting by incorporating prior knowledge or assumptions about the solution.

- **Bayesian Method:** This approach treats the solution  $x$  as a random variable and incorporates prior information through a probability distribution. The solution is obtained by maximizing the posterior distribution:

$$p(x|y) \propto p(y|x) \cdot p(x) \quad (3)$$

where  $p(y|x)$  is the likelihood function representing the probability of observing  $y$  given  $x$ , and  $p(x)$  is the prior distribution that encodes prior knowledge about  $x$ . The posterior distribution  $p(x|y)$  combines the prior and the likelihood, and the solution can be obtained using techniques like Maximum A Posteriori (MAP) estimation or sampling methods such as Markov Chain Monte Carlo (MCMC).

- **Discrete Grid-Based Methods:** These methods discretize the domain into a grid or mesh and approximate the inverse problem using finite differences, finite elements, or finite volumes. The problem is then solved on this discrete grid:

$$Ax = y \quad (4)$$

where  $A$  is a matrix representing the discretized operator,  $x$  is the grid-based solution vector, and  $y$  is the observed data. Regularization techniques or multigrid methods are often employed to stabilize the solution and accelerate convergence.

This last approach, which relies on discretizing the operator, can yield satisfactory results if the problem is simple and the dimension is low. However, it has several drawbacks:

- Traditional discretization methods are not well-suited for high-dimensional data, as the computational cost increases significantly with dimensionality. Even in 3D, discretizing a slightly irregular space is not straightforward.
- These methods require a substantial amount of memory, which can be problematic for large-scale applications.
- These methods struggle to perform well when the observed data lies on an irregular set or domain.

In contrast to this approach, which relies on discretizing the operator, implicit neural representations have recently emerged as a general framework for solving inverse problems.

### 2.3 Neural networks

Neural networks are a subset of machine learning. Their structure is inspired by the human brain, mimicking the way biological neurons communicate with each other. The networks are architected in the form of layers of interconnected neurons that transmit information to one another.

The idea is that complex phenomena can be modeled by decomposing them into simpler phenomena. The function  $f$  on which we perform learning can therefore be composed of  $n$  simpler functions, i.e.,  $f = f_1 \circ \dots \circ f_n$ . What makes this method particularly effective is the ability to use gradient backpropagation. This technique allows for the rapid and easy calculation of the gradients of the loss function by applying the chain rule.

In general, each layer  $f_i$  is the combination of a linear function and a nonlinear function  $a$ , called the activation function:  $f_i(x) = a(w \cdot x + b)$ . The most used and studied activation function is the ReLU function given by :  $\text{ReLU}(x) = \max(0, x)$

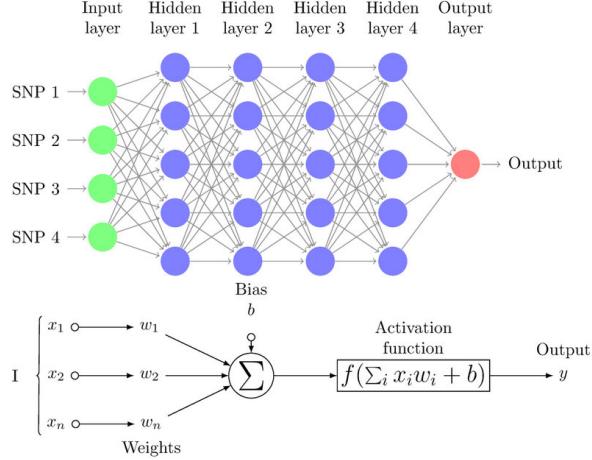


Figure 1: Multilayer perceptron

## 2.4 Problem formulation

Implicit neural representations address the problem of finding continuous representations for data. Given an input  $x$ , we aim to learn a function that maps  $x$  to a quantity of interest  $a(x)$ , while satisfying an implicit equation that depends on  $a(x)$ , its derivatives, and possibly higher-order derivatives:

$$F(x, a(x), \nabla_x a(x), \nabla_x^2 a(x), \dots) = 0 \quad (5)$$

Here, the function of interest could be, for example, an image where the coordinates map to pixel values.

To achieve this, we parametrize  $a(x)$  as a Multi-Layer Perceptron (MLP) network with  $N$  layers. The output of each layer is given by:

$$y_i = \sigma(W_i y_{i-1} + b_i) \quad (6)$$

where  $\sigma$  is the activation function,  $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$  are the weights, and  $b_i \in \mathbb{R}^{n_i}$  are the biases of the  $i$ -th layer. The input coordinate  $x_0$  represents the pixel coordinates for an image. The final output is given by:

$$y_M = W_N y_{N-1} + b_N \quad (7)$$

Implicit neural representations combine elements of both variational and iterative methods. Specifically, a neural network seeks to approximate the operator as a continuous function  $f_\theta$  that satisfies:

$$\theta^* = \arg \min_{\theta \in \Theta} \|f_\theta(x) - y\| \quad (8)$$

where  $\Theta$  is the set of possible parameter values, and  $\theta = \{(W_i)_i, (b_i)_i\}$  represents the parameters of the network. This optimization problem is solved iteratively using stochastic gradient descent.

ReLU (Rectified Linear Unit) has been extensively used for function approximation in MLPs. Although ReLU networks have shown excellent performance across various applications, they often struggle to represent complex signals and capture fine details effectively. This limitation occurs because these networks tend to prioritize learning low-frequency components, as demonstrated in previous studies[12]

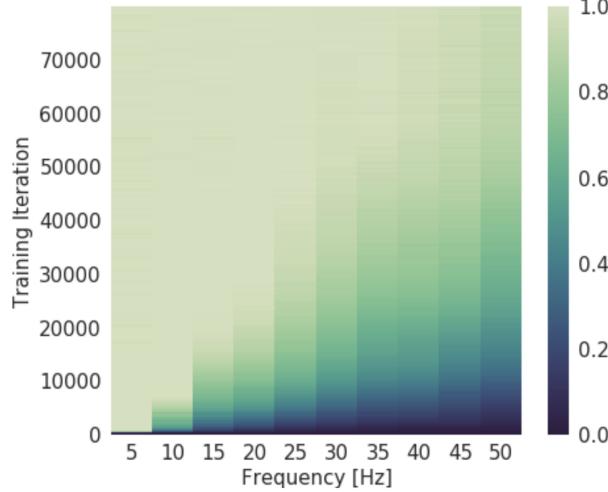


Figure 2: Evolution of the spectrum during training[12]

Consider the problem of approximating the function  $f(x) = \sum_{i=1}^{10} \sin(2\pi k_i z)$ , where  $k_i = (5, 10, \dots, 45, 50)$ , using an MLP with a ReLU activation function. Figure 2, taken from previous work, shows the evolution of the frequency spectrum during training. It illustrates that low frequencies are learned quickly, while the network struggles to learn higher frequencies within a reasonable number of iterations. Consequently, ReLU networks find it challenging to learn high-frequency data, such as images. This phenomenon is known as the spectral bias of neural networks. Numerous approaches have been proposed in the literature to mitigate this bias.

## 2.5 Positional encodings

One approach to learn high-frequency features involves mapping the input coordinates to a higher-dimensional space using a Fourier feature mapping  $\gamma$ . This enables the network to effectively represent these features, addressing limitations in capturing fine details. Several encoders have been proposed in the literature, including:[4]

- **Basic:**  $\gamma(x) = [\cos(2\pi x), \sin(2\pi x)]^T$  where  $\sigma$  is the frequency hyperparameter.
- **Positional Encodings:**  $\gamma(x) = [x, \dots, \cos(2\pi\sigma^{j/m}x), \sin(2\pi\sigma^{j/m}x)]^T$ , where  $\sigma$  represents the frequency hyperparameter and  $m$  the embedding size.

- **Random Fourier Features:**  $\gamma(x) = [\cos(2\pi Bx), \sin(2\pi Bx)]^T$ , with  $B$  being a random Gaussian matrix sampled from  $\mathcal{N}(0, \sigma^2)$ .

After applying the Fourier feature mapping to the input, the resultant features are fed into a Multi-Layer Perceptron (MLP) using ReLU activation functions.

## 2.6 Activation functions

A second approach to bypass the low frequencies bias is to choose a different activation function than ReLu. Its non continuos nature as well as its bias to low freuqencies make it unable to learn frequency features. On the other hand, functions as sigmoid or tanh are not complex enough. Many different activation functions were proposed in the litterature :

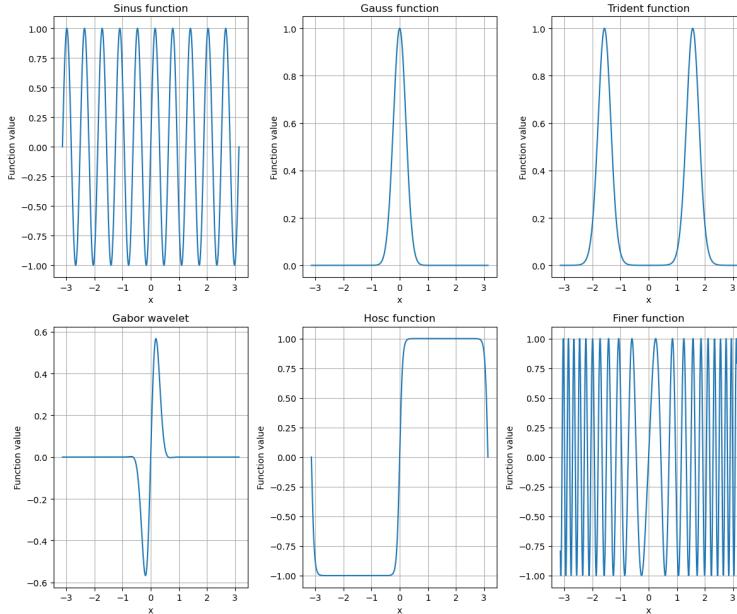


Figure 3: Plot of different acitvation functions

### 2.6.1 SIREN

The authors in [6] proposed SIREN using the sine function as an activation function :

$$\sigma(x) = \sin(\omega_0 x)$$

where  $\omega_0$  is a scaling hyper parameter chosen individually for each task and that aims to modulate the low frequency bias. In addition, an initialization procedure have been proposed by the others, where the idea is to preserve the distribution of function throught the network. Given that the output of the first layer is distributed as an arcsin, then chosing an adequate initialization ensures that the input of each layer is  $N(0, 1)$  and the

output is distributed as an arcsin. The authors propose to initialize weights distributed as  $U(-\sqrt{\frac{6}{n}}, \sqrt{\frac{6}{n}})$ . The periodic nature of the sine function allows it to represent a wide range of frequencies. Additionally, the smoothness of the sine function enables explicit calculation of derivatives, which can be used to solve inverse problems such as differential equations. The authors also propose an initialization scheme to preserve the distribution of activation functions in the network, improving both accuracy and convergence speed. While SIREN can represent high frequencies to some degree while being inexpensive memory-wise, it often fails to faithfully represent fine details due to the simple nature of the network.

### 2.6.2 Gaussian Network

In [8], the network uses the Gaussian function given by:

$$\sigma(x) = \exp(-\sigma x^2)$$

as an activation function. The activation function is smooth but does not have any periodicity, making it less adaptable to representing signals and high frequencies, but it is localized in the space domain and in the frequency domain.

## 2.7 HOSC (Harmonic Oscillator Activation)

Similar to the SIREN network, the authors in [18] proposed the HOSC function given by:

$$\sigma(x) = \tanh(\beta \sin(x))$$

where  $\beta$  is the sharpness factor controlling the effect of the hyperbolic function.

The HOSC function inherits properties from SIREN, as it is fast and inexpensive memory-wise. Moreover, higher values of  $\beta$  allow capturing fine details due to sudden changes in amplitude at  $x = n\pi$ , while lower values of  $\beta$  capture low frequencies similar to SIREN. The authors proposed a network where the sharpness factor increases through layers, allowing the network to represent both low and high frequencies. Additionally, they proposed Ada-HOSC as an alternative where  $\beta$  is a learnable parameter, leveraging the simple derivative form of HOSC.

## 2.8 Sinus Cardinal

Motivated by the classic Shannon sampling theorem and drawing similarities between INR and sampling, the authors in [17] proposed the sinus cardinal as an activation function:

$$\sigma(x) = \text{sinc}(\omega x)$$

The sinus cardinal is localized in the space domain but not in the frequency domain. It also forms a generating system, meaning the set of its translates  $\{x \rightarrow \text{sinc}(x - k) | k \in \mathbb{Z}\}$  can be used for sampling, enabling the network to approximate signals and represent high-order features effectively.

### 2.8.1 Multiplicative Filter Networks (MFNs)

An architecture proposed in [5] that replaces the MLP architecture with successive Hadamard products and nonlinearities:

$$\begin{aligned} y_1 &= g(x; \theta^1) \\ y_i &= (W_i y_{i-1} + b_i) \circ g(x; \theta^{i+1}) \\ y_M &= W_M y_{M-1} + b_M \end{aligned}$$

Where  $\circ$  denotes the Hadamard product

Two functions were proposed for  $g$ :

1. The sinus nonlinearity:

$$g(x; \theta^i) = \sin(\omega^i x + \phi^i)$$

2. The Gabor wavelet:

$$g(x; \theta^i) = \exp(-\gamma^i \|x - \mu^i\|_2^2) \sin(\omega^i x + \phi^i)$$

The authors proved that for both the Gabor and Fourier networks, the output can be represented as a linear combination of Gabor or Fourier filters. Despite their simple structure, these Modified Fourier Networks (MFNs) are able to surpass the performance of SIREN and FFN in many applications and represent high-order details effectively.

### 2.8.2 WIRE (Wavelet-Domain Interpolation ReLU)

In [7], the authors proposed to use the Gabor wavelet as an activation function:

$$\sigma(x) = \exp(-\sigma x^2 + i\omega_0 x)$$

The Gabor wavelet minimizes the product of its standard deviations in the time and frequency domains, which means it can represent both frequency and space features. Additionally, the function is localized in both space and frequency, combining strengths of the SIREN network and Gaussian network.

### 2.8.3 FINER(Flexible implicit neural representations)

The Siren model struggles to represent a broad spectrum of frequencies due to its use of a fixed scaling value. This limitation results in a constrained range of reconstructed frequencies, which may be inadequate, particularly when the frequency distribution of the signal is unknown. To address this issue, the authors in [9] proposed using the activation function given by:

$$\sigma(x) = \sin((|x| + 1)x)$$

In this new approach, the scaling parameter varies throughout the network based on the bias of different nodes. Higher bias values correspond to rapidly varying functions that can

capture high frequencies, while lower bias values correspond to slower variations, representing lower frequencies. This dynamic scaling provides the FINER approach with greater flexibility to represent a broader range of frequencies. Accordingly, the authors to initialize the bias coefficients as :

$$b \sim U(-k, k)$$

Taking a large value of  $k$  such as the set of frequencies won't be limited by the initialization.

## 2.9 INCODE

Another approach [13]that enhances the SIREN architecture involves using an activation function of the form:

$$\sigma(x) = a \sin(b\omega_0 x + c) + d,$$

where:

- $a$  controls the amplitude, influencing the overall strength of the activation function,
- $b$  controls the frequency scaling, determining the range of frequencies the network can learn,
- $c$  is the phase shift, adjusting the horizontal displacement of the waveform,
- $d$  is the vertical shift, affecting the baseline or "brightness" of the signal.

These four parameters ( $a$ ,  $b$ ,  $c$ , and  $d$ ) are dynamically predicted by a separate MLP, referred to as the harmonizer network. This allows the model to adapt its activation function parameters in real-time, leading to more flexible and expressive representations.

### 2.9.1 Trident

Trident[10]is a mix between the two approaches of using an appropriate activation function, and applying a Fourier mapping, the network is given by :

$$y_1 = W_1 \gamma(x) + b_1$$

Where  $\gamma$  is the Fourier features mapping  $\gamma(x) = x, \dots, \cos(2\pi\sigma^{j/m}x), \sin(2\pi\sigma^{j/m}x)]^T$ , and then the output of the others layers is :

$$y_i = \sigma(W_i y_{i-1} + b_i)$$

Where  $\sigma(x) = \exp(-s_0 x^2)$  is the activation function, the function is used to verify three proprieties. Being able to represent high order features, which is assured since we can write :

$$\exp(-\cos(x)^2) = \sum_{n=1}^{+\infty} A_n \cos(2x)^n$$

Secondly, the function is able to represent frequency features. This is assured since we can write the activation function as a serie of shifted and scaled cosine functions, spanning a wide range of frequencies since we have this formula :

$$\cos(x)^n = \frac{1}{2^n} \binom{n}{\frac{n}{2}} + \frac{2}{2^n} \sum_{k=0}^{\frac{n}{2}} \frac{n}{2} \binom{n}{\frac{n}{2}-k} \cos((n-2k)x)$$

And finally, the function is compactly localized in space, by choising the coefficients  $A_n$  to have a Gaussian window .

## 2.10 Reparametrized training

Another approach[11] to bypass the low frequency bias is to use an appropriate weight reparametrization of weights. The idea is to reparametrize each row of the weight matrix of the i-th layer  $W_i$  as a sum of Fourier bases. The output of each layer is given by :

$$y_i = \sigma(\Lambda_i B_i y_{i-1} + b_i)$$

Where  $\Lambda_i$  is a learnable weight matrix and  $B_i \in \mathbb{R}^{M \cdot n_{i-1}}$  is the Fourier basis matrix given by

$$B_i^{k,l} = \cos(w_k z_l + \phi_k), \text{ for } i = 1, \dots, M \text{ and } j = 1, \dots, n_{out}$$

Where  $B_i^{k,l}$  are the entries of  $B_i$ ,  $(z_l)$  is the sampling position sequence and  $M$  is the number of Fourier basis considered.

The authors proposed to choose  $P$  phases as an array of phase shifts, which are evenly distributed over the interval from  $[0, 2\pi]$ , and to take  $2F$  frequencies, consisting of a low frequency basis  $\{\frac{1}{F}, \frac{2}{F}, \dots, 1\}$  and a high frequency basis  $\{1, 2, \dots, F\}$ , which gives a total of  $M = 2FP$  bases.  $F$  and  $P$  are hyper parameters and should be chosen for each task independently.

As for the sampling sequence, the number of sampling points is the number of features, while the points were chosen to be sampled uniformly from the interval  $[-\pi F, \pi F]$ . Finally, it was proposed to draw the weights of the learnable matrix  $\Lambda_i$  following the distribution :

$$\Lambda_i^{k,l} \sim \mathcal{U}\left(-\sqrt{\frac{6}{M \sum_{t=1}^{d_{n-1}} (B_i^{l,t})^2}}, \sqrt{\frac{6}{M \sum_{t=1}^{d_{n-1}} (B_i^{l,t})^2}}\right)$$

## 3 WIREN(Walsh implicit representation networks)

### 3.1 Failure of SIREN

SIREN (Sinusoidal Representation Networks) was one of the first implicit neural representations ever introduced. It achieved state-of-the-art performance compared to traditional methods. However, SIREN did not perform as well on some tasks, particularly those involving image processing.

To understand why SIREN might fail to accurately represent certain frequencies, we can draw parallels between the problem of representing images using implicit neural representations (INRs) and a classical problem in image processing: the sampling problem. This problem is addressed by the classical Shannon sampling theorem. We recall the theorem[14] as follows:

A continuous band-limited signal  $x(t)$  can be reconstructed from its samples using the interpolation formula:

$$x(t) = \sum_{n \in \mathbb{Z}} x(nT) \cdot \text{sinc}\left(\frac{x}{T} - k\right), \quad (9)$$

where  $T$  is the sampling period. For accurate reconstruction,  $T$  must satisfy the condition  $w_{\max} \leq \frac{1}{2T}$ , where  $w_{\max}$  is the maximum frequency present in  $x(t)$ . Following the same approach as in [14], we consider the more general problem of sampling by replacing the sinc function with an arbitrary function  $\phi$ . Specifically, we investigate the problem of finding a function  $\phi$  such that the set  $V(\phi) = \{\sum_{n \in \mathbb{Z}} x(n) \cdot \phi(x - n) | x(n) \in l_2\}$  spans the entire Hilbert space. This problem aims to express a continuous signal as a weighted sum of scaled and shifted functions, which is conceptually similar to what a Multi-Layer Perceptron (MLP) does. In fact, a one-layer MLP can be expressed as:

$$f_\theta(x) = \sum_{i=1}^K a_i \sigma(w_i x + b_i), \quad (10)$$

where  $\sigma$  is the activation function.

Following this intuition, we derive the following theorem:

**Theorem 3.1** *If a signal can be reconstructed using the sampling formula with an arbitrary function  $\phi$ , then a two-layer neural network can approximate the signal as closely as desired.*

**Proof 3.1** 1. Consider the representation:

$$x(t) = \sum_{n \in \mathbb{Z}} x(nT) \cdot \phi\left(\frac{t}{T} - n\right). \quad (11)$$

Let  $\epsilon > 0$  and choose  $A > 0$  such that:

$$\|x(t) - \sum_{n=-A}^A x(nT) \cdot \phi\left(\frac{t}{T} - n\right)\|_\infty < \epsilon. \quad (12)$$

Define a two-layer network with:

- First layer weights  $W_1 = (\frac{1}{T}, \frac{1}{T}, \dots, \frac{1}{T})$  and biases  $b_1 = (-A, \dots, 0, \dots, A)$ .
- Second layer weights  $W_2 = (x(-AT), \dots, x(AT))$  and biases  $b_2 = 0$ .

By construction, the network output  $f_\theta(x)$  will approximate  $x(t)$  as close as desired. Hence:

$$|x(t) - f_\theta(x)|_\infty \leq \epsilon. \quad (13)$$

In particular, this implies that the sampling problem has greater expressive power than the implicit neural network problem. Therefore, if a function cannot be reconstructed using the sampling formula, it cannot be approximated by a neural network based on this framework.

To understand SIREN more, we solve the sampling problem for the sin function.

**Theorem 3.2** *Let  $f$  be a bounded function such that the set  $V(f)$  is of finite dimension. Then, there exist an integer  $r$ , 1-periodic functions  $\phi_i$  and  $\psi_i$ , and a set of frequencies  $(w_i)$  such that:*

$$f(x) = \sum_{i=1}^r (\phi_i(x) \cos(w_i x) + \psi_i(x) \sin(w_i x)). \quad (14)$$

**Proof 3.2** See Annexe

This theorem implies in particular that the set of functions that can be represented sin activation function has a finite dimension and as a consequence it cannot capture all frequencies. Interestingly, a similar but stronger result can be found in [15], where the authors found describe the explicitly the set of functions recoverd by the SIREN network.

The sampling theorem suggests that we could use the sinc function as an activation function in neural networks. However, like the Gaussian activation function, the sinc function lacks periodicity and does not possess the frequency compactness needed to efficiently represent frequency features. Can we design an activation function that addresses the sampling problem while retaining the properties of the SIREN ?

## 3.2 Walsh functions

### 3.2.1 General sampling theory

In [5], the author outlines two key requirements for a function to be used as a sampling basis. The first requirement is that the representation (as a sampling series) must be stable. Specifically, this stability means that a perturbation in the coefficients leads to a controlled perturbation in the reconstructed signal. Mathematically, this is expressed by requiring that the family of functions  $\{\phi_k = \phi(x - k) \mid k \in \mathbb{Z}\}$  must form a Riesz basis.

For a family of functions  $(\phi_k)_k$  to form a Riesz basis, there must exist two strictly positive constants  $0 < A \leq B < +\infty$  such that:

$$\forall (c(k))_k \in l_2, \quad A \cdot \|(c(k))_k\|_{l_2}^2 \leq \left\| \sum_{k \in \mathbb{Z}} c(k) \phi_k \right\|^2 \leq B \cdot \|(c(k))_k\|_{l_2}^2$$

Where  $l_2$  denotes  $L2$  norm on the space of sequences  $l_2$

In particular this implies that if  $\sum_{k \in \mathbb{Z}} c(k) \phi_k = 0$  then  $c(k) = 0$ , which means the set

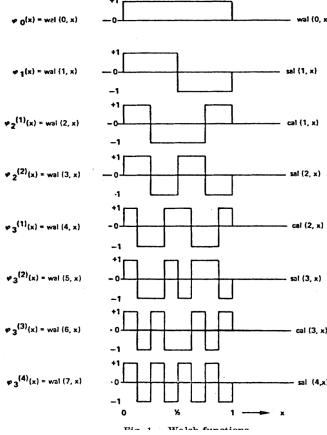


Figure 4: The first 8 Walsh functions[7]

$\{\phi_k = \phi(x - k) \mid k \in \mathbb{Z}\}$  is linearly independent. In addition, the upper bound implies that the space  $V(\phi)$  is well defined and is a subspace of  $L^2$ . The second requirement is that given a sufficiently small sampling step, we can approximate any signal as closely as desired. As shown in [5], this is equivalent to the partition of unity condition :

$$\forall x \in E, \sum_{k \in \mathbb{Z}} \phi(x - k) = 1$$

A function that verify both conditions is called an admissible generating function. subsectionWalsh functions For  $k \in \mathbb{N}$ , the  $k$ -th Rademacher function  $r_k : [0, 1] \rightarrow \{-1, 1\}$  is defined as:

$$r_k(t) = \text{sgn}(\sin(2^{k+1}\pi t)),$$

where  $\text{sgn}(x)$  is the sign function.

Given  $n \in \mathbb{N}$ , let  $n$  be expressed in binary as  $n = \sum_{j=0}^{+\infty} k_j 2^j$ , where  $k_j \in \{0, 1\}$  are the binary coefficients of  $n$ . The  $n$ -th Walsh function  $wal_n(t)$  is then defined as:

$$wal_n(t) = \prod_{j=0}^{+\infty} r_j(t)^{k_j},$$

Walsh functions were introduced by Joseph L. Walsh [16] in 1923 and have since become widely utilized in digital signal processing. Similar to sine and cosine functions, Walsh functions form a complete orthonormal set in Hilbert space. This orthonormal basis property makes them particularly useful for various applications in signal processing, image compression, and communications. Additionally, Walsh functions facilitate the definition of the discrete Walsh transform, analogous to the Fourier transform. In this context, the coefficients are given by:

$$c[k] = \int_0^1 wal_k(t) f(t) dt,$$

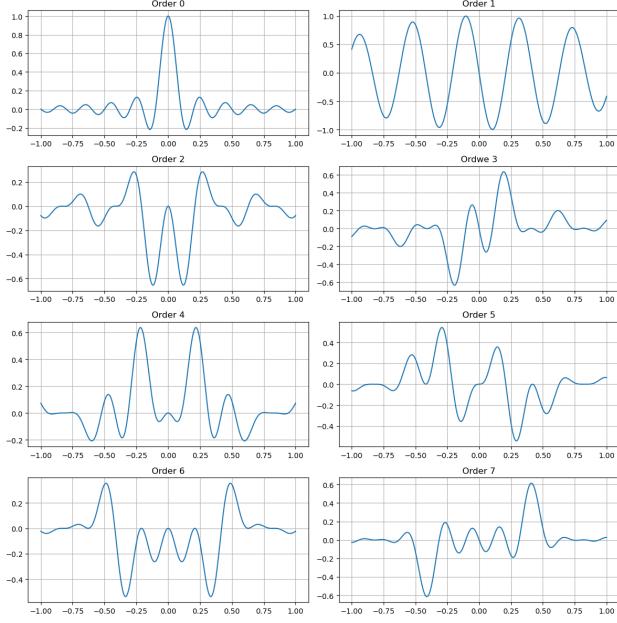


Figure 5: Plot of the Fourier transform of Walsh for order 0 through 7, we represent the imaginary part if the real part is nul

In [16], the author computed the inverse Fourier transform of the Walsh function, or equivalently the Walsh transform of the sinus function and gave a general form. First we define the Gray code as an ordering of the binary system such that two consecutive numbers differs in one number, more concretely

**Definition 2** Let  $n$  be an integer, and  $k_m k_{m-1} \dots k_0$  its binary decomposition, with  $k_n$  beign the most significant bit, we define the gray code  $g_m g_{m-1} \dots g_0$  by :

$$g_n = k_n$$

$$g_i = b_{i+1} \oplus b_i$$

Where  $\oplus$  is the xor operation, that equals 1 if  $b_i$  and  $b_{i+1}$  are different and 0 if they are the same bit

For example, the binary representation of 4 is 100, and we find the gray code is 110, which means that  $G(4) = 6$  We can now give the inverse Fourier transform of Walsh functions

**Theorem 3.3 [16]** The inverse Fourier transform of the  $n$ -th Walsh function is :

$$W_n(\omega) = \mathcal{F}^{-1}(wal_n)(\omega) = (-j)^\alpha \left[ \prod_{k=1}^m \cos\left(\frac{\omega}{2^{k+1}} - \frac{g_k}{2}\right) \right] \text{sinc}\left(\frac{\omega}{2^M}\right)$$

Here  $M$  is the number of bits in the binary presentation of  $n$ ,  $g_k$  is the  $k$ -bit in the Gray code representation of  $n$  and  $\alpha$  is the number of bits equal to 1 in the Gray code representation.

From now on, we will note this functions FWalsh functions. FWalsh are admissible generating functions We prove here that the FWalsh functions of even order are all admissible generating function and as a consequence can be used for sampling and are a possible candidate to be used as an activation function.

**Theorem 3.4** *For all  $n \in \mathbb{N}$ , the family  $W_n(x - k) | k \in \mathbb{Z}\}$  is an orthonormal basis*

**Proof 3.3** *Let  $k$  and  $l$  be in  $\mathbb{Z}$ , and note $_n(x - k) = \phi_k$ : By Parseval theorem, we have :*

$$\langle \phi_k, \phi_l \rangle = \int_{-\infty}^{+\infty} W_n^{-1}(x - k) \cdot_n^{-1} (x - l) dx = \int_{-\infty}^{+} wal_n(x - k) \cdot wal_n(x - l) dx$$

*But since  $W_n$  is nul outside of  $[0, 1]$  then the expression is nul if  $k \neq l$ . Otherwise :*

$$\langle \phi_k, \phi_k \rangle = \int_{-\infty}^{+\infty} wal_n(x - l) \cdot wal_n(x - l) dx = \int_l^{l+1} wal_n(x - l)^2 dx = 1$$

*As Walsh function take its values in  $[0, 1]$ , Which proves that*

$$\langle \phi_k, \phi_k \rangle = \delta_{k,l}$$

Since being orthonormal implies in particular that the sequence is a riesz sequence with  $A = B = 1$ , we conclude the first requirement. For the other requirement, we have the theorem

**Theorem 3.5** *For all  $n \in \mathbb{N}$  and  $x \in H$  we have :*

$$\sum_{k \in \mathbb{Z}} \phi(x - k) = 1$$

**Proof 3.4** *The Poisson formula is given by[9] :*

$$\sum_{k \in \mathbb{Z}} g(k) = \sum_{m \in \mathbb{Z}} \hat{g}(2\pi m)$$

*Fix  $x \in E$  and Let's apply the Poisson summation formula to the function  $g(y) = W_n(x - y) \exp^{-\omega_0 x}$ , whose Fourier transform is  $\hat{g}(\omega) = \exp(-j\omega x) \hat{W}_n(\omega) = \exp(-j\omega x) wal_n(\omega)$  Which yields :*

$$\sum_{k \in \mathbb{Z}} W_n(x - k) = \sum_{k \in \mathbb{Z}} \exp(-j2\pi\omega x) wal_n(2\pi\omega)$$

*But by the definiton of the Walsh function, the second term is :*

$$\sum_{n \in \mathbb{Z}} \exp(-j2\pi\omega n) wal_n(2\pi\omega n) = wal_n(0)$$

*But  $wal_n(0) = 1$  if  $n$  is even and  $-1$  if it is odd. As a result, the FWalsh functions verify the partition of unity condition when  $n$  is even*

We have now a set of functions that could be used as an activation function in INRs

### 3.2.2 WIREN

The sinus cardinalis (sinc function) is the zero-order Walsh function, so the approach in the previous paragraph generalizes the use of the sinc function as an activation function. We can extend this approach to use any even-order Walsh function, with functions of order higher than 0 (other than the sinc function) allowing the network to retain the properties of a SIREN (Sinusoidal Representation Network) architecture. This enables the network to capture high-frequency details due to the oscillatory behavior of trigonometric functions.

We define the Walsh4 (Walsh Implicit Representation Network) as a Multi-Layer Perceptron (MLP) that uses the fourth-order Walsh function as its activation function, given by:

$$\sigma(x) = -\cos\left(\frac{\omega_0 x}{2}\right) \sin\left(\frac{\omega_0 x}{4}\right) \sin\left(\frac{\omega_0 x}{8}\right) \text{sinc}\left(\frac{\sigma_0 x}{8}\right)$$

We also consider Walsh6, where we chose as an activation function, the Walsh function of order 6 given by :

$$\sigma(x) = -\sin\left(\frac{\omega_0 x}{2}\right) \cos\left(\frac{\omega_0 x}{4}\right) \sin\left(\frac{\omega_0 x}{8}\right) \text{sinc}\left(\frac{\sigma_0 x}{8}\right)$$

Here,  $\sigma_0$  and  $\omega_0$  are hyperparameters that can be tuned for each experiment, with default values of  $\omega_0 = 30$  and  $\sigma_0 = 10$ .

## 4 Experiments

### 4.1 Tasks

#### 4.1.1 CT reconstruction

CT reconstruction is a process used in medical imaging to create detailed cross-sectional images of the body from multiple X-ray projections taken at different angles. In radiography, only a two dimensional projection of a patient is possible, which is insufficient to capture all the informations about the patient as he resides in a 3d dimensional space. The idea of ct reconstruction is to solve this problem by combining the information from projections from different directions, to formulate the notion of a projected tomography mathematically we use the Radon transform :

**Definition 3** Let  $f$  be a continuous, we define the Radon transform  $R_f$  as an application defined on the space of straight lines  $L \in \mathbb{R}^2$  by the integration of  $f$  over the line  $L$  :

$$R_f(L) = \int_L f(x) dx$$

We can parametrize any straight line in terms of the angle  $\theta$  from which were take the projection and  $t$  the point where the line crosses the detector :

$$L(\theta, t) = \{x \in \mathbb{R} | x_1 \cos(\theta) + x_2 \sin(\theta) = t\}$$

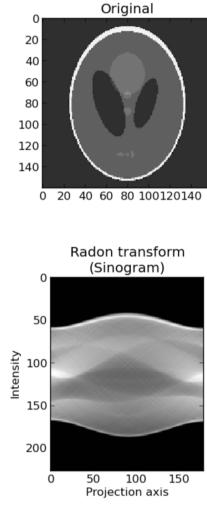


Figure 6: Example of a sinogram taken from : [https://tonysyu.github.io/scikit-image/auto\\_examples/plot\\_radon.html](https://tonysyu.github.io/scikit-image/auto_examples/plot_radon.html)

We define then the radon transform as an application on  $\mathbb{R}^2$

$$R_f(\theta, t) = \int_{L(\theta, t)} f(x) dx$$

We can represent the radon transform as an image that is called the sinogram Classical way to solve it Classically, this problem is solved using the Fourier Slice Theorem, which states that if  $u$  is given by  $f(\theta, \cdot) = R_\theta u$  where  $R_\theta u(t) = R_u(\theta, t)$  then :

$$\hat{f}(\theta, \omega) = \hat{w}\hat{v}$$

Where  $v = (\cos(\theta), \sin(\theta))$ . This theorem relies the Radon transform to the Fourier transform, in which is easier to solve inverse problems. One can show that[10]

$$R^* R u(x) = \int_{\mathbb{R}^2} \frac{u(y)}{\|x - y\|_2} dy$$

, and then apply a high-pass filter  $H$  whose Fourier transform is given by  $\hat{h}(\omega) = |\omega|$  and finally shows that  $R^* H R = I$  solving the problem. In order to reconstruct  $I$ , we had to multiply by  $|\omega|$  which means that the inverse of  $R$  is unbounded and that the problem is ill-posed. An implicit neural representation (INR) can reconstruct an image using a given number of measurements by minimizing the following loss function:

$$\mathcal{L} = \|sinogram(output) - sinogram(truth)\|$$

For all methods, we use a 2-layer network with 300 hidden features across all layers and ran it for 5000 iterations. For evaluating the model, we use the Peak-Signal-to-Noise ratio (PSNR) metric given by :

$$PSNR = -\log\left(\frac{\|u - f\|_2}{\|f\|_{+\infty}}\right)$$

### 4.1.2 Image denoising

Image denoising is an important task in image processing, that consists of removing unwanted noise and artefacts from an image while preserving its essential details and features. The most basic mathematical model is additive noise. For an image  $U$ , and noise  $\epsilon$ , the measured data is  $F$  is expressed as :

$$f = u + \epsilon$$

Where  $\epsilon$  is usually an additive Gaussian noise but we can also have a uniform distribution. There is also a multiplicative model, where we have the equation :

$$f = u\epsilon$$

Usually  $\epsilon$  follows a Gamma distribution. There is also a Poisson noise that is common in medical applications that is neither additive or multiplicative. A common case is when we can write the equation as :

$$f = (h * u) + \epsilon$$

Where  $h$  is a known kernel. In this case, an inverse filtering method is used : we express the equation in Fourier domain as the convolution became a simple multiplication :

$$F = HU +$$

and we recover an approximation  $v$  of  $u$  by :

$$v = \frac{\check{F}}{H}$$

Since we divide by  $H$  in the Fourier domain, this makes the problem ill-posed.

An implicit neural representation (INR) can reconstruct an image using a given number of measurements by minimizing the following loss function:

$$\mathcal{L} = ||model\_output - noisy\_image||$$

For all methods, we use a 2-layer network with 256 hidden features across all layers and ran it for 2000 iterations.

We also use the PSNR metric for the image denoising task

### 4.1.3 Image super resolution

Image super-resolution is a crucial task in image processing that involves enhancing the resolution of a low-resolution image by reconstructing its FINER details and producing a high-resolution output. The goal is to infer the missing high-frequency information that is not present in the low-resolution image. Hence it's a good task to compare the performance of different INRs since their main challenge is to learn the high frequencies.

An implicit neural representation (INR) can reconstruct an image using a given number of measurements by minimizing the following loss function:

$$\mathcal{L} = \|\text{model\_output} - \text{low\_resolution\_image}\|$$

For all methods, we use a 2-layer network with 256 hidden features across all layers and ran it for 2000 iterations.

We also use the PSNR metric for the super resolution task.

#### 4.1.4 Audio reconstruction

Audio reconstruction is the task that aims to restore or recreate an audio signal from incomplete, noisy, or degraded recordings. An implicit neural representation (INR) can reconstruct an audio using a given number of measurements by minimizing the following loss function:

$$\mathcal{L} = \|\text{model\_output} - \text{waveform\_audio}\|$$

For all methods, we use a 2-layer network with 256 hidden features across all layers.

We use the L2 loss as a comparaison metric for the audio reconstruction task.

#### 4.1.5 Occupancy volume

Occupancy volume reconstruction is a process used in fields like 3D modeling, robotics, and computer vision to create a detailed representation of a 3D space or environment. This process involves determining which specific regions within a given space are occupied and which are empty, enabling the accurate reconstruction of the entire volume.

An implicit neural representation (INR) can reconstruct an 3d shape using a given number of measurements by minimizing the following loss function:

$$\mathcal{L} = \|\text{model\_output} - \text{ground\_truth}\|$$

For all methods, we use a 2-layer network with 256 hidden features across all layers. and for comparaison we use the IoU metric which is defined by :

$$\text{IoU} = \frac{\text{Area}(\text{prediction} \cap \text{ground-truth})}{\text{Area}(\text{prediction} \cup \text{ground-truth})}$$

## 4.2 Results

We represent the hyperparameters used for each model in a table :

#### 4.2.1 Audio reconstruction

For audio reconstruction, we use a low learning rate,  $1e - 4$  to  $5e - 4$  for all architectures. In addition, following the same methon in [6], we set a large frequency scaling parameter ( $\omega_0 = 300.0$ ) in the first layer : And the table of the metrics is given by : In audio re-

Method	SIREN	WIRE	Gauss	FINER	INCODE	MFN	Re+fr	Walsh	ReLU+PosEnc
Learning rate	1e-3	5e-3	1e-3	5e-4	5e-3	5e-2	5e-4	5e-3	5e-4
$\omega_0$	30	10	—	30	—	—	—	30	—
$\sigma_0$	—	10	10	—	—	—	—	10	—

Table 1: Hyper parameters

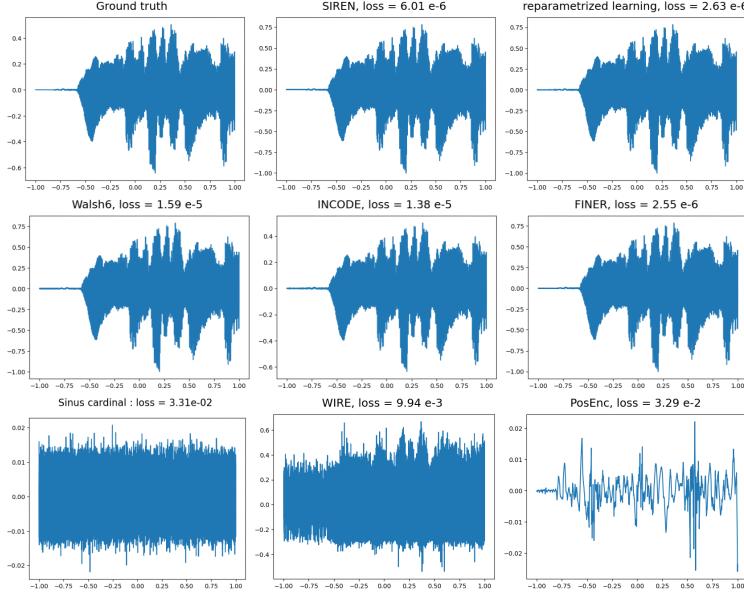


Figure 7: Results of the audio reconstruction task

Method	Re+sin	FINER	SIREN	INCODE	Walsh6	WIRE	PosEnc	Gauss	Sinc
PSNR $\uparrow$	2.63e-6	2.55e-6	6.06e-6	1.38e-5	1.59e-5	9.94e-3	3.29e-2	2.88e-2	3.31e-2

Table 2: Audio reconstruction metrics

construction, SIREN performs exceptionally well due to the highly periodic nature of audio signals. The sinusoidal functions with large frequency scaling parameters  $\omega_0$  used in SIREN are particularly effective at capturing the intricate details of these periodic signals. FINER performs the best due to the inherited properties of SIREN and the dynamic scaling parameter. Walsh6 performs relatively well due to its periodic components which motivates the choice of walsh over sinus cardinal function

#### 4.2.2 CT reconstruction

The results of the experiment are:

Here 're + sin' is the reparametrized learning architecture with sinus as an activation function.

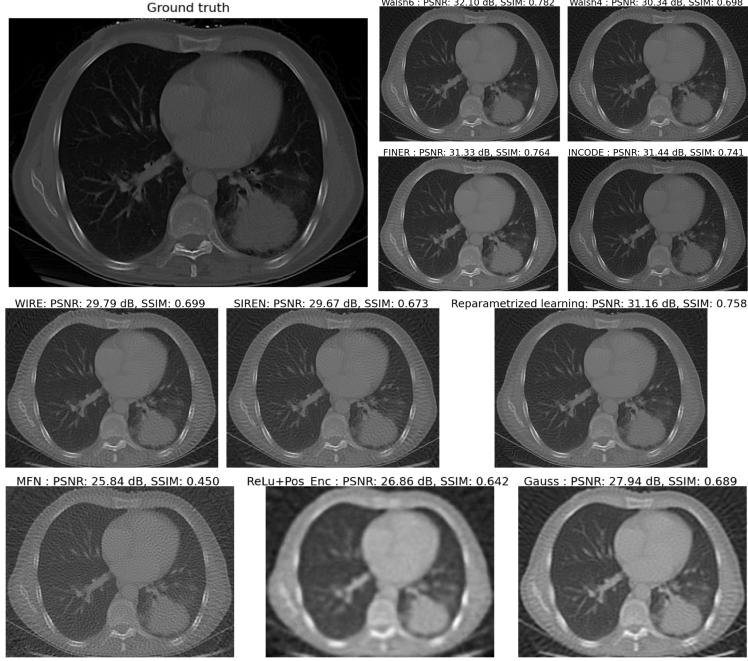


Figure 8: The result of CT reconstruction, the 6-th order Walsh give the best result

Method	Walsh6	FINER	INCODE	Walsh4	Re+sin	WIRE	Gauss	SIREN	PosEnc	MFN
PSNR $\uparrow$	32.10	31.33	31.44	30.34	31.16	29.79	27.94	29.67	26.86	25.84
SSIM $\uparrow$	0.782	0.764	0.741	0.698	0.758	0.699	0.689	0.673	0.642	0.450

Table 3: CT reconstruction metrics

For CT (Computed Tomography) representation, SIREN demonstrates excellent performance because it approximates the Radon transform, which involves integrating over lines and planes at various angles. This can be effectively parameterized using trigonometric functions. Consequently, this characteristic is naturally inherited by both the Walsh function and the FINER function, which achieve the best results in this context. Additionally, WIREN (Walsh Implicit Neural Network) also performs well due to its enhanced capability to represent high frequencies.

On the other hand, positional encoding, despite utilizing Fourier mapping, struggles to capture high frequencies effectively. Finally, Gauss and MFN (Multi-Scale Fourier Networks) deliver acceptable performance but face challenges due to the lack of a periodic component.

#### 4.2.3 Image denoising

In the image denoising task, we focus on removing noise originating from sensor measurements. This noise is composed of two primary components: photon noise, which is modeled by a Poisson distribution, and readout noise, which is modeled by a Gaussian distribution. The overall noise affecting the image is modeled as the sum of these two components.



Figure 9: The result of CT reconstruction, the 6-th order Walsh give the best result

The results of the image denoising task are:

Method	INCODE	Re+sin	FINER	Walsh6	Walsh4	MFN	SIREN	Gauss	WIRE	PosEnc
PSNR $\uparrow$	29.63	29.47	29.05	28.79	28.56	28.22	28.60	28.09	28.74	27.16

Table 4: Image denoising metrics

INCODE achieves the best performance for image denoising, primarily due to its frequency scale parameter  $b$ , which adapts effectively to the denoising task by reducing high-frequency noise. Similarly, FINER benefits from adaptive frequency scaling, with its network bias adjusting to the specific requirements of the task. In contrast, Wire and Walsh do not perform as good in this task context because they tend to learn high frequencies, which is not suitable for image denoising.

#### 4.2.4 3D shape representation

We represent here the results of the IoU metric on the 3D shape representation task : Among

Method	FINER	SIREN	re+sin	Walsh6	WIRE	Gauss
IoU $\uparrow$	0.9798	0.9789	0.9687	0.9676	0.9667	0.9661

Table 5: IoU for different approaches

the methods evaluated, FINER demonstrates the highest performance in fitting 3D shapes,

surpassing the other techniques. SIREN follows closely behind, showing strong results as well. The reparametrized learning method also performs effectively, though it trails behind FINER and SIREN. While walsh6 perform better than the Gabor wavelet

## 5 Scientific conclusion

In this work, we propose a novel implicit neural representation architecture that derives its expressive power from sampling theory. We compared the performance of WIREN to state-of-the-art approaches in implicit neural representations across multiple tasks. WIREN performed exceptionally well, achieving the highest accuracy in the CT reconstruction task and outperforming other activation function-based methods, such as SIREN and WIRE, in both image denoising and CT reconstruction tasks.

For future work, possible directions include further exploring the theoretical properties of the WIREN network, particularly its ability to mitigate biases that affect the performance of MLPs. Additionally, applying WIREN to new tasks, such as image super-resolution and video representation, could broaden its applicability. Finally, investigating an initialization scheme for the network’s weights and biases may lead to further performance improvements and increase its training speed.

## 6 Personal Conclusion

The internship has been a roller coaster of emotions, ranging from despair at being stuck without any clues to enjoying the beautiful city of Cambridge. It was a chance for me to have my first contact with research, to learn how to read papers and analyze them, and to gain first-hand experience of how challenging yet enjoyable research can be. I would like to thank Dr. Angelica Aviles Rivero for making this visit possible, and also thank Joy, Sam, and Zhongying for their valuable mentorship and advice. I would also like to thank the entire Math+ML+X group for the enjoyable lunches and the very useful seminars that exposed me to various research directions. I especially enjoyed the debate seminar and the ‘Sell Your Project’ seminar.

## References

- [1] Jeong Joon Park and Al. "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019.
- [2] Mildenhall, Ben and Al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: Computer Vision – ECCV 2020. Lecture Notes in Computer Science, vol 12346, 2020.
- [3] Lars Mescheder and Al. "Occupancy Networks: Learning 3D Reconstruction in Function Space" In: "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019.
- [4] Matthew Tancik and Al. "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains". In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020)
- [5] Rizal Fathony and Al. "Multiplicative Filter Networks" In: International Conference on Learning Representations (ICLR) 2021
- [6] Vincent Sitzmann and Al. "Implicit Neural Representations with Periodic Activation Functions". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2020
- [7] Vishwanath Saragadam and Al. "WIRE: Wavelet Implicit Neural Representations". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2023
- [8] Sameera Ramasinghe and Al. "Beyond Periodicity: Towards a Unifying Framework for Activations in Coordinate-MLPs". In: Proceedings of the European Conference on Computer Vision (ECCV) 2021
- [9] Liu, Zhen and Al. "FINER: Flexible spectral-bias tuning in Implicit NEural Representation by Variable-periodic Activation Functions". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2024
- [10] Angelica I Aviles-Rivero and Al, "TRIDENT: The Nonlinear Trilogy for Implicit Neural Representations". 2023
- [11] Kexuan Shi and Al. "Improved Implicit Neural Representation with Fourier Reparameterized Training". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

[12] Nasim Rahaman and Al. "On the Spectral Bias of Neural Networks". In: Proceedings of the 36th International Conference on Machine Learning 2019

[13] Amirhossein Kazerouni and Al. "INCODE: Implicit Neural Conditioning With Prior Knowledge Embeddings". In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2024

[14] M. Unser, "Sampling-50 years after Shannon". In: Proceedings of the IEEE, Volume: 88, Issue: 4 2000

[15] Yüce, Gizem and Al. "A structured dictionary perspective on implicit neural representations". In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)

[16] Karl H. Siemens and Al. "A Nonrecursive Equation for the Fourier Transform of a Walsh Function". In: IEEE Transactions on Electromagnetic Compatibility ( Volume: EMC-15, Issue: 2 1973

[17] Hemanth Saratchandran and Al. "A sampling theory perspective on activations for implicit neural representations". In : ICML 2024

[18] Danzel Serrano and Al. "HOSC: A Periodic Activation Function for Preserving Sharp Features in Implicit Neural Representations". In: <https://arxiv.org/abs/2401.10967>

## 7 Annexe

**Proof 7.1 (Theorem 3.2)** Let  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  be a bounded function such that  $V(\phi)$  is of finite dimension  $b$ .

Let  $(g_i)_{i=1,\dots,n}$  be a basis of  $V(\phi)$ , since each  $g_i$  is a finite linear combination of  $(\phi(x - k))_k$  then all  $g_i$  are bounded by boundness of  $\phi$ , it follows that all the function in  $V(\phi)$  are bounded. Define the operator  $T : V(\phi) \rightarrow V(\phi)$  by

$$\forall x \in V(\phi), T(g)(x) = g(x + 1)$$

Let  $\lambda$  be an eigenvalue of  $T$  and  $g$  an eigenvector, then we have that :

$$\forall x \in \mathbb{R} \forall n \in \mathbb{N} g(x + k) = \lambda^k g(x)$$

, since  $g$  is bounded and is not null in all of  $\mathbb{R}$  the necessirarily, we have

$$|\lambda| \leq 1$$

However  $T$  is invertible and its inverse is

$$T^{-1}(g)(x) = g(x - 1)$$

, it follows that  $\lambda^{-1}$  is an eigenvalue of  $T^{-1}$  and by the same reasoning,

$$|\lambda| \geq 1$$

Which means that all eigenvalues of  $T$  have a module equal to 1 We can write then  $Sp(T) = \{\exp^{i\omega_1}, \dots, \exp^{i\omega_m}\}$  the spectrum of  $T$ . We can write then

$$V(\phi) = \bigoplus_{k=1}^m \ker(\exp^{i\omega_k} Id - T)^{m_k}$$

Let  $k \in \{1, \dots, m\}$  and  $g \in \ker(\exp^{i\omega_k} Id - T)^2$ , this means that for all  $x \in \mathbb{R}$  and all  $p \in \mathbb{Z}$  :

$$\exp^{i2\omega_k} g(x + p) - 2\exp^{i\omega_k} g(x + p + 1) + g(x + p + 2) = 0$$

For a fixed  $x$ , This a linear recursive sequence of order 2 as such there exists  $A_0(x)$  and  $A_1(x)$  A such :

$$g(x + p) = (A_0(x) + pA_1(x)) \exp ip\omega_k$$

Since  $g$  is bounded then necessarily,  $A_1(x) = 0$  which implies in particular for  $p = 1$  :

$$g(x + 1) = \exp i\omega_k g(x)$$

Hence,  $g \in \ker(\exp^{i\omega_k} Id - T)$ . Therefore  $\ker(\exp^{i\omega_k} Id - T)^2 = \ker(\exp^{i\omega_k} Id - T)$  and for all  $k$   $m_k = 1$  which means that  $T$  is diagonalizable. Let  $(g_k)_{k=1, \dots, n}$  be a basis of eigenvectors, and  $f \in V(\phi)$  then :

$$f = \sum_{k=1}^n \alpha_k g_k$$

For some complex numbers  $(\alpha_k)_k$ . If we write  $g_k(x) = h_k(x) \exp(i\omega_k x)$  then  $h_k$  is a 1-periodic function, we write

$$h_k(x) = \psi_k(x) + i\gamma_k(x)$$

and

$$\alpha_k = a_k + ib_k$$

By substituting and using the fact that  $f = Re(f)$  we find that

$$f(x) = \sum_{k=1}^n (a_k \psi_k(x) - b_k \gamma_k(x)) \cos(\omega_k x) - (b_k \psi_k(x) + a_k \gamma_k(x)) \sin(\omega_k x)$$

Which what we wanted to prove because the two functions multiplied by sin and cosine are both 1-periodic