DEEP PROBABILISTIC SUBSAMPLING FOR TASK-ADAPTIVE COMPRESSED SENSING

Anonymous authors
Paper under double-blind review

ABSTRACT

The field of deep learning is commonly concerned with optimizing predictive models using large pre-acquired datasets of densely sampled datapoints or signals. In this work, we demonstrate that the deep learning paradigm can be extended to incorporate a subsampling scheme that is jointly optimized under a desired minimum sample rate. We present Deep Probabilistic Subsampling (DPS), a widely applicable framework for task-adaptive compressed sensing that enables end-to-end optimization of an optimal subset of signal samples with a subsequent model that performs a required task. We demonstrate strong performance on reconstruction and classification tasks of a toy dataset, MNIST, and CIFAR10 under stringent subsampling rates in both the pixel and the spatial frequency domain. Due to the task-agnostic nature of the framework, DPS is directly applicable to all real-world domains that benefit from sample rate reduction.

1 INTRODUCTION

In many real-world prediction problems, acquiring data is expensive and often bandwidth constrained. Such is the case in regimes as medical imaging (Lustig et al., 2007; Choi et al., 2010; Chernyakova & Eldar, 2014), radar (Baraniuk, 2007), and seismic surveying (Herrmann et al., 2012). By carefully reducing the number of samples acquired over time, in pixel-coordinate-space or in k-space, efficient subsampling schemes lead to meaningful reductions in acquisition time, radiation exposure, battery drain, and data transfer.

Subsampling is traditionally approached by exploiting expert knowledge of the signal of interest. Famously, the Nyquist theorem states that when the maximum frequency of a continuous signal is known, perfect reconstruction is possible when sampled at twice this frequency. More recently, it has been shown that if the signal is sparse in (a known) domain, sub-Nyquist rate sampling can be achieved through compressive measurements and subsequent optimization of a linear system under said sparsity prior; a framework known as compressed sensing (CS) (Donoho et al., 2006; Eldar & Kutyniok, 2012; Baraniuk, 2007).

CS methods, however, are lacking in the sense that they do not focus solely on the information required under a given data distribution to solve the downstream task of interest, such as disease prediction or semantic segmentation. Formalizing such knowledge is challenging in its own right and would require careful analysis for each modality and downstream task. In this work, we propose to explore the deep learning hypothesis as a promising alternative: reducing the need for expert knowledge in lieu of large data-sets and end-to-end optimization of neural networks.

As subsampling is non-differentiable, incorporating it into an end-to-end optimized deep learning model is non-trivial. Here, we take a probabilistic approach: rather than learning a subsampling scheme directly, we pose a probability distribution that expresses belief over effective subsampling patterns and optimize the distribution's parameters instead. To enable differentiable sampling from this distribution, we leverage recent advancements in a continuous relaxation of this sampling process, known as Gumbel-softmax sampling or sampling from a concrete distribution (Jang et al., 2017; Maddison et al., 2016). This enables end-to-end training of both the subsampling scheme and the downstream model.

Naively, the number of parameters of a distribution over an n-choose-k problem scales factorially, which is intractable for all practical purposes. We propose a novel, expressive yet tractable, pa-

rameterization for the subsampling distribution that conditions on the output sample index. We hypothesize that this conditioning prevents redundant sampling, as it enables modeling the scenario in which multiple candidate samples can be equally good, yet redundant in combination. Further, we investigate a parameter-restricted approach with a single parameter per candidate sample, balancing tractability and exploration. In this case, we adopt a continuous relaxation of top-K sampling to guarantee differentiability (Kool et al., 2019; Plötz & Roth, 2018).

Our main contributions are as follows:

- DPS: A new regime for task-adaptive subsampling using a novel probabilistic deep learning framework for incorporating a sub-Nyquist sampling scheme into an end-to-end network
- DPS enables joint optimization of a subsampling pattern together with a predictive model for a variety of downstream tasks, without the need for explicit knowledge on a sparsifying basis.
- We demonstrate improved performance over strong subsampling baselines in image classification and reconstruction in both Fourier and pixel space.

2 Related work

Some recent works have proposed deep-learning-based subsampling methods for fast MRI, historically being one of the most prominent applications of CS. Weiss et al. (2019) exploit gradient backpropagation to a fixed set of real-valued coordinates, enabled by their subsequent (limited-support) interpolation on the discrete k-space grid, and Bahadir et al. (2019) formulate the sampling problem by learning pixel-based thresholding of i.i.d. samples drawn from a uniform distribution, referred to as LOUPE. Where the former suffers from limited exploratory capabilities (likely due to its compact support), with learned sampling schemes typically not deviating far from their initialization, the latter controls the sample rate only indirectly, through the addition of a sparsity-promoting ℓ_1 penalty on the sampling mask.

Related to our methodology yet different in purpose, Plötz & Roth (2018); Xie & Ermon (2019); Kool et al. (2019) leverage an extension to the Gumbel-max trick (Gumbel, 1954) for (ordered) subset selection using a categorical distribution with $N-1$ free parameters. They rely on relaxed top-K sampling, as first proposed by Vieira (2014). These methods build upon the continuous relaxation of the categorical distribution known in the deep learning community as the Gumbel-softmax trick or the concrete distribution (Jang et al., 2017; Maddison et al., 2016). In this work, we investigate the use of Gumbel-based relaxations for task-adaptive compressive sampling, and further propose a novel parametrization where we condition the sampling distribution on the output sample index.

We differentiate our contribution from deep encoder-decoder methods for data compression (Baldi & Hornik, 1989; Hinton & Zemel, 1993; Blier & Ollivier, 2018; Habibian et al., 2019), which do not aim at reducing data rates already at the sensing and digitization stage. Related work by Mousavi et al. (2019) and Wu et al. (2019), focusses on the problem of learning compressive linear encoders/filters, rather than discrete subsampling as addressed here.

Through the lens of contemporary deep learning, subsampling can be interpreted as a form of *attention* (Bahdanau et al., 2014; Kim et al., 2017; Parikh et al., 2016; Vaswani et al., 2017). Rather than attending on intermediate representations, our model "attends" directly on the input signal. For subsampling to be effective, sparse weights are essential. In the space of attention, this is known as *hard* attention (Xu et al., 2015), and is typically optimized using the REINFORCE gradient estimator (Williams, 1992). In contrast to the method of attention as applied in these works, our method aims for a fixed, reduced subsampling rate.

3 Method

3.1 Task-adaptive system model

We consider the problem of performing some downstream task s through a learned subsampling scheme $\mathbf{A}$, resulting in measurements $\mathbf{y} \in \mathbb{R}^M$ of an underlying fully-sampled signal $\mathbf{x} \in \mathbb{R}^N$, with

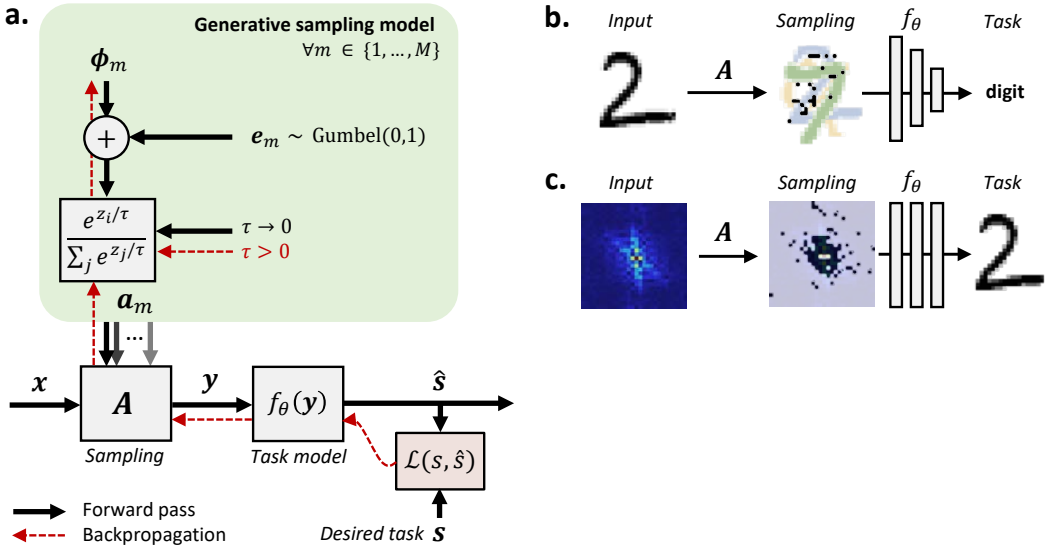


Figure 1: (a) System-level overview of the proposed framework, in which a probabilistic generative sampling model and a subsequent task model are jointly trained to fulfill a desired system task. (b,c) Two illustrative task-based sampling paradigms: image classification from a partial set of pixels (b), and image reconstruction from partial Fourier measurements (c), respectively.

$M << N$:

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{1}$$

where $\mathbf{A} \in \{0,1\}^{M \times N}$ is the subsampling measurement matrix. We here concern ourselves specifically with scenarios in which the rows of $\mathbf{A}$ are constrained to having cardinality one, i.e. $||a_m||_0 = 1, \forall m \in \{1, ..., M\}$. In such cases, $\mathbf{A}$ serves as a subset selector, sampling M out of N elements in $\mathbf{x}$. From the resulting low-rate measurements $\mathbf{y}$ we aim at performing task s through:

$$\hat{s} = f_\theta(\mathbf{y}), \tag{2}$$

with $f_\theta(\cdot)$ being a function that is differentiable with respect to its input and parameters θ, e.g. a neural network.

Given a downstream task and dataset, we are interested in learning both the optimal processing parameters θ, and the sampling scheme as described in eq. (1). To circumvent the non-differential nature of discrete sampling, we will introduce a novel fully probabilistic sampling strategy that allows for gradient-based learning through error backpropagation, on which we detail in the following section.

3.2 DPS: Deep probabilistic sampling

Figure 1 shows a schematic overview of the proposed framework. Since direct optimization of the elements in $\mathbf{A}$ is intractable due to its combinatorial nature, we here instead propose to leverage a tractable generative sampling model that is governed by a learned subsampling distribution, parameterized by $\mathbf{\Phi}$:

$$\mathbf{A} \sim P(\mathbf{A}|\mathbf{\Phi}). \tag{3}$$

Thus, rather than optimizing $\mathbf{A}$, we optimize the distribution parameters $\mathbf{\Phi}$. To warrant sufficient expressiveness while maintaining tractability, we learn the parameters $\phi_m \in \mathbb{R}^N$ of M independent categorical distributions (rather than the joint distribution, which scales factorialy), being the rows of $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$.

Formally, we define each mth measurement row $\mathbf{a}_m \in \{0,1\}^N$ in $\mathbf{A}$ as a one-hot encoding of an independent categorical random variable $\mathrm{r}_m \sim \mathrm{Cat}(N, \boldsymbol{\pi}_m)$. We define $\boldsymbol{\pi}_m \in \mathbb{R}^N = \{\pi_{m,1}, \ldots, \pi_{m,N}\}$, being a vector containing N class probabilities, and parameterize it in terms

of its unnormalized logits $\phi_{m,n}$ such that:

$$\pi_{m,n} = \frac{\exp \phi_{m,n}}{\sum_{i=1}^{N} \exp \phi_{m,i}}. \tag{4}$$

We sample from $\text{Cat}(N, \boldsymbol{\pi}_m)$ by leveraging the Gumbel-max trick (Gumbel, 1954), a reparameterization of the sampling procedure. It perturbs the unnormalized logits $\phi_{m,n}$ with i.i.d. Gumbel noise[1] samples $e_{m,n} \sim \text{Gumbel}(0, 1)$. The argmax of these perturbed logits, results in a sample from the original categorical distribution $\text{Cat}(N, \boldsymbol{\pi}_m)$. A realization $\tilde{\mathrm{r}}_m$ is then defined as:

$$\tilde{\mathrm{r}}_m = \underset{n}{\text{argmax}}\{w_{m,n} + \phi_{m,n} + e_{m,n}\}, \qquad m \in \{1, \ldots, M\}, \tag{5}$$

in which $w_{m,n} \in \{-\infty, 0\}$ for $n \in \{1 \ldots N\}$ mask previous realizations $\tilde{\mathrm{r}}_1, \cdots, \tilde{\mathrm{r}}_{m-1}$, by adding $-\infty$ to the logit of the previously selected category, enforcing sampling without replacement among the M distributions. Introducing the function $\text{one_hot}_N(\cdot)$ as the operator that returns a one-hot vector of length N, we finally obtain:

$$\mathbf{a}_m = \text{one_hot}_N\{\tilde{\mathrm{r}}_m\} = \text{one_hot}_N\Big\{\underset{n}{\text{argmax}}\{w_{m,n} + \phi_{m,n} + e_{m,n}\}\Big\}. \tag{6}$$

To permit error backpropagation for efficient optimization of $\boldsymbol{\Phi}$, we require $\nabla_{\boldsymbol{\phi}_m} \mathbf{a}_m$ to exist $\forall m \in \{1, \ldots, M\}$. Since $\text{argmax}(\cdot)$ is a non-differentiable operator, we adopt the Straight-Through Gumbel Estimator (Jang et al., 2017; Maddison et al., 2016) as a surrogate for $\nabla_{\boldsymbol{\phi}_m} \mathbf{a}_m$:

$$\begin{aligned} \nabla_{\boldsymbol{\phi}_m} \mathbf{a}_m &\coloneqq \nabla_{\boldsymbol{\phi}_m} \mathbb{E}_{\mathbf{e}_m}\big[\text{softmax}_\tau(\mathbf{w}_m + \boldsymbol{\phi}_m + \mathbf{e}_m)\big] = \\ &\nabla_{\boldsymbol{\phi}_m} \mathbb{E}_{\mathbf{e}_m}\left[\frac{\exp\{(\mathbf{w}_m + \boldsymbol{\phi}_m + \mathbf{e}_m)/\tau\}}{\sum_{i=1}^{N} \exp\{(w_{m,i} + \phi_{m,i} + e_{m,i})/\tau\}}\right], \end{aligned} \tag{7}$$

with (row operator) $\text{softmax}_\tau(\cdot)$ as a continuous differentiable approximation of the one-hot encoded $\text{argmax}(\cdot)$ operation. Appendix A shows the full derivation of $\nabla_{\boldsymbol{\phi}_m} \mathbf{a}_m$.

We refer to sampling using the $\text{softmax}_\tau(\cdot)$ function as soft sampling. Its temperature parameter τ serves as a gradient distributor over multiple entries (i.e. logits) in $\boldsymbol{\phi}_m$. Using a relatively high value enables updating of multiple logits during training, even though a hard sample was taken in the forward pass. In the limit of $\tau \to 0$, soft sampling approaches the one-hot encoded $\text{argmax}(\cdot)$ operator in eq. (6) (Jang et al., 2017; Maddison et al., 2016).

To study the trade-off of parameter efficiency and convergence, we further study a more parameter-restricted version of DPS that shares the weights $\boldsymbol{\phi}_m$ across all M distributions. This allows for a more memory-efficient implementation, by leveraging the Gumbel top-K trick (Kool et al., 2019) (rather than Gumbel-max), and its corresponding relaxation (Plötz & Roth, 2018). We refer to this particular model as DPS-topK, and adopt DPS-top1 to indicate the sampling model presented in equations 6 and 7, where weights are not shared across distributions, and one sample is drawn from each distribution. Algorithm 1 in appendix B describes the DPS algorithm.

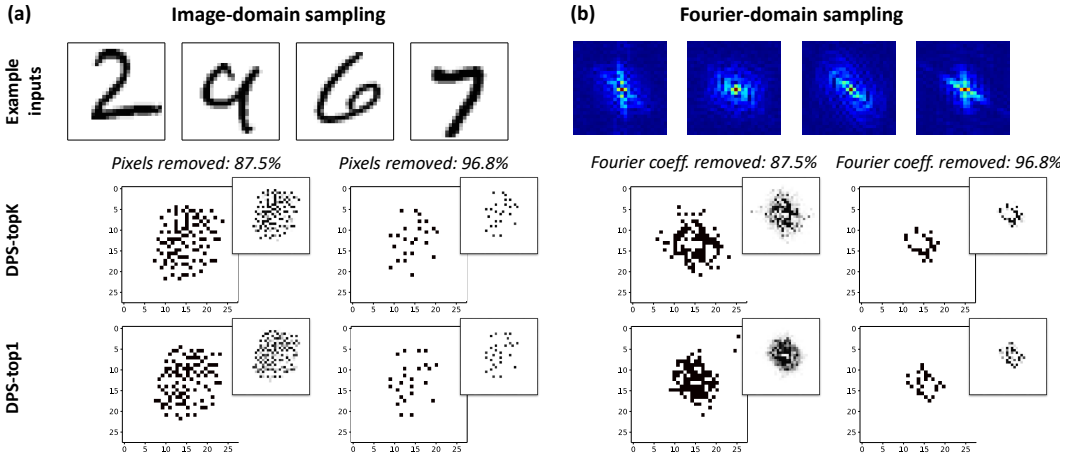
4 Experiments

We test the applicability of the proposed task-adaptive DPS framework for three datasets and two distinct tasks: image classification and image reconstruction. We explore subsampling in pixel coordinate space as well as in k-space. The latter is relevant for scenarios in which data is acquired in the frequency domain, such as (but not limited to) magnetic resonance imaging (MRI).

4.1 MNIST classification

Experiment setup Classification performance was tested on the MNIST database (LeCun et al., 1998), comprising 70,000 28×28 grayscale images of handwritten digits 0 to 9. We split the dataset

[1]The Gumbel distribution is typically used to model the maximum of a set of independent samples. Its probability density function is of the form: $f(x) = \frac{1}{\beta} e^{-z-e^{-z}}$, with $z = \frac{x-\mu}{\beta}$, where μ and β are the mean and standard deviation, respectively. The standard Gumbel distribution, with $\mu = 0$ and $\beta = 1$ is adopted in the Gumbel-max trick.



	Perc. error:	Perc. error:	Perc. error:	Perc. error:
DPS-topK	**2.1%**	**6.6%**	1.3%	**1.3%**
DPS-top1	2.6%	9.4%	**1.2%**	1.8%
LOUPE	**2.1%**	19.9%	6.2%	32.0%
Uniform	4.2%	28.6%	9.1%	54.5%
Disk	6.3%	27.3%	1.3%	1.7%

Figure 2: MNIST classification for (a) image-domain, and (b) Fourier-domain subsampling. (top) Several example images in the respective sampling domains, (2nd and 3rd row) learned task-adaptive (DPS-topK and DPS-top1, respectively) subsampling patterns, with their relative sample incidence across a 1000 such realizations (inset), and (bottom) hold-out classification results of the proposed DPS methods, compared to the LOUPE baseline, and two non-learned baseline sampling approaches.

into 50,000 training images, 5,000 validation, and 5,000 test images. We train DPS-top1 and DPS-topK to take partial measurements in either the image or Fourier domain image, and process them through the task model to yield a classification outcome. We compare our results to those obtained using uniformly distributed pixel/Fourier samples, a sampled disk/low pass filter, and the data-driven LOUPE method (Bahadir et al., 2019).

Besides, to show the benefit of jointly training sampling with a task, we add a case in which we disjointly learn the classification task from the sampling pattern. As such, we train DPS-topK for MNIST reconstruction. The output of this network is fed to the same task network, i.e. classifier, as used in the other experiments. This classifier is subsequently trained without additional subsampling.

Task model After sampling M elements, all N zero-masked samples (or $2N$ in the case of complex Fourier samples) are passed through a series of 5 fully-connected layers, having N, 256, 128, 128 and 10 output nodes, respectively. The activations for all but the last layer were leaky ReLUs, and 20% dropout was applied after the first three layers. The 10 outputs were normalized by a softmax function to yield the respective classification probabilities. Table 1 in app. C shows a detailed overview of the complete architecture of the task mode. Zero-filling and connecting all possible samples, rather than only connecting the M selected samples, facilitated faster co-adaptation of the network to different sampling patterns during training.

Training details We train the network to maximize the log-likelihood of the observations $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{s}_i) \mid i \in 0, \ldots, L\}$ through minimization of the categorical cross-entropy between the predictions and the labels, denoted by $\mathcal{L}_s$. We moreover promote training towards one-hot distributions $\text{Cat}(N, \boldsymbol{\pi}_m)$ by penalizing high entropy adopting:

$$\mathcal{L}_e = -\sum_{m=1}^{M}\sum_{n=1}^{N} \pi_{m,n} \log \pi_{m,n}, \tag{8}$$

with $\pi_{m,n}$ defined as in eq. (4). The total optimization problem is thus:

$$\hat{\boldsymbol{\Phi}}, \hat{\theta} = \underset{\boldsymbol{\Phi},\theta}{\text{argmin}} \Big\{ \mathbb{E}_{(\mathbf{x},\mathbf{s})\sim p_{\mathcal{D}}} \mathcal{L}_s + \mu \mathcal{L}_e \Big\}, \tag{9}$$

where $p_{\mathcal{D}}$ is the data generating distribution. Penalty multiplier μ was set to linearly increase $1e{-}5$ per epoch, starting from 0.0. Increasing this entropy penalty enforces one-hot distributions, and therefore less variations in the realizations, i.e. sampling patterns, when training evolves. The temperature parameter τ in eq. (7) was set to 2, and the sampling distribution parameters $\mathbf{\Phi}$ were initialized randomly, following a zero-mean Gaussian distribution with standard deviation 0.25. Equation 9 was optimized using stochastic gradient descent on batches of 32 examples, approximating the expectation by a mean across the train dataset. To that end, we used the ADAM solver ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e{-}7$) (Kingma & Ba, 2014), and we trained until the validation loss plateaued. We adopted different learning rates for the sampling parameters $\mathbf{\Phi}$ and the parameters of the task model θ, being $2e-3$ and $2e-4$, respectively.

Results The results presented in Figure 2a show that image-domain sampling using DPS significantly outperforms the fixed sampling baselines (uniform and disk). The data-driven LOUPE method is outperformed in case of a strong subsampling factor. The resulting patterns qualitatively demonstrate how, for this task, a sensible selection of pixels that are most informative was made (slightly slanted, and capturing discriminative areas). Notably, partial Fourier measurements (Figure 2b) allowed for a much greater reduction of the number samples, with DPS sampling outperforming uniform sampling and LOUPE, and showing similar performance as fixed disk sampling. Interestingly, the DC and very-low frequency components were consistently not selected.

Figure 3 shows the learned sampling pattern (96.8% pixels removed) when training DPS-topK with the task of reconstruction. The adopted reconstruction network architecture is the same as the one used for CIFAR10 reconstruction (see section 4.3). The subsequently trained classifier resulted in a classification error of 9.3%, compared to 6.6% for task-adaptive learning with DPS-topK. This case clearly illustrates the benefit of task-adaptive learning of a sampling pattern.

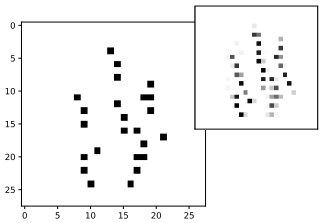


Figure 3: Learned sampling pattern for DPS-topK MNIST reconstruction.

4.2 'Lines and Circles' image reconstruction

Experiment setup To evaluate reconstruction of structured images from highly undersampled partial Fourier (k-space) measurements (keeping 3.1% of the coefficients), we generated synthetic toy data comprising images that each contain up to 5 horizontal lines and randomly-sized circles. Lines and circles were placed at random positions and their pixel intensity was drawn from a uniform distribution between 1 and 10. Examples were generated in an on-line fashion during training. A pre-generated hold-out test set of 1000 randomly generated examples was used for all cases. Two illustrative test examples, along with their Fourier-domain measurement representations, are given in Figure 4 (a,b). We compare the results to those obtained using three fixed partial Fourier measurement baselines, following uniform, random, and low pass subsampling patterns, respectively, and the data-driven LOUPE method.

Task model Image reconstruction from partial Fourier measurements was performed by following the methodology in Zhu et al. (2018). We use a deep neural network consisting of two subsequent fully-connected layers with $tanh$ activations that map the $2N$ (zero-filled) Fourier coefficients (stacked real and imaginary values) to a vector of N pixel values. This vector was subsequently reshaped into a $\sqrt{N} \times \sqrt{N}$ image, and processed by 3 convolutional layers. Table 2 in app. C provides the layer parameters of this task network.

Training details The optimization problem was the same as in Equation 9, however with the loss $\mathcal{L}_s$ defined as the negative log-likelihood:

$$\mathcal{L}_s = \mathbb{E}_{(\mathbf{x},\mathbf{s})\sim p_{\mathcal{D}}} \| f_\theta(\mathbf{A}_{(\mathbf{\Phi})}\mathbf{x}) - \mathbf{s} \|_2^2, \tag{10}$$

where we make the parameterization of $\mathbf{A}$ by $\mathbf{\Phi}$ explicit. The learning rates for $\mathbf{\Phi}$ and θ were $1e-3$ and $1e-4$, respectively, and μ was set to $2e-4$. Training was performed until plateauing of the validation loss. All other hyperparameters were the same as in Sec. 4.1

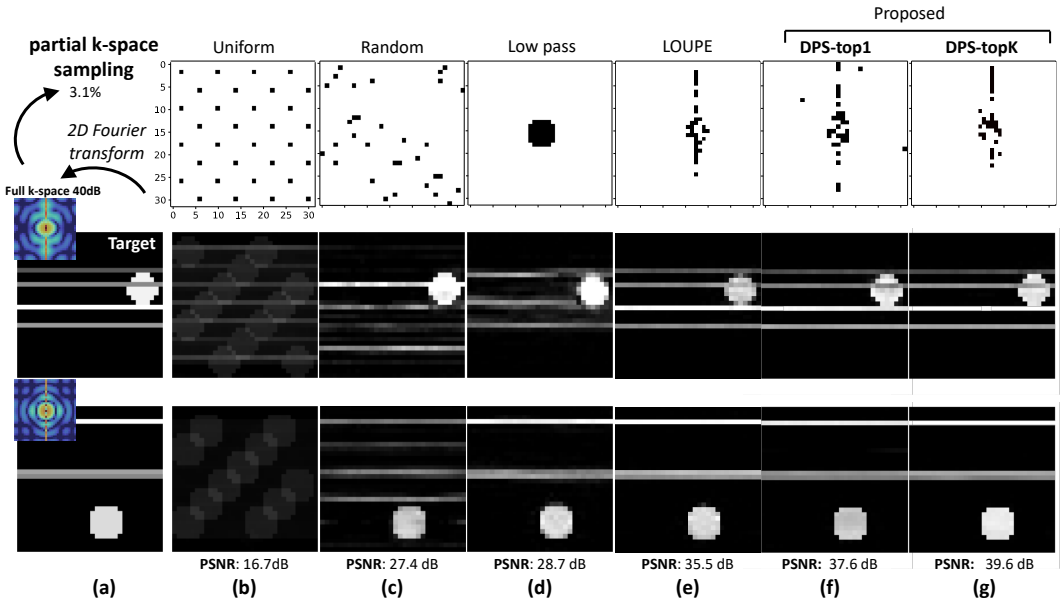


Figure 4: Image reconstruction performance from partial k-space (Fourier) measurements on a custom toy dataset consisting of lines and circles with random locations and sizes. Illustrative examples of the k-space and target images are given in (a). The sampling patterns, reconstructed images and PSNR value (across the entire test set) for the different sampling strategies are displayed in: (b) uniform, (c) random, (d) low pass, (e) LOUPE, (f) DPS-top1, and (g) DPS-topK, respectively. In all cases, only 3.1% of the Fourier coefficients have been selected.

Results An overview of the results is given in fig. 4. As expected, uniform subsampling leads to strong spatial aliasing that can not be recovered by the task model, violating the Nyquist criterion. In turn, random subsampling introduces an incoherent aliasing pattern, that can only in part be recovered. Although not suffering from aliasing, low pass sampling deteriorates resolution, of which the effect is particularly evident for the broadband/sharp horizontal lines. In contrast, data-driven sampling (LOUPE, DPS-top1, and DPS-topK) prevents aliasing by learning an appropriate sampling pattern. DPS sampling specifically, yields high-resolution accurate reconstructions with an improved PSNR evaluated on the entire test set compared to the baselines. Note how the learned sampling patterns (top row) have evolved from a random initialization, similar to that of (c), to the ultimate task-adaptive scheme (e-g).

4.3 CIFAR10 IMAGE RECONSTRUCTION

Experiment setup The CIFAR10 database (Krizhevsky et al., 2009) contains 60,000 images of 32×32 pixels in 10 different classes. We converted all images to grayscale, and subsequently split them into 50,000 training images, 5,000 validation and 5,000 test images. We again learn partial Fourier sampling and image reconstruction, keeping only 12.5% of the Fourier coefficients, and compare ourselves to the three fixed partial Fourier measurement baselines and LOUPE, as described in Sec. 4.2.

Task model The challenging reconstruction task for CIFAR10 motivates the adoption of a structured model to enable strong reconstruction. We draw inspiration from iterative proximal-gradient schemes (Parikh et al., 2014) which are dedicated to solving the ill-posed linear measurement problem in equation 1. To that end, we unfold $K = 5$ such iterations, learning an adequate image-domain proximal mapping $\mathcal{P}_\theta^{(k)}$ and stepsize $\alpha^{(k)}$ at each fold:

$$\hat{\mathbf{s}}^{(k+1)} = \mathcal{P}_\theta^{(k)} \left(\hat{\mathbf{s}}^{(k)} - \alpha^{(k)} \mathbf{F}^H \mathbf{A}_{(\mathbf{\Phi})}^T \left(\mathbf{A}_{(\mathbf{\Phi})} \mathbf{F} \hat{\mathbf{s}}^{(k)} - \mathbf{A}_{(\mathbf{\Phi})} \mathbf{x} \right) \right) \tag{11}$$

where $\mathbf{F} \in \mathbb{C}^{N \times N}$ is a discrete Fourier transform (DFT) matrix, and $(\cdot)^H$ denotes the Hermitian (conjugate transpose). In the above formulation, at each fold a step is taken towards the sampling-

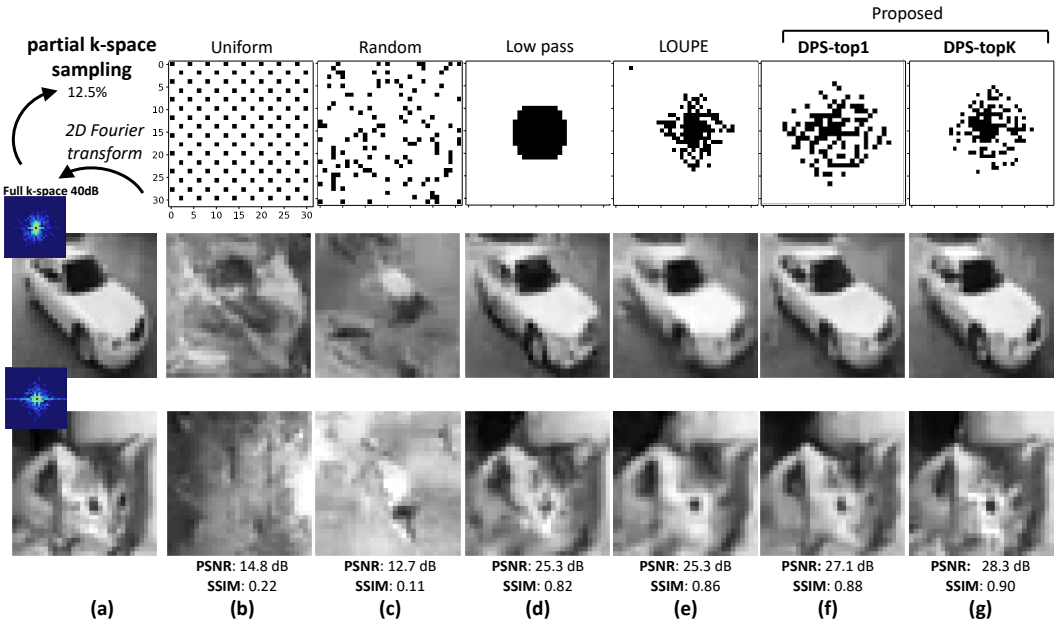


Figure 5: Image reconstruction performance from partial k-space (Fourier) measurements on the CIFAR10 database. Illustrative examples of the k-space and target images are given in (a). The sampling patterns, images and statistical quality metrics for uniform, random, low pass, LOUPE, DPS-top1, and DPS-topK are given in (b-g), respectively. In all cases, 12.5% of the Fourier coefficients have been selected.

consistent subspace that adequately represents the physical measurement of $\mathbf{s}$ by $\mathbf{A}_{(\mathbf{\Phi})}\mathbf{F}$. The trained proximal operator $\mathcal{P}_{\theta}^{(k)}$, a 7-layer convolutional network, then projects this onto the manifold of visually plausible images (Mardani et al., 2018), removing noise, aliasing, or blurring artifacts. Table 3 in appendix C provides provides details on the layers in the task model.

Training details Optimization settings were similar to those in Sec. 4.2, leveraging a mean-squared-error (negative log-likelihood) reconstruction cost and a distribution entropy penalty on $\boldsymbol{\pi}_m$. To promote visually plausible reconstructions, we added an adversarial (Ledig et al., 2017) cost by adopting a discriminator network $D_{\psi}(\mathbf{s})$ that aims to discriminate between images reconstructed from partial Fourier measurements $\hat{\mathbf{s}}$ and actual images $\mathbf{s}$. The discriminator comprised 3 convolutional layers with leaky ReLU activations, followed by global average pooling, 40% dropout, and a logistic binary classification model (parameters are provided in Table 3, app. C). The sampling- and task model parameters were then trained to both minimize the negative log-likelihood cost and maximize the discriminator binary cross-entropy cost $\mathcal{L}_D$, in addition to the entropy penalty $\mathcal{L}_e$:

$$\hat{\mathbf{\Phi}}, \hat{\theta} = \underset{\mathbf{\Phi},\theta}{\operatorname{argmin}} \Big\{ \mathbb{E}_{(\mathbf{x},\mathbf{s})\sim p_{\mathcal{D}}} \|f_{\theta}(\mathbf{A}_{(\mathbf{\Phi})}\mathbf{x}) - \mathbf{s}\|_2^2 + \mu\mathcal{L}_e - \lambda\mathcal{L}_D \Big\}, \tag{12}$$

where λ weighs adherence to the data-driven MSE loss and the discriminator loss. It was empirically set to 0.004. The parameters of the discriminator network, ψ, were jointly optimized with eq. (12) to minimize $\mathcal{L}_D$. The learning rates for $\mathbf{\Phi}$ and θ were $1e-3$ and $2e-4$, respectively, and μ was empirically set to $1e-6$, balancing sharpness and high frequency hallucinations. Training was performed until the validation loss plateaued. All other hyperparameters were set as in Sec. 4.1.

Results Figure 5 shows how task-adaptive sampling significantly outperforms all baselines. Both the uniform and random subsampling schemes suffer from severe aliasing which the task model is not able to adequately restore. While low pass sampling does not lead to aliasing, the absence of real high-frequency information causes the task model to 'hallucinate' such (super-resolution) content. Interestingly, LOUPE (e) learned a rather compact, i.e. around DC frequencies, sampling scheme, resulting in blurry images, but DPS learned a more wide-spread pattern, also capturing higher frequencies. The result of this is particularly notable for the example displayed in the bottom

row of fig. 5, being less structured (or predictable) than that in the row above it. DPS thus enabled high-quality, high-resolution reconstructions that go beyond those obtained with the other methods, reaching a PSNR of 27.1 dB and 28.3 dB, and a structural similarity index (SSIM) of 0.88 and 0.90 across the 5000 test examples, for topK and top1 respectively. For the most competitive methods, i.e. fixed sampling low pass and data-driven LOUPE, these statistics were 25.3 dB and 0.82, and 25.3 dB and 0.86, respectively.

5 Conclusions

We have introduced deep probabilistic sampling (DPS), a framework that enables jointly learning a data- and task-driven sampling pattern, with a subsequent task-performing model. The framework is generic and can be combined with any network architecture that performs the required task using the subsampled set of signal elements. Empirically we find the method to perform strongly on toy datasets and canonical deep learning problems. Further work can explore the effectiveness of DPS in real-world problems such as MRI scanning. Even though LOUPE (Bahadir et al., 2019) showed to find similar subsampling patterns in all experiments, compared to DPS, its performance on the downstreak task showed to be lower. We argue that LOUPE probably suffers from the fact that samples in the training phase are relaxed. As such, the task network is optimized on a relaxed subsampled pattern, whereas a hard thresholded pattern is used for inference on the test set to make fair comparisons to DPS.

In comparing the parameter-restricted DPS-topK versus the more expressive DPS-top1, we find the former to perform better in our experiments. Studying the sampling distributions after training, we find that DPS-top1 exhibits more uncertainty. The insets in figure 2 show this by the higher variance present among the 1000 realizations for DPS-top1 compared to DPS-topK. We tentatively attribute this due to difficulties with convergence. Further work will explore if the DPS-top1's more expressive parametrization can improve over DPS-topK in more complex problems that might benefit from a broader hypothesis space during training, as well as under more carefully tuned softmax temperature annealing schemes.

Our data-driven method does not explicitly need knowledge regarding the, often unknown, sparsifying basis, whereas conventional CS algorithms do require this. Like all data-driven optimized methods, a learned sampling scheme is at risk of overfitting to a small training dataset. Although we did not observe this issue in our experiments, careful regularization might be required to ensure that this effect is minimal in such high-risk tasks. The fully-differentiable DPS framework allows for flexibility, and interesting extensions can be explored in future work. Rather than learning a fixed set of parameters for the subsampling distribution, a neural network can be used to predict the parameters instead, conditioned on contextual data or the samples acquired so far. Finally, our method currently requires the desired sampling rate to be predetermined as a hyperparameter. Future work can explore if this rate can be jointly optimized to incorporate optimization of the subsampling rate.

References

Cagla Deniz Bahadir, Adrian V Dalca, and Mert R Sabuncu. Learning-based optimization of the under-sampling pattern in mri. In *International Conference on Information Processing in Medical Imaging*, pp. 780–792. Springer, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. September 2014.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw.*, 2(1):53–58, January 1989.

Richard G Baraniuk. Compressive sensing. *IEEE signal processing magazine*, 24(4), 2007.

Léonard Blier and Yann Ollivier. The description length of deep learning models. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 2216–2226. Curran Associates, Inc., 2018.

Tanya Chernyakova and Yonina C Eldar. Fourier-domain beamforming: the path to compressed ultrasound imaging. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 61 (8):1252–1267, 2014.

Kihwan Choi, Jing Wang, Lei Zhu, Tae-Suk Suh, Stephen Boyd, and Lei Xing. Compressed sensing based cone-beam computed tomography reconstruction with a first-order method a. *Medical physics*, 37(9):5113–5125, 2010.

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

Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge university press, 2012.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

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

Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. Video compression with Rate-Distortion autoencoders. August 2019.

Felix J Herrmann, Michael P Friedlander, and Ozgur Yilmaz. Fighting the curse of dimensionality: Compressive sensing in exploration seismology. *IEEE Signal Processing Magazine*, 29(3):88–100, 2012.

Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. *NIPS*, 1993.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumbel-softmax. *stat*, 1050:17, 2017.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. Structured attention networks. February 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. *arXiv preprint arXiv:1903.06059*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

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.

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.

Morteza Mardani, Qingyun Sun, David Donoho, Vardan Papyan, Hatef Monajemi, Shreyas Vasanawala, and John Pauly. Neural proximal gradient descent for compressive imaging. In *Advances in Neural Information Processing Systems*, pp. 9573–9583, 2018.

Ali Mousavi, Gautam Dasarathy, and Richard G. Baraniuk. A data-driven and distributed approach to sparse signal representation and recovery. In *ICLR 2019*, 2019. URL https://openreview.net/pdf?id=B1xVTjCqKQ.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. June 2016.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. October 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł Ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

Tim Vieira. Gumbel-max trick and weighted reservoir sampling — graduate descent. https://timvieira.github.io/blog/post/2014/08/01/gumbel-max-trick-and-weighted-reservoir-sampling/, August 2014. Accessed: 2019-9-24.

Tomer Weiss, Ortal Senouf, Sanketh Vedula, Oleg Michailovich, Michael Zibulevsky, and Alex Bronstein. Pilot: Physics-informed learned optimal trajectories for accelerated mri. *arXiv preprint arXiv:1909.05773*, 2019.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3):229–256, May 1992.

Shanshan Wu, Alexandros G Dimakis, Sujay Sanghavi, Felix X Yu, Daniel Holtmann-Rice, Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar. Learning a compressed sensing measurement matrix via gradient unrolling. *International Conference on Machine Learning (ICML)*, 2019.

Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. In *International Joint Conference on Artificial Intelligence*, 2019.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. February 2015.

Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487, 2018.

A GRADIENT OF GUMBEL-SOFTMAX SAMPLING

For any m^{th} pair of rows $(\mathbf{a}_m, \boldsymbol{\phi}_m)$, any n^{th} element $a_{m,n}$ in $\mathbf{a}_m$ can be differentiated towards all elements in $\boldsymbol{\phi}_m$ through:

$$\begin{aligned}
&\nabla_{\boldsymbol{\phi}_m} a_{m,n} \\
&= \nabla_{\boldsymbol{\phi}_m} \mathbb{E}_{\mathbf{e}_m}\Big[\text{softmax}_\tau(\mathbf{w}_m + \boldsymbol{\phi}_m + \mathbf{e}_m)\Big|_n\Big] \\
&= \mathbb{E}_{\mathbf{e}_m}\Big[\nabla_{\boldsymbol{\phi}_m}\text{softmax}_\tau(\mathbf{w}_m + \boldsymbol{\phi}_m + \mathbf{e}_m)\Big|_n\Big] \\
&= \mathbb{E}_{\mathbf{e}_m}\Big[\nabla_{\boldsymbol{\phi}_m} \frac{\exp\{(\mathbf{w}_m + \boldsymbol{\phi}_m + \mathbf{e}_m)/\tau\}}{\sum_{i=1}^N \exp\{(w_{m,i} + \phi_{m,i} + e_{m,i})/\tau\}}\Big|_n\Big]
\end{aligned} \tag{13}$$

Gumbel noise vector $\mathbf{e}_m$ can be reparametrized as a function of uniform noise vector $\boldsymbol{\epsilon}_m \sim \mathcal{U}(0,1)$ i.i.d., through:

$$\mathbf{e}_m = -\log(-\log(\boldsymbol{\epsilon}_m)). \tag{14}$$

This allows rewriting eq. (13) into:

$$\begin{aligned}
&\nabla_{\boldsymbol{\phi}_m} a_{m,n} = \mathbb{E}_{\boldsymbol{\epsilon}_m}\Big[\nabla_{\boldsymbol{\phi}_m} \frac{\exp\{(\mathbf{w}_m + \boldsymbol{\phi}_m - \log(-\log(\boldsymbol{\epsilon}_m)))/\tau\}}{\sum_{i=1}^N \exp\{(w_{m,i} + \phi_{m,i} - \log(-\log(\epsilon_{m,i})))/\tau\}}\Big|_n\Big] \\
&= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathbf{P}\big[\boldsymbol{\epsilon}_m = [k_1, \cdots, k_N]\big] \nabla_{\boldsymbol{\phi}_m} \frac{\exp\{(\mathbf{w}_m + \boldsymbol{\phi}_m - \log(-\log(\boldsymbol{k})))/\tau\}}{\sum_{i=1}^N \exp\{(w_{m,i} + \phi_{m,i} - \log(-\log(k_i)))/\tau\}}\Big|_n dk_N \cdots dk_1 \\
&= \int_{-0}^{1} \cdots \int_{0}^{1} \mathbf{P}\big[\epsilon_{m,1} = k_1\big]\mathbf{P}\big[\epsilon_{m,2} = k_2\big] \cdots \mathbf{P}\big[\epsilon_{m,N} = k_N\big] \cdot \\
&\qquad \nabla_{\boldsymbol{\phi}_m} \frac{\exp\{(\mathbf{w}_m + \boldsymbol{\phi}_m - \log(-\log(\boldsymbol{k})))/\tau\}}{\sum_{i=1}^N \exp\{(w_{m,i} + \phi_{m,i} - \log(-\log(k_i)))/\tau\}}\Big|_n dk_N \cdots dk_1 \\
&= \int_{-0}^{1} \cdots \int_{0}^{1} 1 \cdot \nabla_{\boldsymbol{\phi}_m} \frac{\exp\{(\mathbf{w}_m + \boldsymbol{\phi}_m - \log(-\log(\boldsymbol{k})))/\tau\}}{\sum_{i=1}^N \exp\{(w_{m,i} + \phi_{m,i} - \log(-\log(k_i)))/\tau\}}\Big|_n dk_N \cdots dk_1.
\end{aligned} \tag{15}$$

B DEEP PROBABILISTIC SAMPLING (DPS) ALGORITHM DESCRIPTION

Algorithm 1 Deep probabilistic subsampling (DPS)

Require: Training dataset $\mathcal{D}$, Number of iterations n_{iter}, τ, $\mathbf{w}_0 = \mathbf{0}$, initialized trainable parameters $\mathbf{\Phi}$ and θ.
Ensure: Trained logits matrix $\mathbf{\Phi}$ and task network parameters θ.
  for $i = 1$ to n_{iter} **do**
    - Draw mini-batches ($\mathbf{x}_i$,$\mathbf{s}_i$): a random subset of $\mathcal{D}$
    - Draw Gumbel noise vectors $\mathbf{e}_m$ for $m \in \{1, \ldots, M\}$
    - Compute $\boldsymbol{A}_{\mathbf{\Phi}} = [\mathbf{a}_1; \ldots; \mathbf{a}_M]$ using:
    if DPS-top1 then
      for $m = 1$ to M **do**
        - ind $= \underset{n}{\text{argmax}}\{w_{m,n} + \phi_{m,n} + e_{m,n}\}$
        - $\mathbf{a}_m = \text{one_hot}_N(\text{ind})$
        - $w_m = w_{m-1}$
        - Update mask: $w_{m,\text{ind}} = -\infty$
      end for
    else if DPS-topK then
      - M-hot $= \underset{n}{\text{TopK}}\{w_{m,n} + \phi_{1,n} + e_{1,n}\}$
      - Expand de M-hot vector in M one-hot vectors $\mathbf{a}_m$
      - $\mathbf{w}_m = -\infty \cdot \mathbf{a}_m$
      - Create the cumulative masks $\mathbf{w}_m = \sum_{i=1}^{m} \mathbf{w}_i, \forall m \in \{1, \ldots, M\}$.
    end if
    - Subsample the signal: $\mathbf{y}_i = \boldsymbol{A}_{\mathbf{\Phi}} \mathbf{x}_i$
    - Compute the task: $\hat{\mathbf{s}}_i = f_\theta(\mathbf{y}_i)$
    - Compute loss using : $\mathcal{L}_i = \|\mathbf{s}_i - \hat{\mathbf{s}}_i\|_2^2 + \mathcal{L}_{\text{pen}}$
    - $\forall$m $\in \{1, \ldots, M\}$ redefine $\nabla_{\boldsymbol{\phi}_m} \mathbf{a}_m \coloneqq \nabla_{\boldsymbol{\phi}_m} \mathbb{E}_{\mathbf{e}_m} \left[\text{softmax}_\tau(\mathbf{w}_m + \boldsymbol{\phi}_m + \mathbf{e}_m)\right]$
    - Use Adam optimizer to update $\mathbf{\Phi}$ and θ
  end for

C HYPERPARAMETERS OF ADOPTED TASK MODELS

This appendix contains tables that provide details on the adopted task model architectures for the different experiments. We use the abbreviation FC, which stands for 'fully-connected layer'.

Table 1: The layer parameters corresponding to the task model in MNIST classification.

Layer	Output size	Init	Activation
FC 1	N	Glorot uniform[2]	LeakyReLU (0.2)
Dropout 1 (20%)	N	-	-
FC 2	256	Glorot uniform	LeakyReLU (0.2)
Dropout 2 (20%)	256	-	-
FC 3	128	Glorot uniform	LeakyReLU (0.2)
Dropout 3 (20%)	128	-	-
FC 4	128	Glorot uniform	LeakyReLU (0.2)
FC 5	10	Glorot uniform	Softmax

Table 2: The layer parameters corresponding to the task model in 'lines and circles' reconstruction.

Layer	Output size	Init	Activation	Kernel size	Strides
FC 1	N	Glorot uniform	tanh	-	-
FC 2	N	Glorot uniform	tanh	-	-
Reshape	$\sqrt{N} \times \sqrt{N}$	-	-	-	-
Conv2D 1	$\sqrt{N} \times \sqrt{N} \times 64$	Glorot uniform	ReLU	5×5	1×1
Conv2D 2	$\sqrt{N} \times \sqrt{N} \times 64$	Glorot uniform	ReLU	5×5	1×1
Conv2D 3	$\sqrt{N} \times \sqrt{N} \times 1$	Glorot uniform	None	7×7	1×1

[2]Glorot & Bengio (2010)

Table 3: The layer parameters corresponding to the task model in CIFAR10 reconstruction.

Network type	Layer	Output size	Init	Activation	Kernel size	Strides
Generator						
	Proximal operator $\mathcal{P}_\theta^{(k)}$					
	$6\times$ Conv2D	$\sqrt{N} \times \sqrt{N} \times 64$	Glorot uniform	ReLU	3×3	1×1
	$1\times$ Conv2D	$\sqrt{N} \times \sqrt{N} \times 1$	Glorot uniform	ReLU	3×3	1×1
	Stepsize $\alpha^{(k)}$					
	$1\times$ Conv2D	$\sqrt{N} \times \sqrt{N} \times 1$	Glorot uniform	ReLU	3×3	1×1
Discriminator D_ψ						
	Conv2D 1	$\frac{\sqrt{N}}{2} \times \frac{\sqrt{N}}{2} \times 128$	Glorot uniform	LeakyReLU (0.2)	3×3	2×2
	Conv2D 2	$\frac{\sqrt{N}}{4} \times \frac{\sqrt{N}}{4} \times 128$	Glorot uniform	LeakyReLU (0.2)	3×3	2×2
	Conv2D 3	$\frac{\sqrt{N}}{8} \times \frac{\sqrt{N}}{8} \times 128$	Glorot uniform	LeakyReLU (0.2)	3×3	2×2
	GlobalAveragePooling2D	128	-	-	-	-
	Dropout 40%	128	-	-	-	-
	FC 1	1	-	Sigmoid	-	-

D EXAMPLE TRAINING GRAPH FOR TRAINING OF 'LINES AND CIRCLES' RECONSTRUCTION USING DPS-TOPK

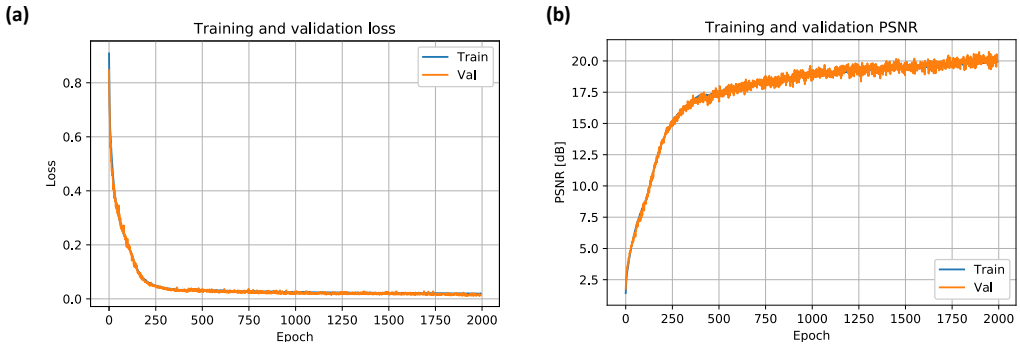


Figure 6: Training graphs for the 'lines and circles' reconstruction problem from section 4.2. (a) The graph shows that even though neural network loss functions are typically non-convex, the optimization trajectory of the loss value still followed a smooth path. The train and validation loss exactly behave the same way, suggesting no overfitting. (b) The corresponding PSNR value during training.