2024 International Workshop on the Theory of Computational Sensing and its Applications to Radar, Multimodal Sensing and Imaging (CoSeRa)

2024 International Workshop on the Theory of Computational Sensing and its Applications to Radar, Multimodal Sensing and Imaging (CoSeRa) | 979-8-3503-6550-4/24/$31.00 ©2024 Crown | DOI: 10.1109/CoSeRa60846.2024.10720377

# OMP-Net: Neural network unrolling of weighted Orthogonal Matching Pursuit

Sina Mohammad-Taheri
*Department of Mathematics and Statistics*
*Concordia University*
Montréal, Canada
sina.mohammadtaheri@concordia.ca

Matthew J. Colbrook
*DAMTP*
*University of Cambridge*
Cambridge, United Kingdom
mjc249@cam.ac.uk

Simone Brugiapaglia
*Department of Mathematics and Statistics*
*Concordia University*
Montréal, Canada
simone.brugiapaglia@concordia.ca

***Abstract*—In recent years, algorithm unrolling has emerged as a promising methodology in various signal-processing applications, intending to combine the strengths of iterative algorithms and deep learning. Despite its success, unrolling greedy sparse recovery algorithms, particularly Orthogonal Matching Pursuit (OMP), has yet to receive much attention. The primary challenge is the non-differentiable nature of the argsort operator, a key component in greedy algorithms, which hinders gradient back-propagation during training. To address this, we introduce 'OMP-Net' by utilizing softsorting to approximate the argsort operator in a differentiable manner. Our numerical and theoretical analysis shows that under certain conditions, the approximation error is minimal, and the performance of the approximated OMP, which we call 'Soft-OMP', closely matches the original. We then incorporate Soft-OMP into the feedforward neural network's layers, integrating learnable weight parameters to connect our approach to weighted sparse recovery. Our numerical results demonstrate that this network is trainable and can surpass the performance of the original OMP in certain scenarios.**

***Index Terms*—Algorithm unrolling, OMP, greedy algorithms, neural networks**

## I. Introduction

Sparse recovery has been a core paradigm in data science over the past few decades. More recently, modern applications such as medical imaging, radio-astronomy, radars, etc., have urged exploring other data-driven approaches beyond conventional sparse recovery techniques, and neural networks have usually been at the center of attention. However, lack of verifiability and interpretability have generally been the main drawbacks of neural networks [1]. Thus, many attempts have been made to revive conventional sparse recovery techniques in the new spirit of neural networks. 'Algorithm unrolling' establishes this connection and has shown the capability to bring rigorous theoretical characteristics to this field [2], [3].

Many well-known sparse recovery algorithms have been unrolled in the literature, including Iterative Shrinkage and Thresholding (ISTA), Iterative Hard Thresholding (IHT), the Alternating Direction Method of Multipliers (ADMM), and the Primal-Dual iteration [1], [2], [4]–[6]. However, little to almost no attempt was devoted to unrolling greedy sparse recovery algorithms, notably Orthogonal Matching Pursuit (OMP) [7], [8] and Compressive Sampling Matching Pursuit (CoSaMP) [9]. The main obstacle that hinders unrolling OMP and CoSaMP is the non-differentiability of these algorithms due to (arg)sorting, which does not allow gradients to flow through the algorithm's steps during the learning phase of a neural network.

In this paper, we re-interpret iterations of OMP through a projection-based lens and exploit softsorting, an approximation of (arg)sorting operator, to derive a differentiable version of OMP with controlled approximation error, that we term 'Soft-OMP.' We then unfold Soft-OMP onto a neural network with weights as semantic learnable parameters associated with the domain (prior) knowledge of the problem we learn by end-to-end training. We call this neural network 'OMP-Net' and show that these weights serve suitably for other weighted sparse recovery algorithms. Therefore, our approach can also be regarded as a meaningful data-driven technique for learning weights. Our contributions can be summarized as follows:

- We propose differentiable OMP, or Soft-OMP, that approximates OMP with minimal approximation error, which we show theoretically and experimentally.
- We unfold iterations of Soft-OMP onto a neural network and illustrate that this network is trainable.
- The parameters to be learned in the training phase of the neural network are semantically defined as weights to discover latent structures within the data.
- We show that the weights learned by OMP-Net can be adopted for other weighted sparse recovery methods.

## II. Background

We consider the linear system of equations

$$y = Ax + e, \tag{1}$$

that we aim to solve for the unknown sparse (or compressible) signal $x \in \mathbb{C}^N$, which is observed by the vector $y \in \mathbb{C}^m$ through the sensing or measurement matrix $A \in \mathbb{C}^{m\times N}$. In other contexts, $A$ might be regarded as an (overcomplete) dictionary in which we seek to find the expansion coefficients of $y$ onto its so-called atoms. The vector $e \in \mathbb{C}^m$ represents any source of error, such as noise or numerical error. In particular, we are interested in solving the system (1) in the underdetermined case, i.e., when $m \ll N$.

979-8-3503-6550-4/24/$31.00 ©2024 Crown


Authorized licensed use limited to: CAMBRIDGE UNIV. Downloaded on January 13,2025 at 17:51:09 UTC from IEEE Xplore. Restrictions apply.

The realm of compressed sensing and sparse recovery has witnessed the emergence of a wealth of methods and algorithms to tackle this problem in the past few decades. Broadly, these methods mostly fall into either the convex relaxation methods or iterative algorithms, which include greedy and thresholding algorithms. Many of these methods have also attempted to employ domain knowledge in their approaches. This is commonly achieved by considering some structure on the signal sparsity, such as sparsity in levels [10], tree sparsity [11], or joint sparsity [12]. To our particular interest, there is also *weighted sparsity*, which is mostly utilized to incorporate prior knowledge or to promote structure rather than being a structure itself. A contextualized discussion on weighted sparsity can be found, e.g., in [13]. Several other seminal works have shown that weights are useful to accelerate the convergence of sparse recovery algorithms, improve the recovery performance, and mitigate the curse of dimensionality in sample complexity [14]–[16].

### A. Weighted OMP (WOMP)

OMP is the most well-studied algorithm among *greedy* sparse recovery algorithms. The reason behind the 'greedy' nomenclature is that instead of solving for the whole solution of the sparse recovery problem all at once, these algorithms gradually construct the signal support by adding validated indices based on a *greedy selection* criterion. This update step is followed by a data-fitting stage in which the coefficients corresponding to the columns of $A$ restricted to the updated support are approximated by minimizing the residual. In OMP, the support is updated by only one index at each iteration, and the data-fitting step translates to solving the least-squares problem. Well-known advantages of OMP are its simple formulation, computational efficiency, and good performance, especially in lower sparsity regimes (when $s$ is not too large). The algorithm below shows the main steps of (W)OMP, an extension of OMP to the weighted case that previously appeared, e.g., in [17], [18], in its simplest form:

$$S^{(n+1)}=S^{(n)} \cup \underset{j\in[N]}{\operatorname{argmax}} \left| \left( WA^*(y-Ax^{(n)}) \right)_j \right|, \quad \text{(OMP.1)}$$

$$x^{(n+1)}=\underset{z\in\mathbb{C}^N}{\operatorname{argmin}} \|y-Az\|_2^2 \ \text{ s.t. } \operatorname{supp}(z)\subseteq S^{(n+1)}, \quad \text{(OMP.2)}$$

where $S^{(n+1)}$ denotes the support of the signal $x^{(n+1)}$ after $n$ iterations, with $S^{(0)} = \emptyset$ and $x^{(0)} = 0$, and the columns of $A$ are assumed to have unit $\ell^2$-norm. There is a key difference between this algorithm and the original OMP algorithm proposed in [7], [8]. Namely, its greedy selection rule, i.e., $WA^*(y - Ax^{(n)})$, includes the weight parameter $W \in \mathbb{R}^{N\times N}$. This parameter opens up the way to generalize OMP to the weighted case. A meaningful choice of $W$ is $W = \operatorname{diag}(w),\ w \in \mathbb{R}^N$, where $\operatorname{diag}(w)$ is a diagonal matrix with $w$ on its main diagonal. Adjusting the elements of $w$, one can prioritize the choice of corresponding indices to enter the support of the signal and thus incorporate prior knowledge into the recovery of $x$ via OMP. More intuitively, at each iteration of OMP, $w$ points towards the support's indices based on prior knowledge. This is not the only method by which one can encourage weights in OMP; see also [19].

### B. Algorithm unrolling

Algorithm unrolling designs deep neural network architectures through the iterations of an iterative algorithm as layers of a neural network [2], [3]. Since the network is constituted of a model-based algorithm, it is highly interpretable. Often, this approach inherits the favorable characteristics of the associated iterative algorithm, such as stability and generalization bounds [1]. By introducing end-to-end training of learnable parameters during the training phase, one can gain superior performance with respect to the original iterative algorithm and possibly with fewer iterations (layers). In recent years, many works have been devoted to unrolling sparse recovery algorithms onto neural networks [2], [4], [5], but to the best of our knowledge, unrolling OMP still needs to be explored. In the following section, we will explain the main challenge of doing so.

### C. Sorting: the main challenge in unrolling OMP

Despite the appeal of algorithm unrolling, many iterative algorithms are not readily 'unrollable,' and further steps should be taken to be able to unroll them onto neural networks appropriately. The main obstacle in this direction is the existence of non-differentiable operators that hinder training unrolled networks via gradient-based optimization schemes. OMP exemplifies such an algorithm due to the existence of the sorting operator within its steps. More specifically, implicit to the greedy selection rule of OMP is the application of functions $\mathsf{sort} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathsf{argsort} : \mathbb{R}^n \to \mathcal{S}_n \subset [n]^n$ defined for any $v \in \mathbb{R}^n$ as

$$\begin{aligned} &\mathsf{sort}(v) = (v_{i_1}, v_{i_2}, \dots, v_{i_n}), \quad v_{i_1} \geq v_{i_2} \geq \cdots \geq v_{i_n}, \\ &\mathsf{argsort}(v) = (i_1, i_2, \dots, i_n). \end{aligned} \quad (2)$$

Here $\mathcal{S}_n$ is the set of permutations on $[n]$. Both of these functions are non-differentiable (in fact, discontinuous). Proper handling of these operators is the main challenge towards directly unrolling OMP onto neural networks. In step (OMP.1), the algorithm needs to select the index corresponding to the largest absolute entry of $WA^*(y - Ax^{(n)})$, which intrinsically involves sorting elements of this vector. This index is later used implicitly in (OMP.2) to restrict the columns of $A$ to $S^{(n+1)}$ and approximately solve the equation $y = A_{S^{(n+1)}}z$ for $z \in \mathbb{C}^{(n+1)}$. Hence, the gradient cannot be computed with respect to any of the parameters of $WA^*(y - Ax^{(n)})$, e.g., $W$ (i.e., $w$), which is essential in the backpropagation procedure of a neural network-based OMP. In the sequel, we leverage the softsort operator to tackle this issue.

## III. Main results

We now introduce Soft-OMP, a differentiable substitute for OMP that allows backpropagation in a gradient-based neural network optimization. We then show that Soft-OMP well approximates OMP under suitable parameter regimes. Lastly,



Authorized licensed use limited to: CAMBRIDGE UNIV. Downloaded on January 13,2025 at 17:51:09 UTC from IEEE Xplore. Restrictions apply.

we unroll iterations of Soft-OMP onto layers of a neural network that we will term 'OMP-Net.'

### A. Soft-OMP

The operators sort and argsort, defined in (2), can be characterized by $n \times n$ permutation matrices defined as

$$P_{\mathsf{sort}}(i,j) = P_{\mathsf{argsort}}(i,j) = \begin{cases} 1 & j = \mathsf{argsort}(v)_i \\ 0 & \text{otherwise} \end{cases},$$

so that $\mathsf{sort}(v) = P_{\mathsf{sort}} v$ and $\mathsf{argsort}(v) = P_{\mathsf{argsort}} \overline{\mathbb{1}}_n$, where $\overline{\mathbb{1}}_n := (1, 2, \ldots, n)^\top$. The key ingredient of Soft-OMP is the softsorting operator, proposed in [20] and defined below, that approximates $P_{\mathsf{argsort}}$ (or equivalently $P_{\mathsf{sort}}$) in a differentiable manner by the approximate permutation matrix $\tilde{P}_{\mathsf{argsort}}$:

$$\tilde{P}_{\mathsf{argsort}} = \mathsf{softsort}(v) = \mathsf{softmax}\left(\frac{-|\mathsf{sort}(v)\mathbb{1}^T - \mathbb{1}v^T|}{\tau}\right), \tag{3}$$

for any $v \in \mathbb{R}^n$, where $\mathbb{1} = (1, 1, \ldots, 1)^\top$ and the *temperature parameter* $\tau$ controls the degree of approximation. In equation (3), the absolute value $|\cdot|$ is applied element-wise, and $\mathsf{softmax}(y)_j = e^{y_j} / \sum_{i=1}^n e^{y_i}$, $y \in \mathbb{R}^n$, is applied row-wise. $\tilde{P}_{\mathsf{argsort}}$ is referred to as 'unimodal row stochastic' matrix, and admits *non-negativity*, *row affinity* and *argmax permutation* properties [20], [21].

We now reinterpret iterations of OMP through a projection-based lens, which paves the way for softsorting to be incorporated. We first note that the link between (OMP.1) and (OMP.2) is established through $S^{(n+1)}$ and the variable in (OMP.2) that holds the dependence with respect to (OMP.1) is $A_{S^{(n+1)}}$, the restriction of columns of $A$ to $S^{(n+1)}$. Fundamental to our derivation is applying a permutation matrix on the space of columns of A, i.e., $\mathrm{span}\{a_j\}_{j=1}^N$, that projects them onto the restricted space $\mathrm{span}\{a_j\}_{j \in S^{(n+1)}}$. In matrix language, this can be represented as $A_{S^{(n+1)}} = A(\Pi^{(n+1)})^T$, where $\Pi^{(n+1)}$ is the aforementioned projection matrix that we approximate with the softsort operator. This leads to the following differentiable OMP algorithm that we term 'Soft-OMP':

$$\begin{cases} \tilde{P}^{(n+1)} = \mathsf{softsort}\left(\left|\left(WA^*(y - A\tilde{x}^{(n)})\right)\right|\right) \\ \tilde{\Pi}^{(n+1)} = \left[\tilde{\Pi}^{(n)}; \tilde{P}^{(n+1)}[1,:]\right] \end{cases}, \quad \text{(Soft-OMP.1)}$$

$$\begin{cases} \tilde{B}^{(n+1)} = A(\tilde{\Pi}^{(n+1)})^T \\ \tilde{w}^{(n+1)} = \underset{z \in \mathbb{C}^{n+1}}{\mathrm{argmin}} \|y - \tilde{B}z\|_2^2 \\ \tilde{x}^{(n+1)} = (\tilde{\Pi}^{(n+1)})^T \tilde{w}^{(n+1)} \end{cases}, \quad \text{(Soft-OMP.2)}$$

where $M[j,:]$ denotes the $j$th row of a matrix $M$. The distinguishing characteristic of OMP with respect to many other algorithms is that the support is updated by only one index per iteration. Hence, in the algorithm above, we only require to add the first row of the approximate projection matrix $\tilde{P}^{(n+1)}$, computed by softsorting and which corresponds to the largest element of $\tilde{v}^{(n+1)} = \left|WA^*(y - A\tilde{x}^{(n)})\right|$, to $\tilde{\Pi}^{(n)}$, whose rows consist of all permutations corresponding to maximal elements of $\tilde{v}^{(k)}$, $k = 1, \ldots, n$.

We next show through the following theorem that iterations of Soft-OMP asymptotically match the ones of OMP, and admit minimal error rates controlled under appropriate values of $\tau$, the temperature parameter of softsorting. The proof of this theorem will be given in [22].

*Theorem 1 (Soft-OMP is a good approximation to OMP):* Let $x^{(n+1)} \in \mathbb{C}^N$ and $\tilde{x}^{(n+1)}(\tau) \in \mathbb{C}^N$ be the signal acquired after $n$ iterations of OMP and Soft-OMP, respectively. Assume that $\tilde{\Pi}^{(n+1)} \in \mathbb{R}^{(n+1)\times N}$ and $\Pi^{(n+1)} \in \mathbb{R}^{(n+1)\times N}$ are projection matrices whose rows consist of all permutations corresponding to maximal elements of $v^{(n+1)} = |A^*(y - Ax^{(n)})| \in \mathbb{R}^N$ and $\tilde{v}^{(n+1)} = |A^*(y - A\tilde{x}^{(n)})| \in \mathbb{R}^N$, associated with OMP and Soft-OMP respectively. Also for all $i \in [n]$, assume that no ties exist in $\tilde{v}^{(i)}$, i.e., $\tilde{v}^{(i)}_{\hat{j}^{(i)}} \neq \tilde{v}^{(i)}_j$, $\forall j \in [N] \backslash \{\hat{j}^{(i)}\}$, where $\hat{j}^{(i)} := \arg\max_{j \in [N]} \tilde{v}^{(n)}$. Then, the following hold:

(i) As $\tau \to 0$, we have $\tilde{\Pi}^{(n)} \to \Pi^{(n)}$ and $\tilde{x}^{(n)} \to x^{(n)}$.

(ii) If $\tau$ satisfies the condition $\tau \leq \tilde{g} / \log(C/\epsilon)$, with

$$\tilde{g} = \min_{i \in [n]} \min_{j \in [N], j \neq \hat{j}^{(i)}} |\tilde{v}^{(i)}_{\hat{j}^{(i)}} - \tilde{v}^{(i)}_j|,$$

$$C = \sqrt{2n}(N-1)\left(\frac{\sqrt{1-\delta_s} + (\sqrt{n}+1)\|A\|_2}{1-\delta_s}\right)\|y\|_2,$$

and where $\delta_s$ and $\|A\|_2$ are, respectively, the $s$th restricted isometry constant (RIC) and the spectral norm of $A$, then $\|x^{(n)} - \tilde{x}^{(n)}\|_2 \leq \epsilon$.

*Remark 1:* For matrices whose spectral norm can be controlled, e.g., subgaussian matrices, the constant $C$ can be explicitly bounded. This leads to a practical range for $\tau$ sufficient to achieve a desirable accuracy [22].

*Remark 2:* Theorem 1 determines the gap between iterations of OMP and Soft-OMP. Additionally, in the spirit of the recovery guarantee of OMP proposed in [23], [24] and the inequality $\|x - \tilde{x}^{(n)}\| \leq \|x - x^{(n)}\| + \|x^{(n)} - \tilde{x}^{(n)}\|$, Theorem 1 establishes a recovery guarantee for Soft-OMP.

### B. OMP-Net

A natural consequence of Theorem 1 is that approximating OMP through the application of softsorting comes at the cost of losing some performance, which is controlled by $\tau$. We can compensate for this performance loss (and gain more) by integrating Soft-OMP into a neural network-based optimization problem and allowing an appropriate parameter to be trained in the learning process. We choose $w$ as the trainable parameter in this work over other potential candidates. As $w$ stands for weights, this can be regarded as a quest for finding a suitable structure in the data and, in this sense, connects our approach to weighted sparse recovery and learning weights through *bilevel optimization* [25]. This places OMP-Net in two prominent positions: (1) a neural network architecture based on OMP designed to adapt to the inner structure of the data and (2) a strategy aimed at learning weights for other weighted sparse recovery methods.

OMP-Net is trained from pairs of data $\{(y^{(i)}, x^{(i)})\}_{i=1}^{N_{\text{train}}}$ that are instances of observation and signal vectors and serve as



Authorized licensed use limited to: CAMBRIDGE UNIV. Downloaded on January 13,2025 at 17:51:09 UTC from IEEE Xplore. Restrictions apply.

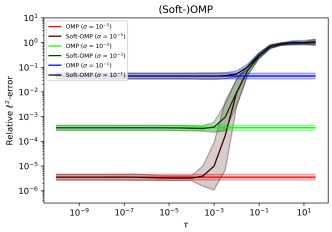


Fig. 1: Relative $\ell^2$-error of Soft-OMP and OMP as a function of $\tau$ and with $\sigma \in 10^{\{-5,-3,-1\}}$.

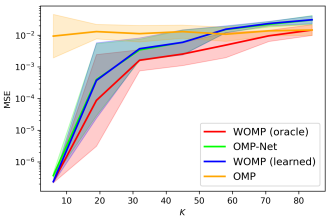


Fig. 3: MSE of OMP-Net, OMP, WOMP with oracle and learned weights, with respect to $K$.

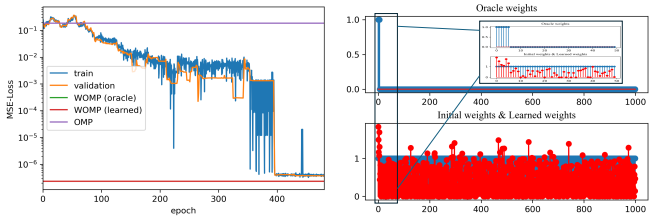


Fig. 2: Left: Training and validation loss of OMP-Net, OMP, WOMP (oracle), and WOMP with learned weights over training epochs. Right: Oracle weights (top), initialization, and learned weights of OMP-Net (bottom).

inputs and outputs of the network, respectively. Then, the Mean Squared Error (MSE) loss

$$\mathcal{L}(\Theta) = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \|x^{(i)} - f_\Theta(y^{(i)})\|_2^2,$$

is minimized using batches of size $N_{\text{batch}}$ by *Stochastic Gradient Descent* (SGD) or any other gradient-based optimization method. Here $f_\Theta$ represents OMP-Net, $\Theta = (W^{(l)})_{l=1}^L$ (possibly shared among the layers, i.e., $W^{(1)} = W^{(2)} = \cdots = W^{(L)}$) the trainable parameters, and $L$ denotes the number of layers, generally corresponding to the number of iterations of OMP.

## IV. Numerical Experiments

### A. Soft-OMP vs. OMP

In the first experiment of this section, we put Theorem 1 to the test. We compare the performance of OMP with its differentiable counterpart, simultaneously investigating the effect of $\tau$. Our goal is to show that in an appropriate parameter regime, Soft-OMP coincides with OMP, and the approximation error is controlled by $\tau$, the temperature parameter of Soft-OMP. The experimental setup is as follows: We generate a measurement matrix $A \in \mathbb{R}^{m\times N}$ with $\ell^2$-normalized columns obtained by rescaling the columns of a matrix $\widetilde{A}$ whose independent entries are drawn from a standard normal distribution with zero mean and unit standard deviation, i.e., $\widetilde{A}_{i,j} \sim \mathcal{N}(0,1),\ i \in [m], j \in [N]$. Non-zero entries of the $s$-sparse signal $x \in \mathbb{R}^N$ also belong to a standard normal distribution, while their positions are chosen randomly and uniformly on a subset $S \subseteq [N]$ of size $s$. Therefore, the observation vector $y \in \mathbb{R}^m$ is obtained as $y = Ax + e$, where $e \in \mathbb{R}^m$ is the measurement error with independently and identically distributed entries from a normal distribution with zero mean and standard deviation $\sigma$. We recover $x$ using $N_{\text{iter}}$ iterations of OMP (weights set to $W = \mathbb{1}$) and Soft-OMP algorithms for different values of $\tau$. We redo this experiment for $N_{\text{trial}} = 50$ repetitions and various noise levels with the following experimental settings: $N = 160,\ m = 100,\ s = 10,\ \sigma \in 10^{\{-5,-3,-1\}},\ N_{\text{iter}} = 10$. Figure 1 shows the results as shaded plots. Solid curves represent mean relative $\ell^2$-errors as a function of $\tau$. These are surrounded from above and below by the boundaries $(\tau, 10^{\mu_\sigma^\tau + \rho_\sigma^\tau})$ and $(\tau, 10^{\mu_\sigma^\tau - \rho_\sigma^\tau})$ respectively, with $\mu_\sigma^\tau = \frac{\sum_{i=1}^{N_{\text{trial}}} (\log(E_\sigma^\tau))_i}{N_{\text{trial}}}$, $\rho_\sigma^\tau = \frac{\sum_{i=1}^{N_{\text{trial}}} (\log(E_\sigma^\tau)_i - \mu_\sigma^\tau)^2}{N_{\text{trial}}-1}$. Here $E_\sigma^\tau$ denotes the relative $\ell^2$-error for the noise level $\sigma$ and at $\tau$, with $E$ defined as $E = \|x - \hat{x}\|_2 / \|x\|_2$, where $\hat{x}$ is the signal recovered by OMP or Soft-OMP. We observe, as predicted by Theorem 1, that if $\tau$ is sufficiently small, Soft-OMP approximates OMP with minimal error.

### B. Learning OMP-Net

In this section, we demonstrate the trainability of OMP-Net and that the weights learned by the network can be utilized for other weighted sparse recovery algorithms, particularly WOMP. We run our experiments in settings where OMP fails to reconstruct the signal (in terms of MSE) appropriately and where prior knowledge about the signal structure promoted through weights is proven to be more beneficial [15], [19]. We always fix the number of layers equal to $s$, let $\tau = 10^{-3}$ and train the network with RMSprop optimizer with learning rate $10^{-2}$ and shared weights among the layers. We deem it necessary to mention that although lower values of $\tau$ improve Soft-OMP's approximation of OMP, as indicated by Theorem 1, they also lead to vanishing gradients in the network. Thus, large values of $\tau$ are generally preferable for gradient stability [20], [21]. However, very large values of $\tau$ also degrade the performance of Soft-OMP with respect to OMP to such an extent that it would be very difficult (if not impossible) for the network to compensate for this. As a result, a trade-off exists on the choice of $\tau$ that should be handled properly. We empirically noticed that such trade-off is balanced for values of $\tau$ in the phase transition region of the curves in Figure 1.

*a) Proof of concept:* We first present a proof of concept for the trainability of OMP-Net. The compressed sensing setup that we run for this experiment is as follows: $N = 1000, m = 32, s = 6, \sigma = 10^{-3}$. The dataset to train the network consists of sparse signals with nonzero entries from only the support's first $s$ elements. However, we do not assume any oracle knowledge for the network, and it is initialized with $w = \mathbb{1}$. The left image in Figure 2 exhibits the training and validation loss over 350 epochs. For comparison, we also plotted OMP and WOMP provided with oracle weights (top right in Figure 2). Furthermore, we feed WOMP with weights learned by OMP-Net (WOMP with learned and oracle weights overlap in the image).



Authorized licensed use limited to: CAMBRIDGE UNIV. Downloaded on January 13,2025 at 17:51:09 UTC from IEEE Xplore. Restrictions apply.

*b) Effect of domain knowledge:* We investigate the effectiveness of OMP-Net in extracting the correct structure within the data while the structure level is subject to change. To that purpose, let $[K] \subseteq [N]$, be the proportion of the signal ambient dimension that consists of the signal's support. For each $K$ in the discrete range $[s, N]$, we generate a new dataset and repeat this procedure $N_{\text{trial}}$ times. We set $N = 84$, but keep $m$, $s$ and $\sigma$ the same as in the previous experiment. OMP-Net is always initialized with $w = \mathbb{1}$ and the weights for WOMP are set to $w_j = 1$ if $j \leq K$ and 0 otherwise. i.e., WOMP is provided with the oracle knowledge, but OMP-Net has to learn the structure through the learning process. We note that at $K = N$, weights would be $W = \mathbb{1}$; thus, WOMP corresponds to OMP. This experiment's result is summarized in shaded plots with respect to MSE (rather than relative error) in Figure 1, where OMP and WOMP with learned weights are also added. We devise checkpoints every ten epochs during training OMP-Net and report the best performance on the test data in Figure 3. It is observable that OMP-Net can learn the structure close to the oracle level represented by WOMP. As $K$ approaches $N$, OMP-Net approaches the performance of OMP, which fails to reconstruct the signal well due to the severely undersampled regime.

## V. Conclusion and extensions

We proposed unrolled OMP, a greedy sparse recovery algorithm, that we termed 'OMP-Net.' We utilized soft-sorting to overcome the intrinsic non-differentiability issue within the iterations of OMP. Reinterpreting OMP through a projection-based lens, we approximated permutation matrices from argsort with stochastic matrices derived from soft sorting. Our numerical and theoretical analysis demonstrated that under certain conditions, the approximation error is minimal, and the performance of the approximated greedy algorithm closely matches the original. The framework proposed in this paper can be exploited for other generalizations of OMP (see, e.g., [19]). Another natural extension currently under development would be CoSaMP, which includes an additional sorting step because of hard thresholding. Unlike OMP, CoSaMP benefits from the advantage that the required number of iterations to converge to an admissible solution is disentangled from $s$, which translates to fewer layers in the unrolled neural networks and thus makes it more suitable for high-dimensional applications such as imaging.

## Acknowledgment

SMT acknowledges the financial support of MITACS. SB acknowledges NSERC through grant RGPIN-2020-06766, and FRQNT through grant 313276. The authors would like to thank Ben Adcock for preliminary feedback on this work.

## References

[1] M. J. Colbrook, V. Antun, and A. C. Hansen, "The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and smale's 18th problem," *Proceedings of the National Academy of Sciences*, vol. 119, no. 12, p. e2107151119, 2022.

[2] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 399–406.

[3] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[4] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A deep learning approach for image compressive sensing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 521–538, 2018.

[5] B. Xin, Y. Wang, W. Gao, D. Wipf, and B. Wang, "Maximal sparsity with deep networks?" *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[6] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.

[7] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*. IEEE, 1993, pp. 40–44.

[8] G. M. Davis, S. G. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *Optical Engineering*, vol. 33, no. 7, pp. 2183–2191, 1994.

[9] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[10] B. Adcock, A. C. Hansen, C. Poon, and B. Roman, "Breaking the coherence barrier: A new theory for compressed sensing," in *Forum of Mathematics, Sigma*, vol. 5. Cambridge University Press, 2017.

[11] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[12] B. Adcock, A. Gelb, G. Song, and Y. Sui, "Joint sparse recovery based on variances," *SIAM Journal on Scientific Computing*, vol. 41, no. 1, pp. A246–A268, 2019.

[13] H. Rauhut and R. Ward, "Interpolation via weighted $\ell_1$ minimization," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 321–351, 2016.

[14] M. P. Friedlander, H. Mansour, R. Saab, and Ö. Yilmaz, "Recovering compressively sampled signals using partial support information," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1122–1134, 2011.

[15] B. Bah and R. Ward, "The sample complexity of weighted sparse approximation," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3145–3155, 2016.

[16] B. Adcock, S. Brugiapaglia, and C. G. Webster, *Sparse Polynomial Approximation of High-Dimensional Functions*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2022. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611976885

[17] J.-L. Bouchot, H. Rauhut, and C. Schwab, "Multi-level compressed sensing Petrov-Galerkin discretization of high-dimensional parametric PDEs," *arXiv preprint arXiv:1701.01671*, 2017.

[18] G. Z. Li, D. Q. Wang, Z. K. Zhang, and Z. Y. Li, "A weighted OMP algorithm for compressive UWB channel estimation," in *Applied Mechanics and Materials*, vol. 392, 2013, pp. 852–856.

[19] S. Mohammad-Taheri and S. Brugiapaglia, "The greedy side of the LASSO: New algorithms for weighted sparse recovery via loss function-based orthogonal matching pursuit," *arXiv preprint arXiv:2303.00844*, 2023.

[20] S. Prillo and J. Eisenschlos, "Softsort: A continuous relaxation for the argsort operator," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7793–7802.

[21] A. Grover, E. Wang, A. Zweig, and S. Ermon, "Stochastic optimization of sorting networks via continuous relaxations," *arXiv preprint arXiv:1903.08850*, 2019.

[22] S. Mohammad-Taheri, M. J. Colbrook, and S. Brugiapaglia, "Deep greedy unfolding," *In preparation*, 2024.

[23] T. Zhang, "Sparse recovery with orthogonal matching pursuit under RIP," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6215–6221, 2011.

[24] A. Cohen, W. Dahmen, and R. DeVore, "Orthogonal matching pursuit under the restricted isometry property," *Constructive Approximation*, vol. 45, no. 1, pp. 113–127, 2017.

[25] S. Dempe, *Foundations of Bilevel Programming*. Springer Science & Business Media, 2002.



Authorized licensed use limited to: CAMBRIDGE UNIV. Downloaded on January 13,2025 at 17:51:09 UTC from IEEE Xplore. Restrictions apply.