signals

MDPI

Article

SpectraMelt: An Open-Source A2I Simulator

Peter Swartz [1,*], Saiyu Ren [1,*], Shuxia Sun [2] and James Martin [3]

[1] Department of Electrical Engineering, Wright State University, 3640 Colonel Glenn Hwy, Dayton, OH 45435, USA
[2] Department of Mathematics and Statistics, Wright State University, 3640 Colonel Glenn Hwy, Dayton, OH 45435, USA; shuxia.sun@wright.edu
[3] School of Electrical and Computer Engineering, University of Oklahoma, 660 Parrington Oval, Norman, OK 73019, USA; james.c.martin@ou.edu
* Correspondence: swartz.17@wright.edu (P.S.); saiyu.ren@wright.edu (S.R.)

Abstract

The Nyquist Folding Receiver is an architecture that uses Compressed Sensing to convert analog radio frequency signals into digital signals. Analog-to-Digital Converter architectures that implement Compressed Sensing are collectively known as Analog-to-Information. Sparse bandlimited analog signals with frequency bands above the Nyquist frequency of a traditional Analog-to-Digital Converter can be recovered by Analog-to-Information receivers. Recovery of these signals is affected by the selection of a Compressed Sensing recovery algorithm. Typical recovery algorithms selected for recovery of Nyquist Folding Receiver-compressed outputs use iterative methods to find the solution. This work presents a machine learning approach to signal reconstruction. The proposed method uses a neural network to learn the mapping from compressed samples to the original signal. The neural network is trained on a set of synthetic signals generated by a new open-source Analog-to-Information simulator called SpectraMelt. The results show that the neural network can effectively reconstruct the original signal from the compressed samples, achieving better performance than traditional iterative methods.

Keywords: analog-to-digital; analog-to-information; cognitive radio; compressed sensing; Nyquist folding receiver; radio frequency

1. Introduction

Modern information systems (ISs) increasingly demand acquisition rates that strain existing hardware, as Nyquist-rate sampling of high-bandwidth signals produces excessive data or imposes impractical capture requirements. The Analog-to-Digital Converter (ADC) is a central bottleneck in this process. A major shift came with Donoho's 2006 introduction of Compressed Sensing (CS) [1], which offered a framework to bypass Nyquist constraints. CS exploits the structure of finite-dimensional inner product spaces by employing the ℓ_1 norm to promote sparsity, enabling accurate reconstruction of certain signals from sub-Nyquist samples.

This breakthrough inspired extensive research into Analog-to-Information (A2I) converters (Figure 1) [2], architectures that embed CS principles directly into data acquisition. Notable A2I designs include the Random Demodulator (RD) [3], the Modulated Wideband Converter (MWC) [4], and the Nyquist Folding Receiver (NYFR) [5]. While none yet rival traditional ADCs due to performance trade-offs [6], the NYFR remains a leading candidate for sub-Nyquist sampling.

Check for updates

Academic Editor: Manuel Duarte Ortigueira

Received: 4 October 2025
Revised: 2 December 2025
Accepted: 22 January 2026
Published: 5 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

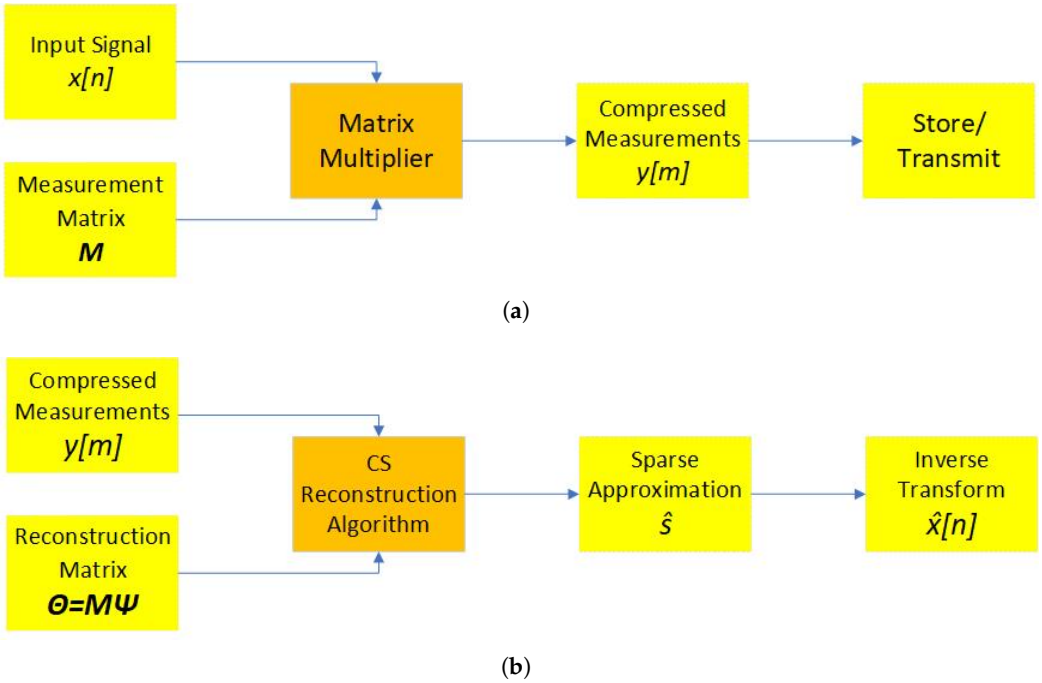


Figure 1. General A2I sampling architecture. (**a**) CS acquisition model. (**b**) CS reconstruction model.

Early modeling work [7] is extended here with a neural network-based reconstruction framework that improves both theoretical accuracy and empirical performance, significantly narrowing the gap between experimental feasibility and practical deployment by surpassing traditional iterative methods on real-world signals. Recent advances in deep learning for high-complexity sensing modalities, such as oriented feature extraction in SAR ship detection, demonstrate the broader trend toward neural models that exploit structured geometry in measurement data, a direction aligned with the reconstruction strategies explored in this work [8]. The neural network is enabled by an open-source A2I simulator called SpectraMelt. This new simulator allows for the creation of A2I digital twins that can generate synthetic data needed for the training of the network without the need for hardware implementation while promoting the development of new A2I architectures.

1.1. Compressed Sensing Background

While a full treatment of CS theory is beyond the scope of this work, a brief overview is provided to establish the mathematical foundations, recovery conditions, and assumptions that justify the signal model. In CS, a vector $\vec{x}$ of length N belongs to $F^N = \{(x_1, \ldots, x_N) : x_j \in F\}$, typically with $F = \mathbb{R}$. An inner product space defines the ℓ_2 norm as $||\vec{x}||_2 = \sqrt{\langle \vec{x}, \vec{x} \rangle}$, with the general ℓ_p norm given by

$$||\vec{x}||_p = \begin{cases} (\sum_{i=1}^{n} |x_i|^p)^{1/p} & \text{for } p \in [1, \infty) \\ \max\limits_{i=1,\ldots,n} |x_i| & \text{for } p = \infty \end{cases}$$

For $0 < p < 1$, these are *quasi-norms*. At $p = 0$, $||\vec{x}||_0 := |\operatorname{supp}(\vec{x})|$, where $\operatorname{supp}(\vec{x}) = \{i : x_i \neq 0\}$ denotes the indices of nonzero elements. ℓ_p norms measure approximation error between $\vec{x}$ and its estimate $\hat{x}$ as $||\vec{x} - \hat{x}||_p$. CS exploits sparsity, especially with the ℓ_1 norm, since errors concentrate in fewer coefficients. A signal $\vec{x} = x[n]$ of length N with at most K nonzeros ($K \ll N$) is K-sparse where $0 \leq n < N$. Many signals become sparse in Fourier $\mathcal{F}$ or Wavelet $\mathcal{W}$ bases. If $\vec{x} \in \mathbb{R}^N$ is sparse when transformed by a universal basis $\mathbf{\Psi}$, where $\mathbf{\Psi}\vec{x} = \vec{s}$ satisfies $||\vec{s}||_0 \leq K$, it is called *compressible*.

For a compressible $\vec{x} \in \mathbb{R}^N$, CS allows reconstruction from measurements $\vec{y} \in \mathbb{R}^P$:

$$\vec{y} = \mathbf{M}\vec{x} = \mathbf{M}\boldsymbol{\Psi}^{-1}\vec{s} = \boldsymbol{\Theta}\vec{s} \tag{1}$$

where P is the number of measurements, $\mathbf{M}$ is the measurement matrix, $\vec{s} \in \mathbb{R}^N$, and $K < P \ll N$. This holds when $\mathbf{M}$ contains Gaussian or Bernoulli random variables.

Since (1) is underdetermined, recovery seeks the sparsest $\hat{s}$ by solving

$$\hat{s} = \underset{\vec{s}}{\mathbf{argmin}} \, ||\vec{s}||_0 \quad \text{subject to} \quad \vec{y} = \boldsymbol{\Theta}\vec{s} \tag{2}$$

where **argmin** finds $\vec{s}$ minimizing $||\cdot||_0$. This NP-hard problem can be relaxed to a convex ℓ_1-minimization if $\mathbf{M}$ satisfies the Restricted Isometry Property (RIP), again true for Gaussian or Bernoulli cases.

1.2. The Nyquist Folding Receiver

The NYFR shown in Figure 2 folds RF subspaces using a modulated local oscillator (LO):

$$s_{\mathrm{LO}}(t) = \sin(\varphi_{s1}(t)) = \sin(2\pi f_{s1} t + \theta_{s1}(t)),$$

with angular frequency $\omega_{s1} = 2\pi f_{s1}$. The narrowband modulation satisfies $\max\left|\frac{d\theta_{s1}(t)}{dt}\right| \ll f_{s1}$ and is defined as

$$\theta_{s1}(t) = \frac{f_\Delta}{f_{mod}} \sin(2\pi f_{mod} t). \tag{3}$$

This sweeps the LO frequency across $f_{s1} \pm \Delta f$, requiring $f_{s1} = Q f_{mod}$, where Q is a positive integer to maintain RIP [9]. A pulse train $\Delta(t)$ from LO zero crossings can be expressed as

$$\Delta_{mod}(t) = p(t) * \frac{d\varphi(t)}{dt} \sum_k 2\pi\delta(\varphi(t) - 2\pi k), \tag{4}$$

where modulation disrupts uniform spacing. Multiplying by the wideband input $x_{\mathrm{WB}}(t)$ applies the Nyquist–Shannon theorem, folding components into the LO-defined baseband consistent with the Union of Subspaces (UoS) model [10].

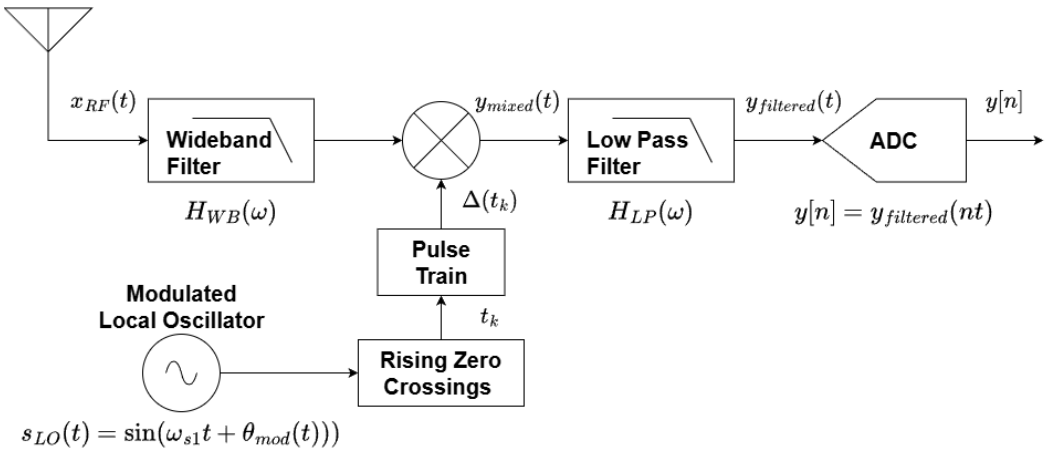


Figure 2. Nyquist Folding Receiver block diagram.

Mixing produces $y_{\mathrm{mixed}}(t)$ with baseband frequencies

$$f_{base} = |f_C - f_{s1} k_H|, \quad k_H = \mathrm{round}\left(\frac{f_C}{f_{s1}}\right).$$

A low-pass filter with cutoff

$$f_{cutoff} = \frac{f_{s1}}{2}$$

suppresses aliases, yielding $y_{\text{filtered}}(t)$ with induced modulation $M\theta_{s1}(t)$, where M is an integer scale factor. Finally, a conventional ADC samples at f_{s2}, near but not equal to the average rate f_{s1} of $\Delta_{mod}(t)$, producing $y[m] = y_{\text{filtered}}(mt)$ where $0 \leq m < P$. This last stage acts like time-domain decimation, multiplying by the ADC's sampling train $\delta_{T_s}(t)$ without introducing new modulation.

Once the compressed input signal is collected, a CS algorithm reconstructs the original signal from measurements, as shown in Figure 1b. Reconstruction accuracy depends on the reconstruction matrix $\mathbf{\Theta} = \mathbf{M\Psi^{-1}}$ as shown in Figure 3 [11], which serves as input to the CS recovery for the NYFR as shown in Figure 1 [2]. The NYFR sampling model specifies vector and matrix dimensions: P measurements by the ADC, and $y[m]$ by the measurement vector. The number of positive folds $Z = N/P$ relates Nyquist frequency f_{nyq} and LO frequency f_{s1} by $f_{nyq} = \frac{Z \cdot f_{s1}}{2}$, set by the wideband filter cutoff f_{cutoff}. For example, if $f_{cutoff} = f_{nyq} = 1\,\text{kHz}$ and $f_{s1} = 100\,\text{Hz}$, then $Z = 20$. The universal basis is Fourier $\mathbf{\Psi} = \mathcal{F}$, so $\mathbf{X} = \mathcal{F}(\vec{x})$. The inverse transform $\mathbf{\Psi}^{-1}$ is a block diagonal matrix with blocks D formed from sub-blocks U and L:

$$D = \begin{bmatrix} \mathbf{\Psi}_{+}^{-1} & 0 \\ 0 & \mathbf{\Psi}_{-}^{-1} \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & L \end{bmatrix} \tag{5}$$

where $\Psi_{m,n}^{-1} = e^{\frac{2\pi jmn}{P}}$.

$$\begin{bmatrix} y[0] \\ \vdots \\ y[P] \end{bmatrix} = \begin{bmatrix} I_P | \ldots | I_P \end{bmatrix} \begin{bmatrix} S_0 & & & & & & & \\ & S_1 & & & & & & \\ & & \ddots & & & & & \\ & & & S_Z & & & & \\ & & & & S_Z^* & & & \\ & & & & & \ddots & & \\ & & & & & & S_1^* & \\ & & & & & & & S_0^* \end{bmatrix} \begin{bmatrix} U & & & & \\ & L & & & \\ & & \ddots & & \\ & & & U & \\ & & & & L \end{bmatrix} \begin{bmatrix} s[0] \\ \vdots \\ s[N] \end{bmatrix}$$

$y[m]$	$\mathbf{R}$	$\mathbf{S}$	$\mathbf{\Psi}^{-1}$	$s[n]$
$P \times 1$	$P \times 2ZP$	$2ZP \times 2ZP$	$2ZP \times ZP$	$ZP \times 1$

Figure 3. Real valued sampling model for the NYFR.

The measurement matrix $\mathbf{M} = \mathbf{RS}$, where $\mathbf{R}$ projects Z NYFR zones onto the baseband of P samples, and $\mathbf{S}$ is a conjugate-symmetric diagonal matrix modulating each zone over time. Each sub-block $S_n = \vec{LO} \cdot I_P$, where $\vec{LO}$ contains sampled LO elements $LO_n = e^{jM\Theta(t_{adc})}$ at ADC sample times t_{adc}. The modulation index $M = 0, \pm 1, \pm 2, \ldots, \pm Z$ defines the original signal frequency and peak deviation MF_Δ, with the sign indicating the sideband and a 90° shift for negative modulation. Thus, $\mathbf{S}$ comprises two conjugate-symmetric blocks with the common time modulation pattern scaled by each zone's M value.

2. SpectraMelt

The promise of CS to recover sampled signals having bandwidths beyond the Nyquist frequency has led to extensive research on A2I architectures, with numerous designs proposed to reduce power consumption compared to traditional ADCs [12]. While current A2I implementations do offer measurable power savings, conventional ADC architectures still surpass them in terms of achievable sampling rates [6]. For A2I technology to move

beyond experimental use and achieve mainstream adoption, it will be essential to identify and refine the most capable designs. Modeling and simulation play a critical role in this process, enabling not only rigorous comparison of candidate architectures but also the exploration of hybrid approaches that combine the strengths of multiple methods. Furthermore, simulation provides a foundation for developing innovative signal recovery algorithms, ultimately helping close the performance gap between A2I converters and traditional ADCs. The NYFR was selected for simulation because of the compelling qualitative analysis performed here [13].

Simulation requires the application of models within a virtual RF signal environment. There are multiple platforms available for generating these virtual environments, with Matlab being a popular choice due to its flexibility and speed through the use of specialized toolboxes. Its widespread adoption is exemplified by its use in the analysis performed here [11]. However, Matlab requires a paid license, which can limit access and reproducibility across organizations, whereas open source alternatives are freely available and facilitate collaborative development. Because of these considerations, an open source model was selected for this work. Among existing options, GNURadio stands out as a mature framework widely used across research, industry, and academia for software-defined radio (SDR) applications and simulations. Built on Python, GNURadio offers a rich library of signal processing blocks enabling complex transmit and receive architectures. Yet, its simulated environment has limitations, such as timing mismatches causing spurious frequency signals and limited environmental control, which constrain detailed testing of A2I architectures.

To overcome these issues while leveraging Python's flexibility and compatibility with GNURadio, a custom RF simulator directly implemented in Python is proposed for greater environmental control and integration with neural network tools. This simulator began development here [7] with the creation of an NYFR digital twin to establish functionality. However, attempted reconstruction using several different CS recovery algorithms yielded inaccurate results. A refactoring of the code into more traditional object-oriented methods was needed before a root-cause analysis could be conducted. By refactoring the code, a more mature simulator, SpectraMelt v1, was developed in Python 3.12 and used for both data collection and analysis.

Unit testing against SpectraMelt followed the sequence diagram shown in Figure 4, which provides a clear visual overview of the testing workflow, illustrates module interactions, and allows readers to understand the code's behavior without presenting the actual source. Currently, all signal generation, including time domain input signals, is carried out within a single, monolithic, Python class. The next iteration of SpectraMelt will decompose this into multiple, independent classes. This modularity will promote ease of use, greater logging capabilities, and expansion of A2I architectures that can be simulated. Another benefit of this will be a reduction in the number of external libraries that will be needed to install SpectraMelt, relying almost solely on Numpy only, and the ability to install using pip. All data paths and software operations have been optimized to allow for OS-agnostic operation.

This testing, along with expert advice, led to the discovery that the frequency of the underlying simulator time was not sufficient to represent analog signals. Originally, input signals leaving the wideband filter were used as the simulator time base. By creating an 'analog' input signal using a vector with time steps much smaller than the one originally used for the output from the wideband filter, overall fidelity of SpectraMelt was improved. Figure 5 gives an example of how the input signal shape in the time domain more accurately represents an analog signal at this time scale versus the original version, highlighting the importance of selecting an appropriate simulator frequency.

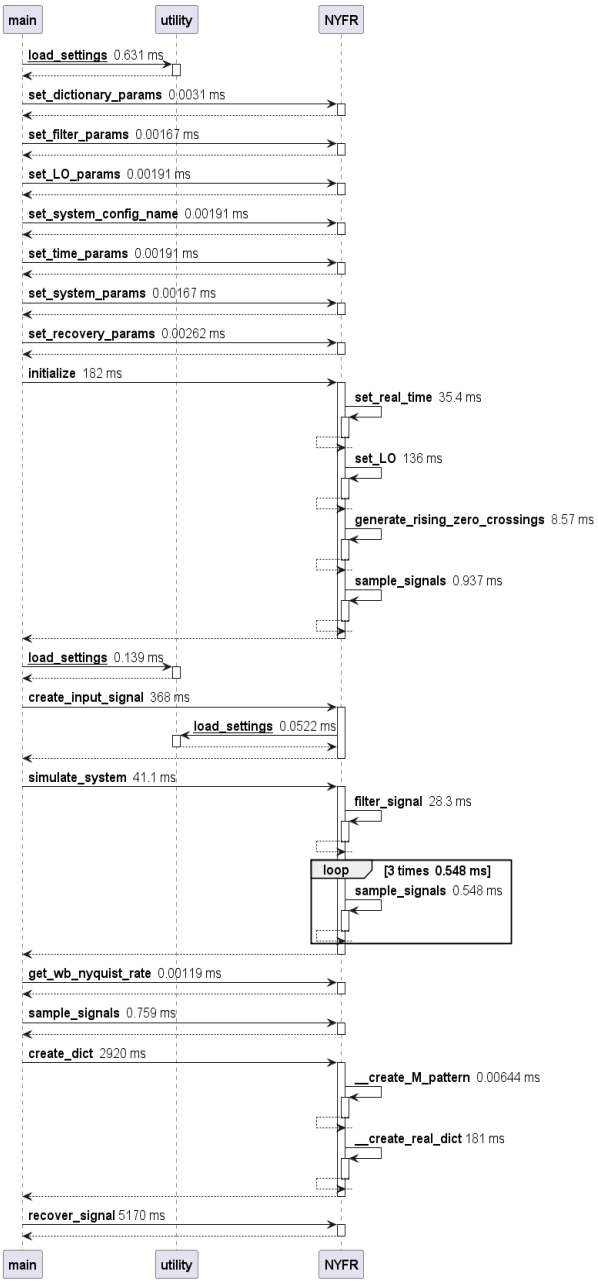


Figure 4. Nyquist Folding Receiver sequence diagram. Dashed arrows indicate return messages from function calls.

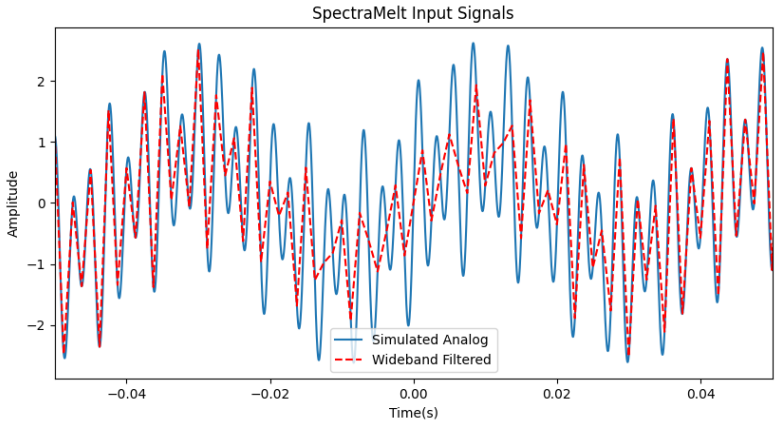


Figure 5. Simulated analog vs. wideband-filtered f_{sim} = 1 MHz vs. f_{wb} = 400 Hz input tones at 25 Hz, 255 Hz, and 395 Hz.

3. Improved A2I Recovery

Now that SpectraMelt is producing accurate compressed samples, the next step is to show how it can be leveraged to improve A2I designs. One way to do this is to produce a new CS recovery method within the simulator that exceeds current, state-of-the-art-practice, traditional iterative CS recovery methods in terms of speed and recovery accuracy. Machine learning (ML) is a signal processing technique that has exploded in popularity in recent years. The most successful ML technique has been the Artificial Neural Network (ANN) originally developed by psychologist Frank Rosenblatt in 1958 [14]. While ANNs have been applied to virtually every problem space thanks to the rise of Large Language Models (LLMs), only recently have they been applied to A2I. In 2023, a team from the University of Electronic Science and Technology of China used LLMs to try and classify the compressed output of the NYFR against a group of radar signal types [15].

The ANN chosen for implementation within SpectraMelt to decompress NYFR outputs is called the Multilayer Perceptron (MLP). An MLP is a simple feedforward network that maps fixed-size inputs to outputs, while an LLM is built on the transformer architecture with self-attention, enabling it to capture long-range dependencies and contextual relationships in variable-length sequences. The architecture of the MLP implemented here is modeled after a more advanced architecture inspired by sparse Bayesian learning (SBL) called Learned-SBL [16]. This new network contains an input layer, one hidden layer, and one output layer. Each layer is fully connected with the previous layer. The input layer contains the same number of inputs as the number of samples captured at the simulated output of the NYFR. The hidden and output layer perceptrons all contain the same number of outputs as the uncompressed input signal. A toy example with four nodes per layer is shown in Figure 6.

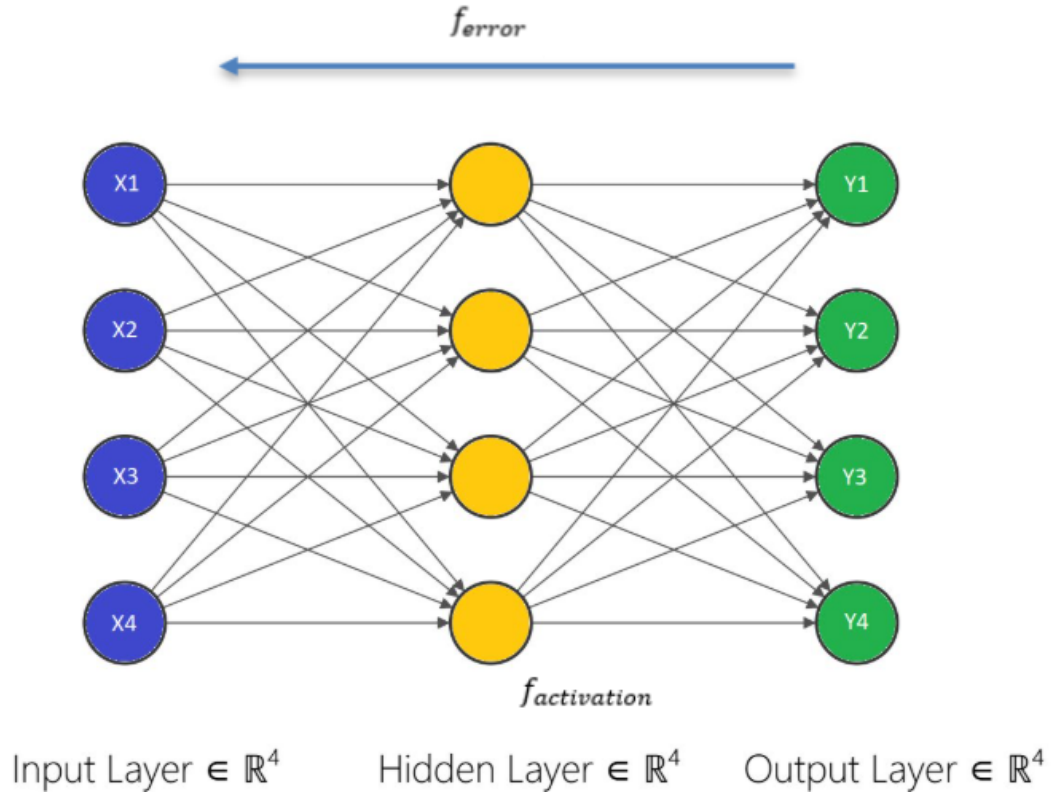


Figure 6. SpectraMelt MLP with N = 4.

Typical activation functions such as the Rectified Linear Unit (ReLU) are applied to perceptron outputs when the network is performing classification tasks, constraining the value between zero and one. Because the network uses unsupervised training, the output

from an MLP layer cannot be constrained this way, leading to the activation function $f_{activation}$ given by

$$f_{activation} = \sum_{i=1}^{n} x_n w_n \tag{6}$$

This means that the output from each layer is just a linear combination of the inputs from the previous layer.

Another consideration is the type of loss function used to train the network during back-propagation. Initially, a mean absolute error (MAE) loss function was selected as it calculates the average of the absolute differences between predictions and true values. This seemed in line with the traditional CS principle from Equation (2) where the recovery problem is relaxed to a convex ℓ_1-minimization problem by setting $p = 1$. In reality, this spread the error for the training over every output node, resulting in no signal recovery, only output noise. A custom root-mean-squared error (RMSE) function f_{error} given by

$$f_{error} = \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

was created for the network to obtain the desired results.

3.1. Dataset Creation

With a working MLP implementation in place, it was time to create datasets from the NYFR digital twin. These datasets would serve three purposes:

- Training data for the new MLP;
- Signals used for recovery to assess MLP performance versus other CS recovery algorithms;
- Assess the impact of varying NYFR LO parameters on data recovery.

The datasets were created as an extension of the simulator settings found here [7], with f_{sim} = 1 MHz, f_{wb_filt} = 400 Hz, f_{ADC} = 100 Hz, f_{LO} = 100 Hz, f_{lp_filt} = 50 Hz, $t_{start} = -2$ s, and t_{stop} = 2 s. This sets the number of positive Nyquist Zones $Z = 8$.

Each input dataset F_{input} was generated from a list of all the possible integer input frequencies that can exist within the wideband filter range. In this case, the possible frequencies $F_{possible} = (f_1, f_2, \cdots, f_{max})$, where $f_1 = 1$, $f_2 = 2, \cdots, f_{max} = 400$, giving the total number of possible frequencies $n_{possible} = 399$. Every F_{input} comprises input signals with tones drawn from $F_{possible}$ and taking n at a time with $n = (1.2), 3, 4, 5$, resulting in a total of four sets containing 79,800 signals per set. The individual tones within each input signal had random amplitudes between 0.5 and 1 taken from a uniform random variable. The input signals created in these initial datasets are idealized to show the functionality of SpectraMelt and quickly compare implemented recovery algorithms. This means that no noise is added to them and signal tones are expressed as linear measures of magnitudes instead of the typical dB.

Recovery algorithms for the NYFR are directly influenced by how the LO parameters are tuned, since these parameters govern the modulation index and resulting frequency zone definitions. Specifically, the modulation index M uniquely determines the mapping of signals into frequency zones, but it is only valid under the condition that the clock modulation $\theta(t)$ remains narrowband. This requires the maximum rate of change in $\theta(t)$ given by

$$\max\left|\frac{d\theta(t)}{dt}\right| = \max|2\pi f_\Delta \cos(2\pi f_{\text{mod}} t)| = 2\pi f_\Delta \tag{8}$$

to be much smaller than $2\pi f_{LO}$. In practice, this translates to at least two orders of magnitude of separation between the modulation deviation f_Δ and the local oscillator frequency f_{LO}. Furthermore, the system is typically constrained such that $f_{LO} = f_{ADC}$, tightly linking

the oscillator design to the sampling process and imposing additional structure on the recovery algorithms.

Selection of specific LO parameters within the current literature relies on Monte Carlo simulations to select appropriate values within the constraints given above [11]. This search was performed by first developing some qualitative analysis about LO parameter selection and then using recovery results from a single CS recovery algorithm. Understanding how to select appropriate LO parameters also requires understanding how the LO signal modulation affects the shape of the modulated pulse train $\Delta_{mod}(t)$ in both time and frequency domains using the analysis performed here [7]. The modulation frequency f_{mod} must be tuned to match the size of the zones created by the NYFR with some integer multiple of the width of the LO frequency domain lobes. The matching of lobe widths will reduce mutual coherence between zones, increasing recovery accuracy.

Beyond the analysis and constraints to the LO parameters just described, there is currently no prescription on which specific values to select for f_{mod} and f_{Δ} given f_{adc} and f_{wb_filter}, meaning that some Monte Carlo simulations are still required. On top of that, there has been little analysis to determine if these values are specific only to one selected CS recovery algorithm. SpectraMelt helps address these issues by generating F_{output} datasets for the NYFR with varying LO parameters of the digital twin. For the system parameters used to create each F_{input}, the following list of LO parameters was used to create 16 unique F_{output} datasets: $f_{mod} = 0.1, 0.2, 0.25, 0.5$, $f_{\Delta} = 0.1f_{mod}$, $0.8f_{mod}$, $1.2f_{mod}$, $10f_{mod}$. These LO parameters were also used to create CS recovery dictionaries needed for reconstruction based on Figure 3.

3.2. Compressed Data Recovery

Improvements to $\mathbf{\Theta}_{NYFR}$ based on comparisons between the actual output of the NYFR Digital Twin $\vec{y}_{NYFR}$ and the approximation given by $\hat{y}_{model} = \mathbf{\Theta}_{NYFR}\vec{x}$ was discussed in [7]. However, this approach is flawed. What is really needed is adjustments based on the initial guess $\widetilde{x} = \Theta^{\dagger}_{NYFR}\vec{y}_{NYFR}$, where $\Theta^{\dagger}_{NYFR}$ is the pseudoinverse of the NYFR recovery dictionary. Graphical examination of $\widetilde{x}$ applied to Digital Twin outputs shows that orientation mismatches are corrected using the pseudoinverse. The magnitude discrepancies are also corrected within iterative CS algorithms by normalizing the output $\widetilde{x} = \Theta^{\dagger}_{NYFR}(\vec{y}_{NYFR}/\vec{y}_{norm})$. To correct for magnitude mismatches within the MLP training data and improve reconstruction speed, $\widetilde{\mathbf{X}}$ is created by pre-multiplying the measurement matrix with a magnitude correction factor to give $\widetilde{x} = (f_{ADC}/4)\Theta^{\dagger}_{NYFR}\vec{y}_{NYFR}$. An example signal is shown in Figure 7.

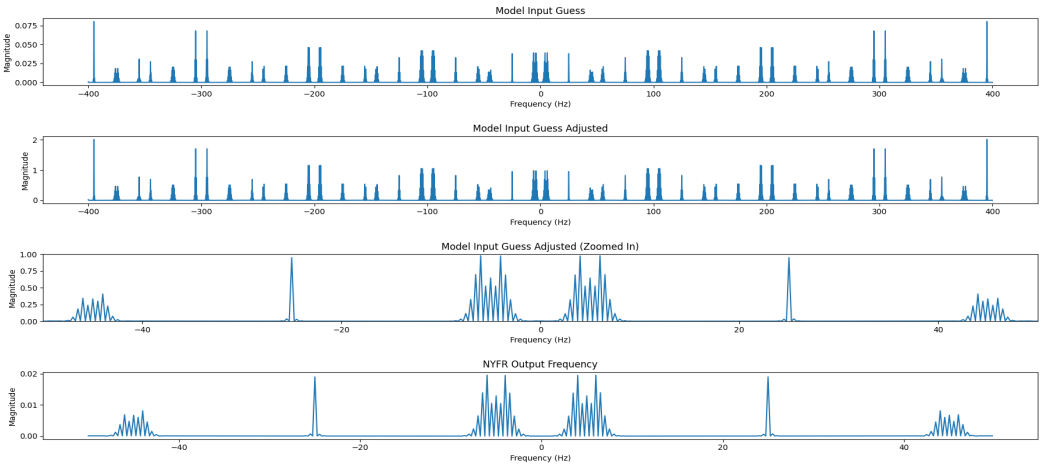


Figure 7. Initial CS recovery guess vs. NYFR outputs: frequency domain.

In this initial implementation, the magnitude correction factor was selected empirically as a quick, practical means of aligning the pre-multiplication signal with the original input waveform and ensuring that the resulting frequency-domain magnitudes remained within a small percentage of their true values. This ad hoc choice provided inputs of an appropriate scale for effective MLP training, but it is not a principled or generalizable solution. The correct approach is to normalize all input and output signals prior to training so that the network operates on consistent, statistically comparable input ranges. Full-signal normalization eliminates the need for manual correction factors and supports reproducibility across datasets, hardware configurations, and noise environments. Implementing a rigorous normalization pipeline is a priority for the next major revision of SpectraMelt, where this temporary workaround will be replaced with a theoretically grounded, systematically optimized preprocessing stage. This will also enable the use of more complex activation functions—such as Softmax or Sigmoid—that assume inputs and outputs confined to well-defined numerical ranges (e.g., [0, 1]), ensuring stable gradients and more reliable convergence during training.

MLPs are trained using supervised learning, where the network weights are optimized over many passes through the dataset to minimize a loss function that maps inputs to desired outputs. In the present work, training begins by splitting the dataset into training and test partitions. The network is then trained for with a maximum number of epochs set, with each epoch processing the training data in batches of a specified number of training signals. Batch-based optimization significantly reduces memory requirements and accelerates convergence compared to processing the entire dataset at once. During each epoch, the model shuffles the training samples to reduce overfitting and ensure that gradient updates are not biased by sample ordering. These design choices favor stable optimization and permit efficient use of compute hardware, allowing MLPs to learn complex nonlinear mappings that iterative sparse-recovery algorithms cannot represent.

To further control training cost and prevent unnecessary computation, early stopping is employed through a set of configurable parameters. The training process monitors a chosen metric—monitor, typically "val_loss"—and halts if improvements fall below a minimum threshold delta for a specified number of epochs. In typical training workflows, "val_loss" represents the model's error on a held-out validation set, providing an unbiased estimate of how well the network generalizes to unseen data and serving as the primary metric for guiding early stopping. Early-stopping checks are delayed until enough epochs have passed to establish a meaningful performance trend. Together, these mechanisms provide a principled way to limit training duration and computational expenditure, in contrast to iterative methods, which must solve a new optimization problem for each input instance. Once an MLP is trained, inference is effectively instantaneous, making the up-front training cost a one-time expense that yields substantial runtime speed advantages.

There is a need to systematically compare the performance of this new MLP network against the previous state-of-the-art iterative CS recovery methods. Orthogonal Matching Pursuit (OMP), Spectral Projected Gradient for L1 minimization (SPGL-1), and Iterative Hard Thresholding (IHT) were added to SpectraMelt as the state-of-the-practice methods because of their relative successes in CS recovery. The OMP algorithm used for reconstruction was modified from the original algorithm within the Scikit-Learn Python library to allow for complex valued dictionaries. A custom complex version of IHT was also created based on [17]. SPGL-1 reconstruction was implemented using the SPGL1 Python library found in the PyPi repository.

The input and output datasets from the previous section have been recovered using four separate CS recovery algorithms incorporated into SpectraMelt: IHT, OMP, SPGL1, and the newly trained MLP network. Several assumptions about the recovery results

should be mentioned before discussing the recovery results. One is that the IHT algorithm was given the correct number of unknown tones a priori. This was carried out for baseline results, as IHT is the oldest of the four algorithms. No other algorithm had this knowledge. Secondly, an individual MLP network was trained per signal set, for a total of four networks. These networks only recovered signals with the corresponding number of tones per signal. This was carried out because the training of the same MLP on different datasets caused catastrophic forgetting, a well-known limitation of MLPs when trained on tasks sequentially. Catastrophic forgetting refers to the tendency of neural networks to lose previously learned information when trained on new tasks. Various strategies, such as regularization or memory-based methods, have been proposed to mitigate this effect [18].

For each recovery set, several metrics were used to assess the quality of the recovered signals. In order for a signal tone to be considered recovered, it must first exactly match one of the input tone frequencies. If the magnitude exceeds half of the original tone's magnitude, then the signal tone is marked as recovered. If any recovered signal also exceeds this threshold but does not match an input tone frequency, it is considered a spur. The average magnitude error of the recovered signals is considered along with the average magnitude of the spurs. The recovery accuracy is found by dividing the number of recovered signals by the total number of input signals. The spur rate divides the total number of spurs by the total number of input signals. An example of signal recovery from each recovery method in SpectraMelt is shown in Figure 8.

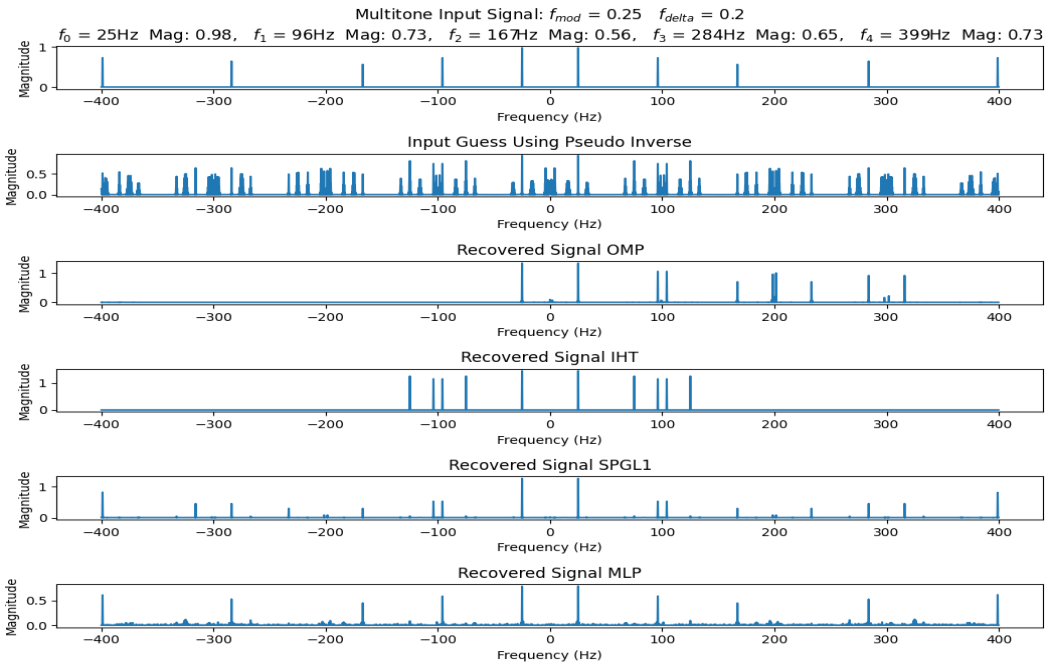


Figure 8. SpectraMelt-recovered NYFR outputs: frequency domain.

One of the broad questions that this work is trying to answer is "What are the specific NYFR LO settings that maximize reconstruction?" From the results obtained by the three iterative algorithms, no effect on signal recovery was seen by varying LO parameters. However, there were variations seen from the MLP recovery results which seemed to favor $f_{mod} = 0.25$. No accompanying variations were seen by different f_Δ values with the previous frequency across all datasets. It should be noted that the standard deviation in recovery accuracy between different datasets and LO parameters never exceeded 0.009 from the MLP recovery results. These results together seem to suggest that there is no preferable LO parameter setting as long as the previously stated constraints are held. This matches the previous analysis performed, showing that f_Δ and f_{mod} jointly constrain the system's dy-

namic reconstruction range, balancing signal recovery, zone identification, pulse resolution, and phase-noise limitations [11].

Table 1 shows the compiled results from the NYFR digital twin simulations using the datasets described in the previous two sections. All reported values are averages over 100 recovered signals. This table shows a clear distinction between the IHT and OMP algorithms. The a priori knowledge given to the IHT improved the average spur rate produced by the algorithm. It did not, however, improve the reconstruction accuracy, as this is the main measure of a selected CS algorithm's performance. This shows that the OMP is a better CS algorithm for the NYFR than the IHT. There is also a clear distinction between the two older algorithms, IHT and OMP, versus the newer two, SPGL1 and MLP, in terms of recovery accuracy. Of final note, the MLP network's recovery accuracy is on average about double that from SPGL1, 63.8% vs. 38.3% as computed from the values in Table 1, while almost completely eliminating spurs.

Table 1. Comparison of recovery methods across different tone counts.

Tones	Method	Rec. Acc.	Spur Rate	Mag. Err.	Spur Mag.
1–2	IHT	32.25%	58.75%	0.367	1.05
3	IHT	32.83%	65.83%	1.053	1.09
4	IHT	29.63%	70.38%	1.685	1.17
5	IHT	28.9%	71.10%	2.480	1.15
1–2	OMP	34.5%	191%	0.291	52.46
3	OMP	33.17%	146%	0.905	8124
4	OMP	31.5%	117%	1.784	131.47
5	OMP	32.3%	100%	2.721	185.59
1–2	SPGL1	50%	129%	0.574	0.825
3	SPGL1	40.3%	88.8%	1.32	0.725
4	SPGL1	33.5%	58.5%	2.14	0.730
5	SPGL1	29.4%	46%	2.933	0.670
1–2	MLP	78.3%	1%	0.83	0.723
3	MLP	68.7%	1%	1.58	0.763
4	MLP	56.4%	1.8%	2.31	0.629
5	MLP	51.9%	1.1%	3.11	0.660

4. Conclusions and Future Work

The previous analysis demonstrates the superiority of the new MLP-based recovery method over state-of-the-practice CS recovery techniques, while underscoring the importance of modeling and simulation in advancing A2I architectures. Digital twins enable the design of novel recovery algorithms and exploration of alternative A2I architectures. The open-source nature of SpectraMelt will allow researchers to rapidly prototype and evaluate new designs, accelerating adoption of A2I as NYFR recovery accuracy improves. The SpectraMelt code will be made publicly available in an online GitHub repository and was version-controlled using Git version 2.47.1.windows.2 to support reproducibility.

A limitation of the MLP approach is the potentially long training time, though most of this cost is concentrated in generating the pre-multiplication initial guess. Loading new dictionaries into a signal processing chain should be manageable, except when hardware implementation is required. In practice, alternative LO settings may not even be necessary. Techniques such as Learned-SBL could enable adaptive behavior, while dataset mixing strategies can mitigate catastrophic forgetting. In this work, the MLP is trained for a fixed tone count, and its performance is evaluated under that constraint. We acknowledge that generalization is an important consideration for practical deployment. Although it is theoretically possible for a single neural network to learn a representation that supports

variable tone counts, this capability has not yet been validated within the current study. Future work will investigate whether a unified model can reliably generalize across different sparsity levels without retraining.

More representative NYFR datasets can be generated by incorporating environmental effects such as additive white Gaussian noise (AWGN), phase noise, and signal tones in realistic ranges (e.g., 2–3 GHz). Non-integer input frequencies should also be included, with input magnitudes and recovered outputs expressed in dB to better capture signal-to-noise relationships. An enhanced NYFR digital twin can further model quantization effects in the ADC, allowing metrics such as SINAD and ENOB to be computed. Testing the MLP on signals beyond five tones will demonstrate its generalization capability. The flexibility of SpectraMelt also supports compound A2I architectures, such as the Nyquist Folding Wavelet Bandpass Sampler (NFWBS), which combines the NYFR with the Non-Uniform Wavelet Bandpass Sampler (NUWBS) [19]. This architecture can be validated within SpectraMelt by implementing its CS measurement model and verifying signal reconstruction.

Author Contributions: P.S.: Conceptualization, methodology, software, formal analysis, investigation, data curation, writing—original draft preparation; S.R.: Supervision, writing—review and editing; S.S.: Writing—review and editing; J.M.: Conceptualization (Thesis foundation), consultation, limited but critical feedback on simulator implementation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The SpectraMelt code will be made publicly available in a GitHub repository upon publication and was developed using Python 3.12. It was used for both data collection and analysis in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Donoho, D. Compressed Sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [CrossRef]
2. Rani, M.; Dhok, S.B.; Deshmukh, R.B. A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications. *IEEE Access* **2018**, *6*, 4875–4894. [CrossRef]
3. Kirolos, S.; Laska, J.; Wakin, M.; Duarte, M.; Baron, D.; Ragheb, T.; Massoud, Y.; Baraniuk, R. Analog-to-Information Conversion via Random Demodulation. In Proceedings of the IEEE Dallas Circuits and Systems Workshop on Design, Applications, Integration and Software, Richardson, TX, USA, 29–30 October 2006. Available online: https://ieeexplore.ieee.org/document/4115115 (accessed on 21 January 2026).
4. Mishali, M.; Eldar, Y. From Theory to Practice: Sub-Nyquist Sampling of Sparse Wideband Analog Signals. *IEEE J. Sel. Top. Signal Process.* **2010**, *4*, 375–391. [CrossRef]
5. Fudge, G.L.; Bland, R.; Chivers, M.; Ravindran, S.; Haupt, J.; Pace, P.E. A Nyquist Folding Analog-to-Information Receiver. In Proceedings of the Asilomar Conference on Signals, Computing, and Systems, Pacific Grove, CA, USA, 26–29 October 2008. Available online: https://ieeexplore.ieee.org/document/5074464 (accessed on 21 January 2026).
6. Torlak, M.; Namgoong, W. Spectral Detection of Frequency-Sparse Signals: Compressed Sensing vs. Sweeping Spectrum. *IEEE Access* **2021**, *9*, 30060–30070. [CrossRef]
7. Swartz, P.; Ren, S.; Sun, S. Improved Sampling Model for the Nyquist Folding Receiver. In Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 15–18 July 2024; pp. 173–179. [CrossRef]
8. Guan, T.; Chang, S.; Deng, Y.; Xue, F.; Wang, C.; Jia, X. Oriented SAR Ship Detection Based on Edge Deformable Convolution and Point Set Representation. *Remote Sens.* **2025**, *17*, 1612. [CrossRef]
9. Chen, S.; Jiang, K.; Wu, S.; Zhu, J.; Tang, B. The Impact of NYFR's Parameters on Reconstruction. In Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking, Chennai, India, 23–25 March 2016; pp. 1686–1692. Available online: https://ieeexplore.ieee.org/document/7566427 (accessed on 21 January 2026).
10. Mishali, M.; Eldar, Y; Elron A. Xampling: Signal Acquisition and Processing in Union of Subspaces. *IEEE Trans. Signal Process.* **2011**, *59*, 4719–4734. [CrossRef]

11. Martin, J.C. Analysis of the Nyquist Folding Receiver. Master's Thesis, University of Oklahoma, Norman, OK, USA, 2018. Available online: https://shareok.org/server/api/core/bitstreams/99b3cb54-12e6-4efa-aba8-69d7756bde8b/content (accessed on 21 January 2026).
12. Abari, O.; Lim, F.; Chen, F.; Stojanović, V. Why Analog-to-Information Converters Suffer in High-Bandwidth Sparse Signal Applications. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2013**, *60*, 2273–2284. [CrossRef]
13. Maleh, R.; Fudge, G.; Boyle, F.; Pace P. Analog-to-Information and the Nyquist Folding Receiver. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2012**, *2*, 564–578. [CrossRef]
14. Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* **1958**, *65*, 386–408. [CrossRef] [PubMed]
15. Wan, T.; Jiang, K.-l.; Ji, H.; Tang, B. Deep Learning-Based LPI Radar Signals Analysis and Identification Using a Nyquist Folding Receiver Architecture. *Def. Technol.* **2023**, *19*, 196–209. [CrossRef]
16. Peter, R.J.; Murthy, C.R. Learned-SBL: A Deep Learning Architecture for Sparse Signal Recovery. *arXiv* **2019**. [CrossRef]
17. Blumensath, T.; Yaghoobi, M.; Davies, M. Iterative Hard Thresholding and L0 Regularisation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, HI, USA, 15–20 April 2007. Available online: http://www.mehrdadya.com/doc/BYDICASSP07.pdf (accessed on 21 January 2026).
18. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3521–3526. [CrossRef] [PubMed]
19. Pelissier, M.; Studer, C. Non-Uniform Wavelet Sampling for RF Analog-to-Information Conversion. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 471–484. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.