ACTIVE DEEP PROBABILISTIC SUBSAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Subsampling a signal of interest can reduce costly data transfer, battery drain, radiation exposure and acquisition time in a wide range of problems. The recently proposed Deep Probabilistic Subsampling (DPS) method effectively integrates subsampling in an end-to-end deep learning model, but learns a static pattern for all datapoints. We generalize DPS to a sequential method that actively picks the next sample based on the information acquired so far; dubbed *Active-DPS* (A-DPS). We validate that A-DPS learns an optimal sampling pattern in a challenging toy problem and improves over DPS for MNIST classification at high subsampling rates. We observe that A-DPS learns to actively adapt based on the previously sampled elements, yielding different sampling sequences across the dataset. Finally, we demonstrate strong performance in active acquisition Magnetic Resonance Image (MRI) reconstruction, outperforming DPS and other deep learning methods.

1 INTRODUCTION

Present-day technologies produce and consume vast amounts of data. Such data is typically acquired using an analog-to-digital converter (ADC). The amount of data digitized by such ADCs is determined not only by the temporal sampling rate of such ADCs, but also by the manner in which spatial acquisitions are taken, e.g. by using a specific design of sensor arrays.

Reducing the number of sample acquisitions needed can lead to meaningful reductions in scanning time, e.g. in Magnetic Resonance Imaging (MRI), radiation exposure, e.g. in Computed Tomography (CT), battery drain, and bandwidth requirements. While the Nyquist theorem is traditionally used to provide theoretical bounds on the sampling rate, in recent years signal reconstruction from sub-Nyquist sampled data has been achieved through a framework called Compressive Sensing (CS).

First proposed by Donoho (2006), and later applied for MRI by Lustig et al. (2007), CS leverages structural signal priors, specifically sparsity under some known transform. By taking compressive measurements followed by the optimization of a linear system under said sparsity prior, reconstruction of the original signal is possible while sampling at sub-Nyquist rates. Researchers have employed CS with great success in a wide variety of applications, such as radar (Baraniuk & Steeghs, 2007; Ender, 2010), seismic surveying (Herrmann et al., 2012), spectroscopy (Sanders et al., 2012), and medical imaging (Han et al., 2016; Lai et al., 2016).

The practical applicability of CS is hampered by a number of factors. First, a sparsifying basis of the data has to be known *a priori*. Second, typical CS reconstruction algorithms are iterative, making them inherently slow. Both issues have been overcome by the introduction of deep learning reconstruction models such as U-net (Ronneberger et al., 2015) and unfolded proximal gradient networks (Mardani et al., 2018). Third, the (typically random) nature of the adopted measurement matrix in CS does, despite adhering to the given assumptions, not necessarily result in an optimal measurement given the underlying data statistics and the downstream system task. Therefore, algorithms that learn the sampling scheme from a training distribution of data have recently been proposed.

In general, these data-driven algorithms can be divided into two categories: algorithms that learn sampling schemes which are fixed once learned (Huijben et al., 2020a;b;c; Ravishankar & Bresler, 2011; Sanchez et al., 2020; Bahadir et al., 2019; Bahadir et al., 2020; Weiss et al., 2019), and algorithms that learn active sampling schemes (Zhang et al., 2019; Jin et al., 2019), which pick new samples based on sequential acquisition of the information. While the former are easier to train and implement in hardware, the latter promises lower achievable sampling rates and better performance

on multimodal data, thanks to the ability to adapt the acquisition to incoming data. Fixed-once-learned sampling schemes on the other hand learn to perform well on average across the entire dataset, without the flexibility to adjust to new data.

One example of the former category is the Deep Probabilistic Subsampling (DPS) framework (Huijben et al., 2020a). DPS allows for joint optimization of a task-adaptive sampling scheme with a downstream neural network trained to perform a specific task, e.g. classification or reconstruction. Joint training of the sampling pattern and the downstream task network was shown to provide better results than disjoint training (Huijben et al., 2020a;b).

In this work we propose an extension of DPS that allows it to not only be fully task-adaptive but also active, by making the sampling procedure iterative and conditional on the samples already acquired. We refer to this method as Active Deep Probabilistic Subsampling (A-DPS). Our main contributions are as follows:

- We introduce A-DPS, a generalization of DPS that enables active acquisition.
- We show the wide applicability of A-DPS through three distinct experiments.
- We demonstrate the superior performance of A-DPS by comparing it with other state-of-the-art models for learning sub-Nyquist sampling regimes.

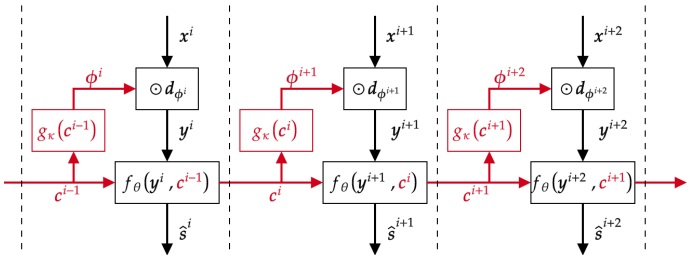


Figure 1: Architectural overview for 3 iterations of A-DPS, with extensions of DPS shown in red.

2 Related work

Recently, several techniques for learning a fixed sampling pattern have been proposed, especially in the field of MR imaging, in which Ravishankar & Bresler (2011) were one of the firsts. In this work, the authors make use of non-overlapping cells in k-space, and move samples between these cells. During training Ravishankar & Bresler (2011) alternate between reconstruction and relocation of sampling positions. After a reconstruction step they sort the cells in terms of reconstructing error and an infinite-p norm. Selected samples from lower scoring cells are relocated to higher scoring cells in a greedy fashion. Sanchez et al. (2020) also propose a greedy approach, in which samples are not relocated between cells, but greedily chosen to optimize a reconstruction loss on a batch of examples. Both of the types of greedy optimization do however not allow for joint learning of sampling together with a downstream reconstruction/task model, as the reconstruction has to either be parameter-free or pretrained to work well with a variety of sampling schemes.

Bahadir et al. (2019) on the other hand propose to learn the sampling pattern by thresholding pixel-based i.i.d. samples drawn from a uniform distribution, dubbed Learning-based Optimization of the Under-sampling PattErn (LOUPE). The sample rate of LOUPE is indirectly controlled by promoting sparsity through the use of an ℓ_1 penalty on the thresholds.

Active sampling schemes have also been proposed. Zhang et al. (2019) propose an active acquisition scheme for MRI by leveraging a reconstruction and adversarial neural network. Whereas the reconstruction network is trained to reconstruct MR images from the subsampled Fourier space (k-space), the adversarial network is trained to distinguish between already sampled, and omitted lines in this space. The k-space line that is most believed to be 'fake' (i.e. filled in by the reconstruction network) by the adversarial network, is sampled next. However, This framework only works for undersampled Fourier to image reconstruction tasks, as the discriminator requires mappings of the

image in k-space. Jin et al. (2019) also put forth an active acquisition scheme for MRI by leveraging reinforcement learning. Two neural networks, one for sampling and one for reconstruction are trained jointly using a Monte-Carlo tree search, resulting in a sampling policy that is dependent on the current reconstruction of the image.

We differentiate our work from projection matrix methods such as (Carson et al., 2012; Li et al., 2013), whose aim it is to learn linear projection matrices (as commonly used in CS), rather than discrete sampling patterns. Even though subsampling is an extreme form of data compression, we differentiate from typical data compression architectures like deep encoder-decoder structures (Theis et al., 2017; Ballé et al., 2017), as these methods do not reduce data rates at the measurement stage. The feedback recurrent autoencoder proposed by Yang et al. (2020) is however related to A-DPS through its use of a recurrent context. But whereas Yang et al. (2020) learn a recurrent context to inform the encoder stage of the network, A-DPS uses this to inform the sampling pattern.

3 METHOD

3.1 GENERAL FRAMEWORK

Given a prediction task s we are interested in learning to predict an optimal subsampling scheme $\boldsymbol{A} \in \{0,1\}^{M \times N}$ (with $M \ll N$) on an input signal $\boldsymbol{x} \in \mathbb{R}^N$, resulting in a measurement $\tilde{\boldsymbol{y}} \in \mathbb{R}^M$:

$$\tilde{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{x}. \tag{1}$$

Each row in $\boldsymbol{A}$ is constrained to have ℓ_0-norm of 1, while each column in $\boldsymbol{A}$ is constrained to have an ℓ_0-norm of either 0 or 1, i.e. each of the N candidate samples is selected at most once. In the rest of this paper we will index these candidate samples with $n \in \{1, \ldots, N\}$, and the selected samples with $m \in \{1, \ldots, M\}$. The percentage of selected samples from the candidate samples is called the sampling ratio $r = M/N \cdot 100\%$.

We also introduce a non-compressed form of the measurement $\tilde{\boldsymbol{y}}$, called $\boldsymbol{y} \in \mathbb{R}^N$, that contains $N - M$ zeros, and M non-zeros at the sampled indices specified by $\boldsymbol{A}$, i.e. the *masked* input. This way, the location of samples from $\boldsymbol{x}$ is preserved, which is especially useful when $\boldsymbol{A}$ changes during training. To acquire $\boldsymbol{y}$ from $\boldsymbol{x}$, one seeks a subsampling mask $\boldsymbol{d}$ that can be applied on $\boldsymbol{x}$ via:

$$\boldsymbol{y} = \boldsymbol{d} \odot \boldsymbol{x} = \boldsymbol{A}^T \boldsymbol{A}\boldsymbol{x}, \tag{2}$$

where $\odot$ denotes an element-wise multiplication. From the resulting measurement $\boldsymbol{y}$ we then aim at predicting the downstream task s through:

$$\hat{s} = f_\theta(\boldsymbol{y}), \tag{3}$$

where $f_\theta(.)$ is a function that is differentiable with respect to its input and parameters θ, e.g. a neural network. Normally, optimization of the task model $f_\theta(.)$ is achieved through backpropagation of some loss function $\mathcal{L}(s, \hat{s})$. However, calculating gradients on the sampling matrix is blocked by its combinatorial nature, inhibiting joint training of the task with the sampling operation. The DPS framework provides a solution to this problem, on which we will elaborate in the next section.

3.2 DPS: DEEP PROBABILISTIC SUBSAMPLING

To enable joint training of the sampling operation with the downstream task model, Huijben et al. (2020a) introduce DPS. Rather than optimizing $\boldsymbol{A}$ directly, they propose to optimize a generative sampling model $P(\boldsymbol{A}|\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ are learned unnormalized logits of (possibly multiple) categorical distribution(s). Each distribution expresses the probabilities for sampling any of the elements x_n from $\boldsymbol{x}$ through sampling matrix $\boldsymbol{A}$. More specifically, $\phi_{m,n}$ is the log-probability for setting $a_{m,n} = 1$, and thus sampling x_n as m^{th} sample.

To generate a sampling pattern from these unnormalized logits, i.e. implementation of this conditional model, the Gumbel-max trick is leveraged (Gumbel, 1954). In the Gumbel-max trick the unnormalized logits are perturbed with i.i.d. Gumbel noise samples $e_{m,n} \sim \text{Gumbel}(0,1)$. By selecting the maximum of this perturbation a realization of the sampling mask can be found using:

$$\boldsymbol{A}_{m,:} = \text{one-hot}_N \left\{ \underset{n}{\text{argmax}} \left\{ w_{m-1,n} + \phi_{m,n} + e_{m,n} \right\} \right\}, \tag{4}$$

where $\boldsymbol{A}_{m,:}$ denotes the m-th row of $\boldsymbol{A}$ and one-hot$_N$ creates a one-hot vector of length N, with the one at the index specified by the argmax operator. Moreover, the cumulative mask $w_{m-1,n} \in \{-\infty, 0\}$ masks previously selected samples by adding minus infinity to those logits, thereby ensuring sampling without replacement.

During backpropagation, gradients are computed by relaxing this sampling procedure using the Gumbel-softmax trick (Jang et al., 2016; Maddison et al., 2017), resulting in:

$$\nabla_{\phi_m} \boldsymbol{A}_{m,:} := \nabla_{\phi_m} \mathbb{E}_{e_m} \left[\text{softmax}_\tau \left\{ w_{m-1,n} + \phi_{m,n} + e_{m,n} \right\} \right], \tag{5}$$

where τ denotes the temperature parameter of the softmax operator. Setting $\tau > 0$ results in a smoothed sampling matrix $\boldsymbol{A}$ (i.e. elements can have values between 0 and 1 as well), allowing gradients to distribute over multiple logits during training. In the limit of $\tau \to 0$ the softmax operator approaches the one-hot argmax function of equation (4). Although this approach – also known as straight-through Gumbel-softmax – leads to biased gradients, it has been shown to work well in practice, and Huijben et al. (2020a) keep τ at a fixed value during training.

Huijben et al. (2020a) propose two regimes of DPS. First, Top-1 sampling, an expressive form of DPS where each of the M selected samples are separately conditioned on all N candidate samples, resulting in $M \times N$ trainable logits $\phi_{m,n}$. Second, Top-M sampling (called Top-K in their paper), a constrained form where all M samples together are conditioned on all N candidate samples, i.e. the logits ϕ_n are shared between the M rows of $\boldsymbol{A}$, resulting in only N trainable logits. While Top-1 sampling is more expressive, Huijben et al. (2020a) noticed slightly better results for the Top-M regime, possibly thanks to the smaller amount of trainable logits, therefore facilitating optimization. Mainly for scaleability reasons, we thus choose to continue with Top-M sampling in this work and refer to this regime as DPS in the rest of this paper. We refer the reader to Huijben et al. (2020a) for more details regarding DPS.

3.3 A-DPS: Active Deep Probabilistic Subsampling

We have seen how DPS enables the learning of a sampling scheme that selects M out of N samples. However, these samples are selected simultaneously. A-DPS selects its samples in an iterative fashion, separating the logits into I time steps, i.e. $\boldsymbol{\phi}^i$ with $i \in \{1, 2, \ldots, I\}$ and $I = M$.

Active acquisition is then achieved by introducing dependency between samples, i.e. the sampling distribution at time step i should depend on the information acquired in previous time steps. To that end, we introduce a context vector $\boldsymbol{c_i}$, that accumulates information from all previous time steps. We then condition the sampling distribution on this context by learning a transformation $\boldsymbol{\phi} = g_\kappa(\boldsymbol{c})$, where $g_\kappa(.)$ is a function that is differentiable with respect to its input and parameters κ. Thus instead of optimizing the parameters directly (as DPS does), we optimize $g_\kappa(\boldsymbol{c})$, which we will refer to as the sampling model.

The question then arises how to best generate this context from previous samples. Here, we follow the *analysis-by-synthesis* principle, and let the analysis (the sampling model) depend on the synthesis (the task model). This way, the task model can inform the sampling model what information it needs to achieve its assigned task. We propose to leverage a recurrent neural network structure in the form of a Long Short-Term Memory (LSTM) cell (Hochreiter & Schmidhuber, 1997). The iterative *analysis-by-synthesis* scheme of A-DPS is formalized as follows:

$$\boldsymbol{\phi}^i = g_{\boldsymbol{\kappa}}(\boldsymbol{c}^{i-1}), \tag{6}$$

$$\hat{s}^i, \boldsymbol{c}^i = f_{\boldsymbol{\theta}}(\boldsymbol{y}^i, \boldsymbol{c}^{i-1}) = f_{\boldsymbol{\theta}}(\boldsymbol{d}_{\boldsymbol{\phi}^i} \odot \boldsymbol{x}^i, \boldsymbol{c}^{i-1}). \tag{7}$$

We visualize the architecture of the A-DPS framework in Fig.1 and discuss the computational complexity of A-DPS in Appendix A. We study an ablated model without the LSTM in Appendix B.

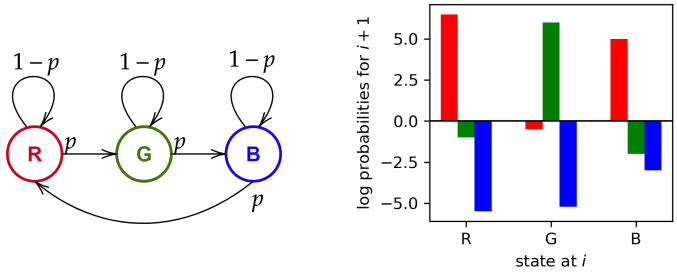
4 Experiments

To show the wide applicability of A-DPS we will evaluate its performance in a series of experiments. In Section 4.1 we provide a proof of principle by demonstrating the ability of A-DPS to optimally adapt its sampling pattern based on the incoming data in a challenging toy problem. In Section

4.2 we will compare A-DPS with DPS at different subsampling ratios on an MNIST classification example, and lastly, we will compare A-DPS with contemporary CS and deep learning methods on an MRI example in Section 4.3.

4.1 TOY EXAMPLE

Experiment setup We first demonstrate the augmented capabilities of A-DPS through a challenging toy problem, designed in such a way that it can not be solved without adequate active sampling. We synthetically generate a stream of data using a hidden Markov model. As shown in Fig. 2a, the model consists of three states: Red, Green, and Blue. From every state there is a probability p to go to the next state, and probability $1-p$ to stay there. Every state has an associated output vector containing 3 elements: $R = [2, -1, -1]$, $G = [-1, 2, -1]$, $B = [-1, -1, 2]$.. A sequence $\boldsymbol{sq}$ of arbitrary length can be generated form the given Markov model, e.g.: $\boldsymbol{sq} = [R, R, G, B, B]$.



a b

Figure 2: a) Hidden Markov model used to generate data for the toy example. b) Generated log probabilities (logits) of the sampling model based on the previous state of the hidden Markov Model.

We are now interested in a recurrent task model that scans over this sequence, classifying every segment to one of the three states. Using its memory of previous states, the task model gets to sample only one position out of every three, i.e. it observes only one color channel. Note that in the proposed hidden Markov model, only 2 out of 3 elements of any given state vector are fully informative, namely, the element associated with the current state, and the element associated with the next state. As knowledge of either of these two elements allows one to tell what will happen at the next step. The third candidate element however, does not contain such information, and should not be selected for sampling. The challenge for the network thus lies in actively identifying which elements are actually informative, and adapting when the hidden Markov model progresses.

Task model The A-DPS network is implemented following the architecture shown in Fig.1. The task model $f_{\boldsymbol{\theta}}(.)$ consists of an LSTM cell of one layer with a hidden size of 20. The output of this LSTM is then used as input for two fully connected layers (20 and 3 nodes, respectively), activated using a ReLU and softmax function, respectively. The sampling model $g_{\boldsymbol{\kappa}}(.)$ also takes the output of the LSTM as input. It consists of two fully connected layers with 20 and 3 nodes, respectively, and a ReLU activation function only after the first layer.

Training details We train the model to minimize the categorical cross-entropy between the predicted labels and the ground truth labels. The temperature parameter was set to 1 and the probability p of the hidden Markov model was set to 0.2. We optimize using stochastic gradient descent (SGD) on sequences of length 100. The model is trained using the Adam solver (lr = $5e-3$, $\beta_1 = 0.9$, $\beta_2 = 0.9$, and $\epsilon = 1e-7$) (Kingma & Ba, 2015) for a total of 200 sequences.

Results After training, the network achieves an accuracy of 99.5%. Note that it can never reach the theoretical maximum of 100% due to the random nature of the first sample in a sequence. The network achieves this high accuracy by successfully exploiting prior knowledge when selecting a new sample. As can be seen from Fig. 2b, A-DPS learns to select an informative sample based on the current state.

4.2 MNIST

Experiment setup Classification performance at different sampling rates was tested on the MNIST database (LeCun et al., 1998), consisting of 70,000 grayscale images of 28×28 pixels of handwritten digits between 0 and 9. We keep the original data split of 60,000 training and 10,000 testing examples. We train both DPS top-M and A-DPS to take partial measurements in the pixel-domain at different sampling rates.

Reaching back to Fig.1 and equations (6) and (7), DPS top-M sampling only consists of the sampling and task model ($f_{\boldsymbol{\theta}}(.)$) parts. All M samples are selected at the same time and used once by $f_{\boldsymbol{\theta}}(.)$ to predict which digit the network is looking at. In the case of A-DPS however, only 1 sample is taken at a time and used as input for $f_{\boldsymbol{\theta}}(.)$. Here, $f_{\boldsymbol{\theta}}(.)$ also creates a context that is used by the sampling network $g_{\boldsymbol{\kappa}}(.)$ to select the next sample. A-DPS iterates through this loop M times in order to select all the samples. We keep $f_{\boldsymbol{\theta}}(.)$ the same for both DPS and A-DPS. Resulting in the fact that the last iteration of A-DPS is similar to that of DPS top-M (i.e. M samples are selected and fed through $f_{\boldsymbol{\theta}}(.)$).

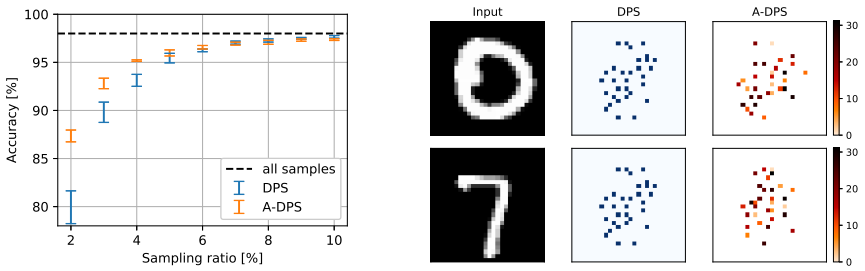
Task model In the classification network $f_{\boldsymbol{\theta}}(.)$, all 784 ($28 \times 28$) zero-masked samples are used as input for 4 fully connected layers. The fully connected layers have 784, 256, 128, and 128 nodes, respectively. Moreover, they are all activated by leaky ReLU activation functions with a negative slope of 0.2. The first three layers also have a dropout of 30%.

The output of the last fully connected layer is then fed into an LSTM with a hidden size of 128. The output of the LSTM is both used as the context for the sampling network, as well as the input for one final fully connected layer. This final fully connected layer maps the output of the LSTM into the 10 class labels, and is therefore followed by a softmax activation function.

In case of A-DPS the output of the LSTM is also used as the input for the sampling network $g_{\boldsymbol{\kappa}}(.)$. The sampling network consists of two linear layers with output sizes of 256 and 784, respectively. Moreover, after the first layer a leaky ReLU activation function is used with a negative slope of 0.2, and a dropout of 30% is applied.

Training details Both sampling strategies were trained to minimize the categorical cross-entropy loss. The temperature parameter was fixed to 2. We again employ SGD with the Adam solver (lr = $2e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.9$, and $\epsilon = 1e-7$) to minimize the loss function. Training was performed on batches of 256 examples for 50 epochs.

Results The resulting accuracy on the test set is shown in Fig. 3a. A-DPS outperforms DPS especially when the actual amount of samples taken is very low. It is hypothesized that it is especially important to select those candidate samples that carry a lot of of information based on the previous samples for very low data rates. Two examples of the selected sampling masks at $r = 4\%$ are displayed in Fig. 3b. Here, it is shown how DPS selects all samples at once, while A-DPS selects them in an iterative fashion, resulting in different sampling patterns for the two examples.



a b

Figure 3: Results for the MNIST experiment. a) Classification accuracy on the test set for different sampling ratios. Confidence bounds of one standard deviation are shown. The 'all samples' line indicates the accuracy achieved without subsampling. b) Comparison of the sampling masks for two examples at $r = 4\%$. The colors denote the order in which the samples were taken.

4.3 MRI

Experiment setup To show the applicability of A-DPS, we demonstrate its performance on line-based MRI. We make use of the NYU fastMRI database of knee MRI volumes (Zbontar et al., 2018). Only the single-coil measurements were selected, from which the outer slices were removed. The resulting data was split into 8,000 training, 2,000 validation, and 3,000 testing MRI slices. All slices were cropped to the central 208×208 pixels and normalized between 0 and 1. The subsampling operation on one of these MRI slices is then performed in k-space (Fourier-space):

$$\boldsymbol{Y} = |\mathcal{F}^H \boldsymbol{D} \odot \mathcal{F}\boldsymbol{X}|, \tag{8}$$

where $|.|$ is the magnitude operator. Moreover, $\boldsymbol{X} \in \mathbb{R}^{N \times N}$ is the fully sampled ground truth image and $\boldsymbol{Y} \in \mathbb{R}^{N \times N}$ is the subsampled image, both in the pixel domain. In this case N is equal to 208. Furthermore, $\mathcal{F}$ and $\mathcal{F}^H$ denote the forward and inverse 2D-Fourier transform, respectively. $\boldsymbol{D} \in \{0, 1\}^{N \times N}$ denotes the sampling mask in k-space.

Normally $\boldsymbol{Y}$ would be complex, due to the asymmetrical nature of MRI measurements and the incomplete subsampling mask. Here, we choose to take the magnitude of Y to simplify reconstruction. We hypothesize that doing so does not significantly change the problem, as the imaginary part of fully sampled images in the NYU fastMRI dataset is very small compared to the real part.

Task model To reconstruct an estimate of the original image $\hat{\boldsymbol{X}}$ from the partial measurement $\boldsymbol{Y}$ a deep unfolded proximal gradient method is used (Mardani et al., 2018), in which K iterations of a proximal gradient method are unfolded as a feed forward neural network following:

$$\hat{\boldsymbol{X}}^{(k+1)} = \mathcal{P}^{(k)}_{(\zeta)} \left\{ \hat{\boldsymbol{X}}^{(k)} - \alpha^{(k)}_{(\psi)} \left(|\mathcal{F}^H \boldsymbol{D} \odot \mathcal{F}\hat{\boldsymbol{X}}^{(k)}| - \boldsymbol{Y} \right) \right\}, \tag{9}$$

where $\mathcal{P}^{(k)}_{(\zeta)}(.)$ is a trainable image-to-image proximal mapping and $\alpha^{(k)}_{(\psi)}$ is the step size, parameterized by ζ and ψ, respectively. We implement this proximal gradient method for $k = 3$ steps, with the trainable step size $\alpha^{(k)}_{(\psi)}$ implemented as a 3×3 convolutional layer. Each proximal mapping is implemented as a series of 4 convolutions with 16, 16, 16, and 1 feature(s) each and a kernel size of 3×3. All convolutions but the last are followed by ReLU activation functions.

We will compare A-DPS to to several relevant sampling baselines, namely, random uniform, low-pass, variable density (VDS), greedy mask selection (Sanchez et al., 2020), LOUPE (Bahadir et al., 2019; Bahadir et al., 2020), and DPS. While we also would have liked to compare A-DPS to other active sampling strategies, all previously mentioned methods (Jin et al., 2019; Zhang et al., 2019) have unfortunately not made their code available.

Under a random uniform regime all N lines are equally likely to be sampled, while under a low-pass regime the M lines closest to the DC frequency will be selected. VDS on the other hand is a heuristic regime that employs a probability density from which the desired amount of samples are drawn. Following (Lustig et al., 2007), we here use a polynomial probability density function with a decay factor of 6.

For the greedy mask selection we follow the approach by Sanchez et al. (2020) and first optimize the sampling mask using the NESTA solver (Becker et al., 2011). After this, we fix the sampling mask and train our proximal gradient network. Results for both reconstruction algorithms are reported.

To generate a sampling mask $\boldsymbol{D}$ using A-DPS we use a sampling network $g_{\boldsymbol{\kappa}}(.)$. This network takes two inputs. Firstly, the output of the proximal network, i.e. the reconstructed image at that iteration. This image is analyzed using 3 convolutional layers with kernels sizes of 3×3 followed by ReLU activation functions. The output features are of sizes 16, 16, and 32, respectively. The final feature map is aggregated into a feature vector using global average pooling. Next to this feature vector, the indices of the selected lines at the previous iteration(s) are also used as input for the sampling network, encoded as a M-hot vector of dimension 208. Both the feature and selected indices vector are concatenated and used as input for an LSTM cell, with output dimension of 208. This is followed by two fully connected layers with 208 neurons each, having a ReLU activation after the first layer.

Training details To promote the reconstruction of visually plausible images, we leverage both a Mean Squared Error (MSE) and adversarial loss (Ledig et al., 2016). To that end we introduce

a discriminator network that is trained to distinguish between real and reconstructed MR images. The discriminator is implemented using three convolutional layers with kernel sizes of 3×3, stride 2, and 64 feature maps, each with Leaky ReLU activations. After the last convolutional layer the feature maps are aggregated into a feature vector using global average pooling, with a dropout rate of 40%, which is mapped to a single output probability using one fully connected layer followed by a sigmoid activation function. Next to the MSE loss and adversarial loss, we add a third loss term that penalizes the MSE loss between the discriminator features of real and generated images. The total loss function is a weighted summation of these three losses, with weights 1, $5e-6$, and $1e-7$, respectively.

All sampling mask selection strategies were then trained using SGD on batches of 4 images for a total of 10 epochs. We again employ the Adam solver (lr = $2e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.9$, and $\epsilon = 1e-7$) to minimize the loss function, and set the temperature parameter to 2. We choose $M = 26$, which results in a sampling ratio of $r = 12.5\%$, or an acceleration factor of 8.

Results We score the different strategies based on 3 metrics: the normalized mean square error (NMSE), the peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM) (Wang et al., 2004). Averaged results over 5 runs are shown in Table 1. An example of A-DPS reconstruction is shown in Fig. 4, while a comprehensive overview of all baselines for this example can be found in Appendix D.

Table 1: Averaged results over five runs on the hold-out test set for different subsampling strategies used to select $M = 26$ k-space lines.

Sampling Model	Reference	NMSE	PSNR	SSIM
Random uniform		0.324	16.0	0.47
Low pass		0.043	25.2	0.71
Variable density	(Lustig et al., 2007)	0.041	25.4	0.73
Greedy Mask NESTA	(Sanchez et al., 2020)	0.043	24.8	0.71
Greedy Mask Prox. Grad.		0.041	25.4	0.73
LOUPE	(Bahadir et al., 2019)	0.045	25.7	0.74
DPS	(Huijben et al., 2020a)	0.039	25.7	0.74
A-DPS (proposed)		**0.036**	**26.1**	**0.75**

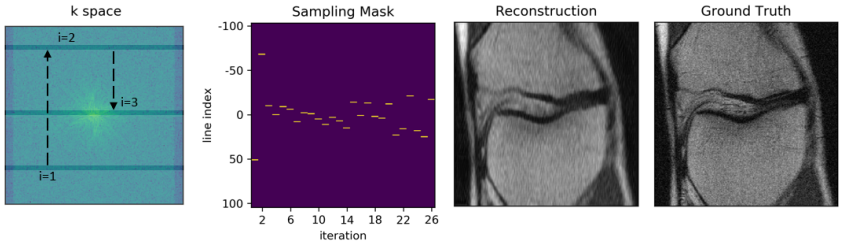


Figure 4: A-DPS MR image reconstruction of an test-set image by adaptively sampling 26 k-space lines. From left to right: 1) k-space with the first 3 sampled lines, 2) sequence of line indices selected by A-DPS, 3) reconstructed image, 4) fully sampled MR image (ground truth).

5 Conclusion

In this paper we proposed a generalization of DPS, which enables active acquisition, called A-DPS. We showed its applicability on a wide variety of tasks. Moreover, we found that A-DPS outperforms other sampling pattern selection methods on downstream task performance. Future work includes exploration on how to further improve conditioning of the sampling scheme on earlier acquired information and meta-information (such as resolution, sampling ratio, and weighting). Potential future applications include 3D and dynamic MRI, CT, ultrasound, radar, video, and MIMO systems.

REFERENCES

Cagla D. Bahadir, Adrian V. Dalca, and Mert R. Sabuncu. Learning-based Optimization of the Under-sampling Pattern in MRI. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11492 LNCS:780–792, 1 2019.

Cagla D. Bahadir, Alan Q. Wang, Adrian V. Dalca, and Mert R. Sabuncu. Deep-learning-based optimization of the under-sampling pattern in mri. *IEEE Transactions on Computational Imaging*, 6:1139–1152, 2020.

Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end Optimized Image Compression. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 11 2017.

Richard Baraniuk and Philippe Steeghs. Compressive radar imaging. In *IEEE National Radar Conference - Proceedings*, pp. 128–133. Institute of Electrical and Electronics Engineers Inc., 2007. ISBN 1424402840. doi: 10.1109/RADAR.2007.374203.

Stephen Becker, Jérôme Bobin, and Emmanuel J. Candès. NESTA: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 1 2011. ISSN 19364954. doi: 10.1137/090756855.

William R Carson, Minhua Chen, Miguel R D Rodrigues, Robert Calderbank, and Lawrence Carin. Communications-Inspired Projection Design with Application to Compressive Sensing. *SIAM Journal on Imaging Sciences*, 5(4):1185–1212, 10 2012. doi: 10.1137/120878380.

David L Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*, 52(4):1289, 2006. doi: 10.1109/TIT.2006.871582.

Joachim H.G. Ender. On compressive sensing applied to radar. *Signal Processing*, 90(5):1402–1414, 5 2010. ISSN 01651684. doi: 10.1016/j.sigpro.2009.11.009.

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

Yo Seob Han, Jaejun Yoo, and Jong Chul Ye. Deep Residual Learning for Compressed Sensing CT Reconstruction via Persistent Homology Analysis. 11 2016.

Felix J. Herrmann, Michael P. Friedlander, and Özgür Yilmaz. Fighting the curse of dimensionality: Compressive sensing in exploration seismology. *IEEE Signal Processing Magazine*, 29(3):88–100, 2012. ISSN 10535888. doi: 10.1109/MSP.2012.2185859.

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

Iris A. M. Huijben, Bastiaan S. Veeling, and Ruud J. G. Van Sloun. Deep Probabilistic Subsampling for Task-Adaptive Compressed Sensing. In *International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia, 2020a.

Iris A. M. Huijben, Bastiaan S. Veeling, and Ruud J. G. Van Sloun. Learning sub-sampling and signal recovery with applications in ultrasound imaging. *IEEE Transactions on Medical Imaging*, 2020b.

Iris A. M. Huijben, Bastiaan S. Veeling, and Ruud J. G. Van Sloun. Learning Sampling and Model-Based Signal Recovery for Compressed Sensing MRI. In *International Conference on Acoustics, Speech and Signal Processing , ICASSP 2020*, Barcelona, Spain, 2020c.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 11 2016.

Kyong Hwan Jin, Michael Unser, and Kwang Moo Yi. Self-Supervised Deep Active Accelerated MRI. 1 2019.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 12 2015.

Zongying Lai, Xiaobo Qu, Yunsong Liu, Di Guo, Jing Ye, Zhifang Zhan, and Zhong Chen. Image reconstruction of compressed sensing MRI using graph-based redundant wavelet transform. *Medical Image Analysis*, 27:93–104, 1 2016. ISSN 13618423. doi: 10.1016/j.media.2015.05.012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998. ISSN 00189219. doi: 10.1109/5.726791.

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *CVPR*, 2017-January:105–114, 9 2016.

G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai. On projection matrix optimization for compressive sensing systems. *IEEE Transactions on Signal Processing*, 61(11):2887–2898, 2013.

Michael Lustig, David Donoho, and John M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 12 2007. ISSN 07403194. doi: 10.1002/mrm.21391.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Machine Learning*, 2017.

Morteza Mardani, Qingyun Sun, Shreyas Vasawanala, Vardan Papyan, Hatef Monajemi, John Pauly, and David Donoho. Neural Proximal Gradient Descent for Compressive Imaging. 6 2018.

Saiprasad Ravishankar and Yoram Bresler. Adaptive sampling design for compressed sensing mri. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3751–3755, 2011.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, volume 9351, pp. 234–241. Springer, 2015. ISBN 9783319245737. doi: 10.1007/978-3-319-24574-4_28.

Thomas Sanchez, Baran Gozcu, Ruud B. van Heeswijk, Armin Eftekhari, Efe Ilicak, Tolga Cukur, and Volkan Cevher. Scalable Learning-Based Sampling Optimization for Compressive Dynamic MRI. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8584–8588, Barcelona, Spain, 4 2020. doi: 10.1109/icassp40776.2020.9053345.

Jacob N. Sanders, Semion K. Saikin, Sarah Mostame, Xavier Andrade, Julia R. Widom, Andrew H. Marcus, and Alán Aspuru-Guzik. Compressed sensing for multidimensional spectroscopy experiments. *Journal of Physical Chemistry Letters*, 3(18):2697–2702, 9 2012. ISSN 19487185. doi: 10.1021/jz300988p.

Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy Image Compression with Compressive Autoencoders. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 3 2017.

Laurens Van Der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Zhou Wang, Alan C Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *Search Results Web results IEEE Transactions on Image Processing*, 13(4), 2004.

Tomer Weiss, Ortal Senouf, Sanketh Vedula, Oleg Michailovich, Michael Zibulevsky, and Alex Bronstein. PILOT: Physics-Informed Learned Optimized Trajectories for Accelerated MRI. 9 2019.

Y. Yang, G. Sautière, J. J. Ryu, and T. S. Cohen. Feedback recurrent autoencoder. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3347–3351, 2020.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Matthew J. Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, James Pinkerton, Duo Wang, Nafissa Yakubova, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastmri: An open dataset and benchmarks for accelerated MRI. *CoRR*, abs/1811.08839, 2018.

Zizhao Zhang, Adriana Romero, Matthew J. Muckley, Pascal Vincent, Lin Yang, and Michal Drozdzal. Reducing Uncertainty in Undersampled MRI Reconstruction with Active Acquisition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:2049–2053, 2 2019.

A COMPUTATIONAL COMPLEXITY

One of the drawbacks of A-DPS compared to learned fixed sampling schemes is it higher amount of computational complexity. The main source of this complexity is the unrolling of iterations, leading to a computational complexity of $O(I) = O(M/\rho)$. Although we set ρ equal to 1 in all our experiments, one can in fact seamlessly interpolate between A-DPS and DPS by choosing $1 \leq \rho \leq M$. This constitutes a trade-off between computational complexity and adaptation rate. We leave further exploration of this trade-off to future work.

We can also express computational complexity in terms of run-time on a machine, in our case a GeForce GTX 1080 Ti. A comparison of DPS and A-DPS in terms of training time per epoch can be seen in Fig. 5. We can see that the training time for A-DPS increases for higher sampling ratios where it needs to unroll through more iterations. By combining the results from Fig. 3a and Fig. 5 one can make a trade-off between run-time and accuracy. Where A-DPS achieves higher accuracy for stricter sampling regimes, while at the same time not increasing run-times by a lot.

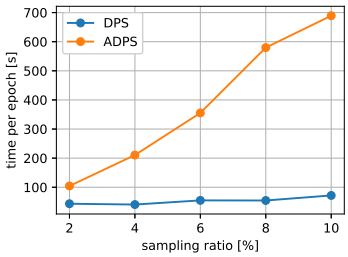


Figure 5: Comparison between DPS and A-DPS of time taken to train for one epoch for the MNIST example on a GeForce GTX 1080 Ti.

For the MRI experiment the training times per epoch are 2 and 52 minutes, for DPS and A-DPS, respectively. Inference is however fast: A-DPS only requires $\sim 13\,ms$ of processing time to determine the next-to-acquire K-space line and reconstruct an image after each step. This is well below the shortest reported Time of Echo (TE) for this MRI acquisition, being 27 ms (Zbontar et al., 2018).

B LSTM ABLATION STUDY

To study the effect of the LSTM, we here perform an ablation study by replacing it with a simple fully connected (FC) layer for the Toy and MNIST examples. Doing so resulted in unstable training for the toy example, see Fig. 6 left. While it was still possible to for one of the fully connected runs to achieve the 100% accuracy, in general the fully connected model became 'stuck' between 60% and 80% accuracy. Performing the ablation on MNIST resulted in lower down-stream accuracy on the stricter sampling regimes, see Fig. 6 right.

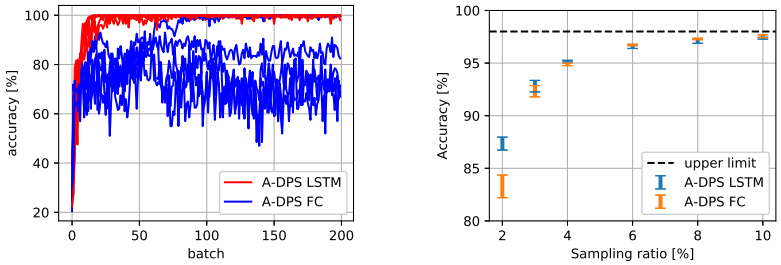
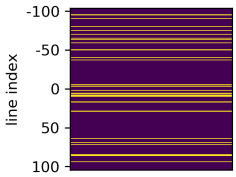
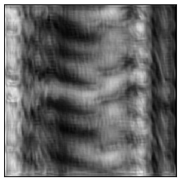
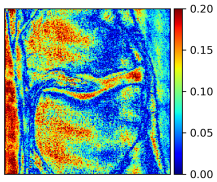
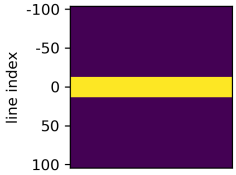
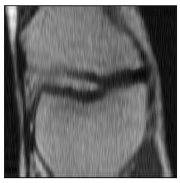
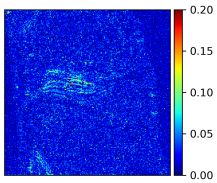
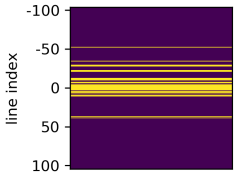
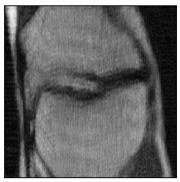
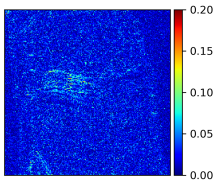
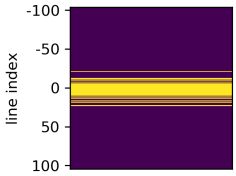
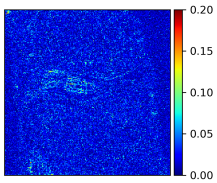
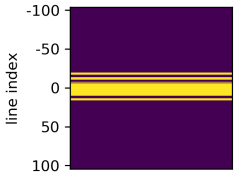
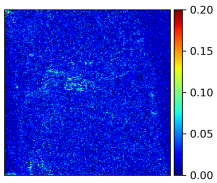
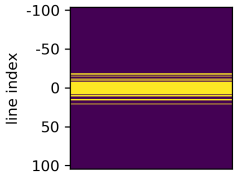
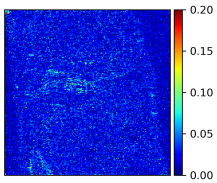
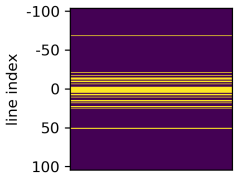
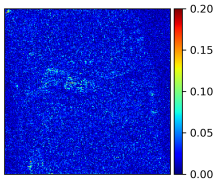


Figure 6: Left: training curves for 5 runs of the toy example, the training becomes noisy and less reliable without the use of the LSTM. Right: Resulting accuracy confidence bounds of one standard deviation are shown over 5 runs for A-DPS on MNIST using either an LSTM or a Fully Connected (FC) layer to create the context vector $\boldsymbol{c}$.

C MRI RECONSTRUCTION EXAMPLES

Method	Sampling Mask	Reconstruction	Error Map
Ground Truth			
Random Uniform			
Low Pass			
Variable Density			
Greedy Mask Selection			
LOUPE			
DPS			
A-DPS			

D ANALYSIS OF SAMPLING PATTERNS

To analyze the sampling patterns learned by A-DPS for the MNIST experiment we employ t-SNE (Van Der Maaten & Hinton, 2008). The t-SNE plot of all sampling patterns on the test set for $r = 4\%$ is shown in Fig. 7, where the colors indicate the ground truth label (digit). The clustering in this figure indicates how similar selected sampling patterns are. For example, the sampling patterns for zero tend to be dissimilar from all the other sampling patterns, while the digits four, six, and nine get more similar sampling patterns from A-DPS.

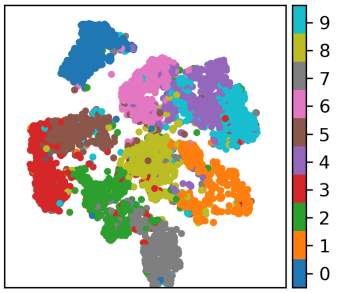


Figure 7: The t-SNE plot of the sampling patterns colored by digit shows that A-DPS actively adapts the pattern based on earlier observations.

To analyse the distribution of sampling schemes learned by A-DPS for MRI we plot the relative occurrence of the different line indices in Fig. 8 on the left side. We also employ t-SNE to cluster the sampling schemes, which is shown in Fig. 8 on the right. Unlike the MNIST experiment, there are no class labels available to color.

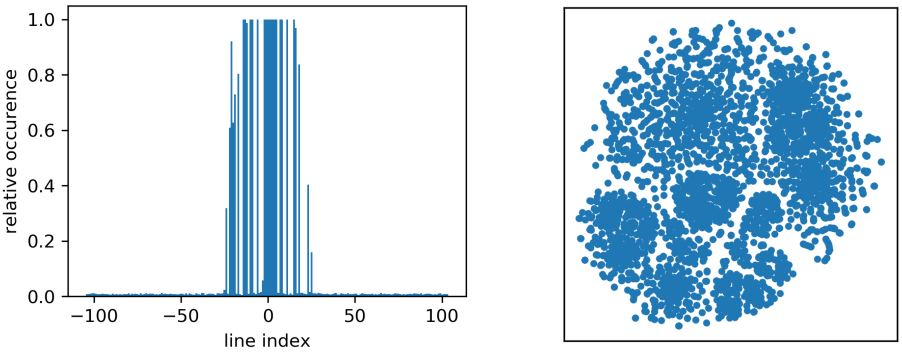


Figure 8: Left: Relative occurrence of the line indices in A-DPS sampling schemes for MRI. Right: t-SNE embedding of the A-DPS sampling schemes for MRI.