Learning Structured Sparse Matrices for Signal Recovery via Unrolled Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Countless signal processing applications include the reconstruction of an unknown signal from very few indirect linear measurements. Because the measurement operator is commonly constrained by the hardware or the physics of the observation process, finding measurement matrices that enable accurate signal recovery poses a challenging discrete optimization task. Meanwhile, recent advances in the field of machine learning have highlighted the effectiveness of gradient-based optimization methods applied to large computational graphs such as those arising naturally when unrolling iterative algorithms for signal recovery. However, it has remained unclear how to leverage this technique when the set of admissible measurement matrices is both discrete and sparse. In this paper, we tackle this problem and propose an efficient and flexible method for learning structured sparse measurement matrices. Our approach uses unrolled optimization in conjunction with Gumbel reparametrizations. We empirically demonstrate the effectiveness of our method in two prototypical compressed sensing situations.

1 Introduction

Linear measurement operators following a structure that is constrained by the physics of the measurement process or the hardware of the measuring device are ubiquitous in signal processing. In particular, structured sparse matrices appear in many signal processing applications [20, 31, 9, 25]. The optimal design of such measurement operators—which reside in a discrete subset—to improve the performance of downstream tasks poses great computational challenges. Classical approaches commonly employ discrete optimization, as no gradients can be directly computed.

On the other hand, in the field of machine learning, gradient-based optimization via backpropagation through massive nonlinear computational graphs has shown impressive performance in many applications. A promising concept to enable gradient-based learning in non-differentiable settings is given by Gumbel reparametrizations [16, 21], which allow estimating the gradients of categorical random variables. In this work, we employ this technique to fuse gradient-based learning with the design of sparse measurement operators that are constrained to a discrete set.

Signal recovery problems are often solved by convex optimization methods. It is well established that the computational graph of many iterative convex optimization schemes can be unrolled to obtain a neural network that can be readily backpropagated through [11], enabling the computation of gradients with respect to the measurement operator and other involved parameters. However, it has remained unclear how this technique can be leveraged when the set of admissible measurement operators is sparse and discrete. We present a novel approach to tackle this problem. Our main contributions can be summarized as follows:

- We propose an efficient and easy-to-implement method for learning structured sparse measurement matrices for signal recovery by using unrolled optimization and Gumbel reparametrizations. Due to its compatibility with automatic differentiation, our method provides a novel framework for the data-driven design of practicable measurement operators.
- We successfully apply our algorithmic approach in two prototypical situations, namely compressed sensing with left-d-regular graphs and single-pixel imaging, outperforming conventional setups based on randomization.

Submitted to the NeurIPS 2021 Workshop on Deep Learning and Inverse Problems. Do not distribute.

2 Background & Related Work

2.1 Linear Inverse Problems & Compressed Sensing

In linear inverse problems, the basic task is to recover an unknown target signal $x \in \mathbb{R}^n$ from indirect observations of the form $y = \Phi x + e$, where $\Phi \in \mathbb{R}^{m \times n}$ is a known measurement matrix and $e \in \mathbb{R}^m$ noise. The number of measurements m is usually much smaller than the signal dimension n, making the inverse problem ill-posed and only solvable under prior knowledge about the underlying signals. A prominent example is given by compressed sensing, in which the signal is assumed to be sparse [6, 8]. In particular, when x is s-sparse, it is possible to robustly reconstruct x from $m \in O(s \log(n/s))$ measurements via convex optimization, given that Φ fulfills the null space property [10]. The convex program that is to be solved corresponds to a LASSO problem [27] with hyperparameter λ:

$$\min_{\hat{x}} \|\Phi \hat{x} - y\|_2^2 + \lambda \|\hat{x}\|_1. \tag{1}$$

Many randomized measurement operators fulfill the null space property with high probability [10], allowing theoretical guarantees on the reconstruction of generic sparse signals. However, for real-world applications, randomization is not a necessary prerequisite.

A widely-used class of algorithms for solving LASSO-type problems in compressed sensing are gradient-based methods. For example, an iterative proximal scheme, known as the Iterative Soft Thresholding Algorithm (ISTA) [7], can be used to solve (1):

$$\hat{x}^{(t+1)} = \text{prox}_{\lambda \|\cdot\|_1}\Big(\hat{x}^{(t)} + \gamma \nabla_{\hat{x}^{(t)}}\big(\|y - \Phi \hat{x}^{(t)}\|_2^2\big)\Big), \tag{2}$$

where $\text{prox}_{\lambda \|\cdot\|_1}(v) = \text{sign}(v) \max(|v| - \lambda, 0)$. Another popular method is based on Iterative Hard Thresholding (IHT) [4], in which the proximal operator in is replaced by a projection onto the set of s-sparse vectors by clipping the entries of all but the largest s absolute values to zero.

2.2 Unrolled Optimization in Linear Inverse Problems

Many iterative optimization schemes can be viewed as a neural network via unrolling the computational graph of a finite number of iterations T. Once unrolled, the parameters θ of a given recovery algorithm $f_\theta : \mathbb{R}^m \to \mathbb{R}^n$ as well as the measurement operator Φ can be fit to a (training) dataset by minimizing a loss function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ via (stochastic) gradient-based optimization:

$$\min_{\theta, \Phi} \mathbb{E}_x \Big[\mathcal{L}(f_\theta(\Phi x), x)\Big]. \tag{3}$$

In the context of linear inverse problems, a prominent example of unrolled optimization is Learned ISTA (LISTA) [11], where ISTA is unrolled and the involved matrices, step-sizes, and thresholds are learned in an end-to-end fashion. Such a data-driven approach can reduce the required number of iterations to solve recovery problems by orders of magnitude. Further improvements to LISTA are made by Analytic LISTA [19], and Neurally Augemented LISTA (NA-ALISTA) [2]. Unrolled optimization provides a natural approach to learning data-driven Φ by including it in an end-to-end training procedure. For unconstrained, dense Φ this has been explored in [30, 1]. However, in real-world applications, the measurement matrix must often follow constraints imposed by hardware or the underlying physics, limiting the practicability of such approaches.

2.3 Gumbel Reparametrizations

Gradient-based learning has been shown to be effective and scalable [26, 5]. However, when discrete nodes are included in the computational graphs, gradients cannot be computed directly and have to be estimated. Formally, we consider a computational graph including a discrete random variable v taking one of the values $1, \ldots, a$ (in one-hot encoding), where its unnormalized log-probabilities are denoted by $\varphi = [\varphi_1, \ldots, \varphi_a]^T \in \mathbb{R}^a$. The value of v is then passed through a deterministic, differentiable function f. As v is discrete, it is not possible to directly backpropagate through v. Therefore, an estimate of the gradient $\nabla_\varphi \mathbb{E}_v[f(v)]$ must be computed. The Gumbel-softmax trick [16, 21] allows computing a differentiable relaxation of v by adding component-wise i.i.d. Gumbel noise to φ before applying a softmax. This modification of the Gumbel-max trick [12, 22] enables the backpropagation through discrete random variables outperforming previous approaches to estimating the gradient of discrete nodes [3, 29, 23]. In cases where true discreteness is needed, it is possible to use the argmax operator in the forward pass and the softmax in the backward pass of backpropagation. This is known as the straight-through Gumbel softmax estimator. It can be directly extended to taking multiple samples without replacement from a categorical distribution, namely by selecting the top-K values instead of only the largest one [28].

3 Method

We propose a method that enables gradient-based learning of structured sparse matrices for signal recovery by using unrolled optimization in conjunction with Gumbel reparametrizations. To this end, we model the measurement matrix $\Phi \in \mathbb{R}^{m \times n}$ as being governed by a learnable parameter $\varphi \in \mathbb{R}^{m \times n}$, employing Gumbel top-$k$ operators on a partition of the indices of φ. Formally, consider the index set $\mathcal{I} := \{1 \dots m\} \times \{1 \dots n\}$. Let $\mathcal{P}(\mathcal{I})$ denote a partition of $\mathcal{I}$, i.e., a set of non-empty pairwise disjoint subsets $I_1, \dots, I_l \subset \mathcal{I}$ such that $\bigcup_{i \in \{1 \dots l\}} I_i = \mathcal{I}$. We introduce the notation $\varphi[I_i]$ for indexing the elements of φ in the index sets I_i. Then, for each $i \in \{1 \dots l\}$, we use a Gumbel reparametrization to select d_i elements from the i-th set in $\mathcal{P}(\mathcal{I})$. This is done by adding element-wise i.i.d. Gumbel noise to each element of $\varphi[I_i]$, and then taking the top-d_i values in the forward pass. In the backward-pass of the backpropagation, the gradient of the softmax is used instead of the hard top-d_i. A pseudo-code implementation of this procedure, assuming a software framework capable of automatic differentiation, is provided in Algorithm 1.

The binary matrix resulting from Algorithm 1 can be used like any other matrix in a framework with automatic differentiation, as the gradient with respect to φ is well-defined. This means that Φ can be constructed from this binary matrix by combination with other constants or learned parameters. In that way, our method can be used in an out-of-the-box fashion to learn a masking pattern for fixed matrices such as Fourier matrices. A simpler special case is obtained by choosing the trivial partition, where all entries of Φ are governed by a single Gumbel reparametrization; in other words, a single top-K is used to select elements of the entire matrix. This can be readily applied whenever learning a single sampling mask is required, for example, in the context of compressive magnetic resonance imaging [15, 14], where a fixed number of rows from a Fourier matrix is selected. However, the flexibility of our method comes from the fact that the partition can be freely chosen to take any fine-grained structure of the matrices into account.

In many applications, such as single-pixel imaging or compressed sensing with left-d-regular graphs, the partition is naturally given by the rows or columns of φ, while the number of elements d to be selected from each partition component is equal. In this case, vectorized, and therefore computationally efficient softmax top-K operations over an axis can be applied.

In principle, our approach enables learning any type of measurement operator that can be constructed by differentiable transforms of a sparse binary matrix (which is structured in the sense that entries are subselected from a specific partition). With this in mind, Algorithm 1 should be seen as an instance of a more general framework for the data-driven design of measurement operators. A full evaluation of this methodology and potential generalizations of Algorithm 1 are beyond the scope of this paper.

Algorithm 1 Learning a binary matrix with d_i ones per set I_i of the partition.

Input: signal x (training data), top-K-keeps $d_1, \dots, d_l \in \mathbb{N}$, differentiable reconstruction algorithm $f : \mathbb{R}^m \to \mathbb{R}^n$, index partition $\mathcal{P}(\mathcal{I}) = \{I_1, \dots, I_l\}$

Learnable Parameters: Parameters of measurement matrix $\varphi \in \mathbb{R}^{m \times n}$, parameters of reconstruction algorithm θ

```
G ~_{i.i.d.} Gumbel(0,1)^{m×n}
for i ∈ {1 ... l} do
    logits := (φ[I_i] + G[I_i])
    probs := softmax(logits)
    hard := topk(probs, d_i)
    Φ[I_i] := hard.detach() + probs - probs.detach()
end for
y := Φx
loss := L(f_θ(y), x)
loss.backward()
```

4 Application: Single Pixel Imaging

A remarkable application of compressed sensing is single pixel imaging [9], which makes imaging possible using compressive measurements acquired by a spatial light modulator or a digital micromirror device that collects light onto a single pixel using a series of m masks. Mathematically, this process can be modeled as a compressed sensing problem using a binary measurement matrix Φ. While theoretical results highlight that random masks have favorable reconstruction properties for

generic sparse signals, this may not be the case when the physics of the measurement process or additional image structures are taken into account.

Our approach is well suited to incorporate such aspects. For our case study, we partition the indices of φ into the set of row vectors and use Algorithm 1 to learn adaptive masks for the MNIST dataset ($n = 784$) [18]. We unroll IHT, which has no additional learnable parameters beyond the measurement operator, and NA-ALISTA, which adaptively predicts step-sizes and thresholds by an LSTM-network [13]. The results in Figure 1 show that our method reduces the number of measurements required for good reconstruction and greatly speeds up convergence; see Appendix for implementation details. Figure 2 visualizes learned and random masks as well as some reconstructed MNIST digits: the learned masks have clear structure and lead to better reconstruction. Additional samples and masks can be found in the Appendix.

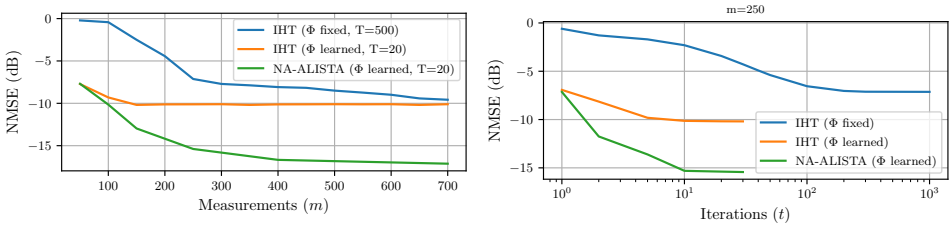
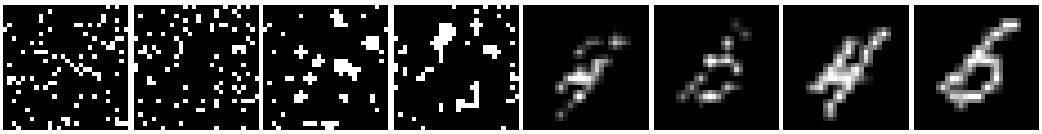


Figure 1: MNIST reconstruction in single pixel imaging setup with $d = 36$ ones per row and varying number of measurements (left), as well as varying number of iterations with fixed $m = 250$ (right).



(a) Random Masks (b) Learned Masks (c) Images (random Φ) (d) Images (learned Φ)

Figure 2: Random and learned masks as well as reconstructed samples for $m = 250$ and $d = 90$ ones per row. The reconstructions are obtained by $T = 20$ IHT-iterations.

5 Application: Compressed Sensing with Left-d-Regular Graphs

Another class of structured random matrices suitable for compressed sensing is given by adjacency matrices left-d-regular graphs. In particular, for large n, adjacency matrices of random left-d-regular graphs with $d \in O(n/s)$ and $m \in O(s \log(n/s))$ allow stable signal recovery [10]. However, for smaller values of m and n, such random graphs are unlikely to enjoy the required properties and therefore may not be usable for compressed sensing. This regime is again well suited for our approach. Invoking Algorithm 1, we partition the entries of the measurement matrix into its row vectors, and select $d = 7$ ones per row. As unrolled method, we consider Iterative Hard Thresholding for expanders (E-IHT):

$$\hat{x}^{(t+1)} = \mathcal{H}_s\big(\hat{x}^{(t)} + \mathcal{M}(y - \Phi\hat{x}^{(t)})\big). \tag{4}$$

Here, $\mathcal{H}_s$ denotes the hard thresholding operator and $\mathcal{M} : \mathbb{R}^m \to \mathbb{R}^n$ the median operator, taking the median of the d values connected to each left vertex by the graph Φ. Similarly to E-IHT, we obtain E-NA-ALISTA by replacing the adjoint Φ^T in each iteration by the median operator $\mathcal{M}$. The experimental results in Figure 3 demonstrate that our approach significantly improves signal recovery of generic sparse vectors as well as the convergence speed; see Appendix for implementation details.

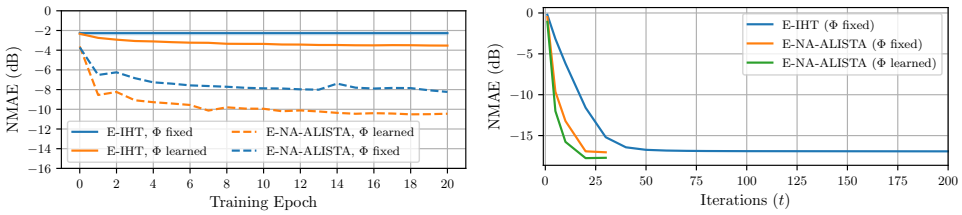


Figure 3: NMAE of compressed sensing with left-d-regular graphs on synthetic data during training with $m = 250, n = 784, s = 50, T = 20$ (left) and evaluation of convergence speed with $m = 250, n = 784, s = 50$ (right).

References

[1] Amir Adler, Michael Elad, and Michael Zibulevsky. Compressed learning: A deep neural network approach. *arXiv preprint arXiv:1610.09615*, 2016.

[2] Freya Behrens, Jonathan Sauder, and Peter Jung. Neurally augmented alista. *arXiv preprint arXiv:2010.01930*, 2020.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[4] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[6] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[7] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[8] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[9] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 25(2):83–91, 2008.

[10] Simon Foucart and Holger Rauhut. An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, pages 1–39. Springer, 2013.

[11] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

[12] Emil Julius Gumbel. Statistical theory of extreme values and some practical applications. *NBS Applied Mathematics Series*, 33, 1954.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Iris AM Huijben, Bastiaan S Veeling, and Ruud JG van Sloun. Deep probabilistic subsampling for task-adaptive compressed sensing. In *International Conference on Learning Representations*, 2019.

[15] Iris AM Huijben, Bastiaan S Veeling, and Ruud JG van Sloun. Learning sampling and model-based signal recovery for compressed sensing mri. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8906–8910. IEEE, 2020.

[16] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[19] Jialin Liu and Xiaohan Chen. Alista: Analytic weights are as good as learned weights in lista. In *International Conference on Learning Representations (ICLR)*, 2019.

[20] Michael Lustig, David Donoho, and John M Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[21] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[22] Chris J Maddison, Daniel Tarlow, and Tom Minka. A* sampling. *arXiv preprint arXiv:1411.0030*, 2014.

[23] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799. PMLR, 2014.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[25] Hendrik Bernd Petersen, Bubacarr Bah, and Peter Jung. Practical high-throughput, non-adaptive and noise-robust sars-cov-2 testing. *arXiv preprint arXiv:2007.09171*, 2020.

[26] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017.

[27] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[28] Tim Vieira. Gumbel-max trick and weighted reservoir sampling, 2014.

[29] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[30] Shanshan Wu, Alex Dimakis, Sujay Sanghavi, Felix Yu, Daniel Holtmann-Rice, Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar. Learning a compressed sensing measurement matrix via gradient unrolling. In *International Conference on Machine Learning*, pages 6828–6839. PMLR, 2019.

[31] Xiao Xiang Zhu and Richard Bamler. Tomographic sar inversion by l_1-norm regularization—the compressive sensing approach. *IEEE transactions on Geoscience and Remote Sensing*, 48(10):3839–3846, 2010.

A Implementation Details

In our experimental setup,[1] we optimize Φ and θ in Eq. (3) using the Adam optimizer [17] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ (default in PyTorch [24]) and a fixed learning rate of 0.0002. We use mini-batches of size 512 training samples. The parameters φ are initialized using a standard Gumbel distribution. We find that when learning both Φ and θ, training is more stable when rescaling the Gumbel noise by a factor of 0.01, both for initialization as well as for the Gumbel reparametrizations during training. We keep the softmax temperature in all Gumbel reparametrizations fixed at $\tau = 1$, which has been demonstrated to work well in practice [16].

A.1 Single Pixel Imaging

As the MNIST digits are not very sparse in the image domain, we use bi-orthogonal 2.2 wavelets with one level as sparsifying transform. For this, we denote the wavelet transform as Ψ and reparametrize $\overline{x} = \Psi\hat{x}$, such that the LASSO problem in Eq. (1) yields the synthesis formulation:

$$\min_{\overline{x}} \|\Phi\Psi^*\overline{x} - y\|_2^2 + \lambda\|\overline{x}\|_1. \tag{5}$$

The target loss function during training in Eq. (3) is set to be the squared loss $\mathcal{L}(\hat{x}, x) = \|\hat{x} - x\|_2^2$. Additive i.i.d. Gaussian noise with a signal-to-noise ratio of 40dB is added to the measurement vector y. As a metric, we use the normalized mean squared error (NMSE), defined as:

$$\text{NMSE}(x, \hat{x}) = 10\log_{10}\Big(\frac{\mathbb{E}_x[\|\hat{x} - x\|_2^2]}{\mathbb{E}_x[\|x\|_2^2]}\Big). \tag{6}$$

A.2 Compressed Sensing with Left-d-Regular Graphs

In the experiments on compressed sensing with left-d-regular graphs, the performance is evaluated on synthetic generic sparse vectors. The support of the synthetic data is generated using i.i.d. Bernoulli random variables, and the non-zero components are drawn from a normal distribution. We use $\mathcal{L}(x, \hat{x}) = \|x - y\|_1$ as the loss function, as the mean absolute error is the usual choice in expander theory. The experiments in this setup are conducted with heavy tailed noise (student t-distributed with 1 degree of freedom) and a signal-to-noise ratio of 40dB. As a metric, we employ the normalized mean absolute error (NMAE), defined as:

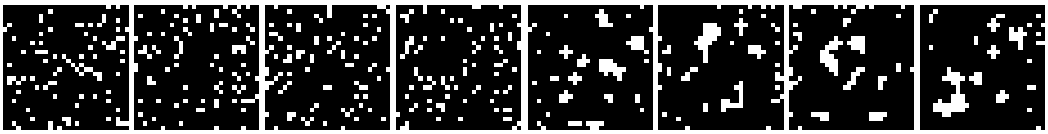
$$\text{NMAE}(x, \hat{x}) = 10\log_{10}\Big(\frac{\mathbb{E}_x[\|\hat{x} - x\|_1]}{\mathbb{E}_x[\|x\|_1]}\Big). \tag{7}$$

B Computational Cost

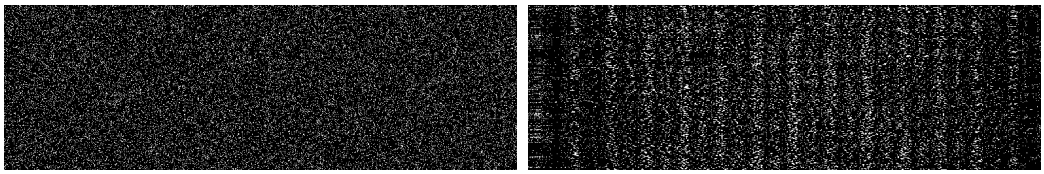
In Algorithm 1, φ has $m \cdot n$ learnable parameters, which must be stored in memory during training time. This is identical to the cost of learning a dense sensing matrix without constraints. At test time, a single mask is sampled after adding Gumbel noise and performing the top-K, and then kept fixed. This means that there is no additional computational cost compared to using a conventional (randomly chosen, but fixed) matrix satisfying the constraints. Hence, our method is feasible for training and comes at no additional expenses during testing or deployment.

[1] The full code of our implementation will be made publicly available upon final publication.

C Supplementary Experimental Results



(a) Random Masks (b) Learned Masks



(c) Random Φ (d) Learned Φ

(e) Reconstruction (random Φ) (f) Reconstruction (learned Φ)

Figure 4: Results from the single pixel imaging setup for $m = 250$ measurements and $d = 90$ ones per row (i.e., ones per mask). The masks correspond to the first four rows of Φ. The reconstructions are obtained by $T = 20$ IHT-iterations.

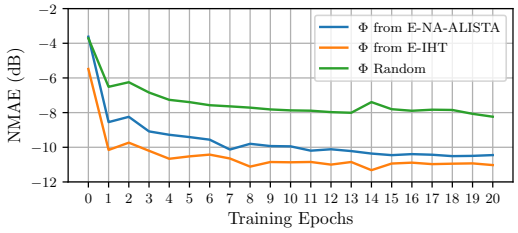


Figure 5: Comparison of training E-NA-ALISTA using a random but fixed left-d-regular graph (green), a fixed left-d-regular graph previously learned by E-IHT with $T = 16$ iterations (orange), and one previously learned by E-NA-ALISTA with $T = 16$ iterations (blue). This highlights that the Φ learned for E-IHT generalizes to E-NA-ALISTA, meaning that Φ fits to this compressed sensing task without overfitting to the reconstruction algorithm.

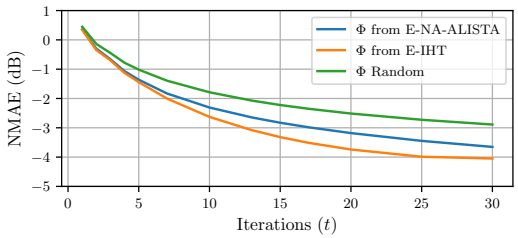


Figure 6: Comparison of E-IHT after varying the number of iterations t using a random but fixed left-d-regular graph (green), a fixed left-d-regular graph previously learned by E-IHT with $T = 16$ iterations (orange), and one previously learned by E-NA-ALISTA with $T = 16$ iterations (blue). This shows that our approach is able to learn a Φ suitable for the task, but does not overfit to the reconstruction algorithm at hand.