DeepTensor: Low-Rank Tensor Decomposition with Deep Network Priors

Vishwanath Saragadam, Randall Balestriero, *Member, IEEE,* Ashok Veeraraghavan, *Fellow, IEEE,*
and Richard G. Baraniuk, *Fellow, IEEE*

Abstract—DeepTensor is a computationally efficient framework for low-rank decomposition of matrices and tensors using deep generative networks. We decompose a tensor as the product of low-rank tensor factors (e.g., a matrix as the outer product of two vectors), where each low-rank tensor is generated by a deep network (DN) that is trained in a *self-supervised* manner to minimize the mean-square approximation error. Our key observation is that the implicit regularization inherent in DNs enables them to capture nonlinear signal structures (e.g., manifolds) that are out of the reach of classical linear methods like the singular value decomposition (SVD) and principal components analysis (PCA). Furthermore, in contrast to the SVD and PCA, whose performance deteriorates when the tensor's entries deviate from additive white Gaussian noise, we demonstrate that the performance of DeepTensor is robust to a wide range of distributions. We validate that DeepTensor is a robust and computationally efficient drop-in replacement for the SVD, PCA, nonnegative matrix factorization (NMF), and similar decompositions by exploring a range of real-world applications, including hyperspectral image denoising, 3D MRI tomography, and image classification. In particular, DeepTensor offers a 6dB signal-to-noise ratio improvement over standard denoising methods for signal corrupted by Poisson noise and learns to decompose 3D tensors 60 times faster than a single DN equipped with 3D convolutions.

Index Terms—Tensor Decomposition, Matrix Factorization, Low-Rank Completion, Deep Network, Self-Supervised Learning.

1 Introduction

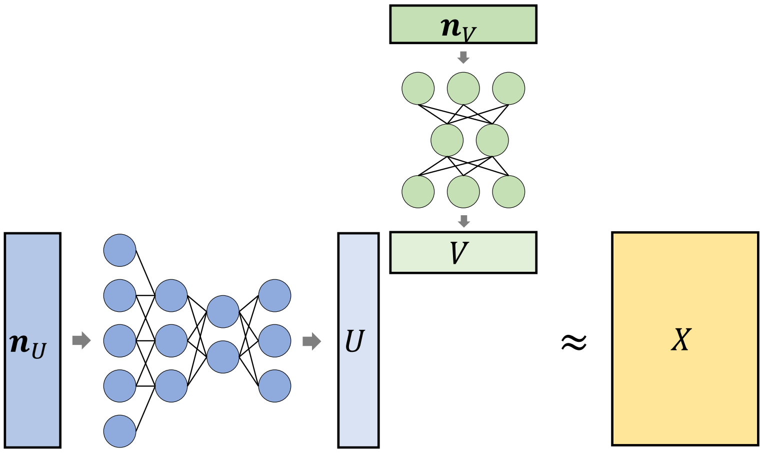
Low-rank representations of matrices and tensors are truly ubiquitous and applied across all fields of science and engineering, from statistics [1–5] to control systems [6, 7] to computer vision [8–10], and beyond. Low-rank representations seek to represent a large matrix/tensor as a product of smaller (and hence lower rank) matrices/fibers. For instance, the classical approach to representing matrices in a low-rank manner is via the *singular value decomposition* (SVD), which expresses a matrix as a product of two smaller orthonormal matrices (containing the singular vectors) and a diagonal matrix (containing the singular values). Thresholding the singular values creates a matrix that inhabits a lower dimensional subspace. The SVD is a pervasive technique for data preprocessing and dimensionality reduction across a wide entire range of machine learning (ML) applications, including *principal component analysis* (PCA) and data whitening.

No matter how powerful or pervasive, however, the SVD and PCA are not without their shortcomings. SVD/PCA is an optimal low-rank decomposition technique only under a narrow set of assumptions on the statistics of the signal and noise in the task at hand [11]. When the signal or noise is non-Gaussian, the resulting decomposition is *not* optimal and results in a subspace that differs from the true low-rank approximation of the underlying matrix. These issues have been addressed somewhat successfully in the past with several signal- and application-specific regularizers that include sparsity on error [12–14], total variation penalty [15–17], and data-driven approaches [18, 19]. The key observation is that a good signal model can act as a strong regularizer for estimating the low-rank factors. Unfortunately, finding a useful signal model/regularizer for a new application can be a daunting task. Does there exist a generalized regularizer that can encompass a large class of signals and applications? We found the answer to be hidden implicitly in deep networks (DNs).

*In this paper, we propose **DeepTensor**, a new approach for low-rank matrix/tensor decomposition that is robust to a wide class of signal and noise models.* Our core enabling observation is that DNs produce signals that are implicitly regularized due to the networks' inherent inductive bias. We exploit DNs as priors by representing a matrix/tensor in terms of factors output from a set of *untrained* generative networks (see Fig. 1). The parameters of the networks are learned in a self-supervised manner for each matrix/tensor using a simple MSE loss between the matrix/tensor and the product of the deeply generated factors. The inductive bias of the generative networks enables DeepTensor to better identify the underlying low-dimensional subspace while rejecting noise, resulting in a more accurate estimate of the noise-free matrix/tensor.

DeepTensor is a computationally efficient, drop-in replacement for many existing matrix/tensor factorization approaches that combines the simplicity of low-rank decomposition with the power of deep generative networks. Further, DeepTensor can significantly improve the performance of downstream tasks, such as image classification, that rely on low dimensional representation. We empirically back these claims via experiments on a wide variety of real-world tasks, including denoising with low-rank approximation, "eigenfaces" [20] for facial recognition (see Fig. 2), solving linear inverse problems with

- *V. Saragadam, R. Balestriero, A. Veeraraghavan, and R. Baraniuk are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX, 77005.*
E-mail: vishwanath.saragadam@rice.edu



(b) Low-rank representation via DeepTensor

Fig. 1: **DeepTensor** is a new a low-rank decomposition technique that exploits the implicit regularization capabilities of DNs. Conventional low-rank factorization such as (a) SVD relies on linear factors to represent the matrix. DeepTensor represents via factor matrices that are outputs of DNs. Factorization is then achieved by learning parameters of the network in a self-supervised manner that reduces the (ℓ_2) loss between the factor and input matrix.

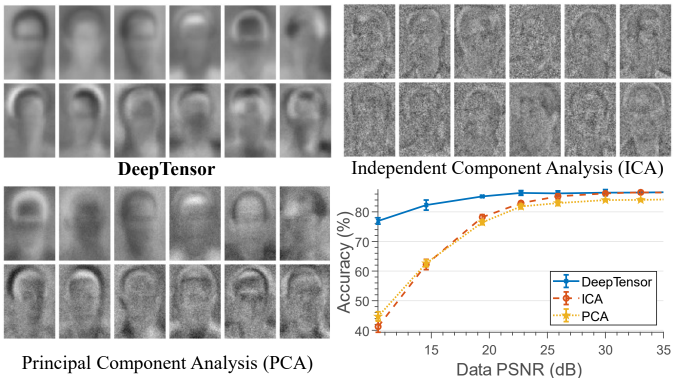


Fig. 2: **DeepTensor eigenfaces.** DeepTensor is a drop-in replacement for most dimensionality reduction techniques and is robust to a wide range of signal and noise models. Here we show an "eigenface" decomposition with facial data corrupted by Poisson noise. The top twelve bases vectors visualized above at 15 dB input PSNR show that DeepTensor learns principal components that are significantly lower in noise levels compared to ICA or PCA. This in turn affects the classification accuracy (bottom right) that involves taking projection of input data on the principal components followed by a linear SVM classifier.

compressively sensed videos, tensor denoising, and recovery of 3D volumes from computed tomographic (CT) measurements. We also highlight DeepTensor's computational efficiency and scalability for large and higher-order tensor decomposition by demonstrating that it offers a 60× or more speedup for decomposing 3D tensors as compared to a single generative network equipped with full 3D convolutions.

2 Background and Prior Work

DeepTensor leverages classical work on low-rank approximation and recent self-supervised learning techniques with DNs [21, 22]. We provide a brief overview of these topics to introduce our notation and intuition and to set context for our work.

Low-rank approximation. Low-rank approximation seeks to represent a matrix $X \in \mathbb{R}^{M\times N}$ of rank $R = \min(M, N)$ as a product of two smaller matrices, $U \in \mathbb{R}^{M\times k}, V \in \mathbb{R}^{N\times k}$ where k is generally taken to be smaller than R. The specific constraints on U, V, and the desired objective give rise to different types of low-rank approximation algorithms. For example, one recovers PCA [23], nonnegative matrix factorization (NMF) [24] and k-means [25] via

$$\min_{U,V} \|X - UV^T\|_F \quad \text{s.t.} \quad U = V^T \qquad \text{(PCA/SVD)} \tag{1}$$

$$\min_{U,V} \|X - UV^T\|_F \quad \text{s.t.} \quad U \geq 0, V \geq 0 \qquad \text{(NMF)} \tag{2}$$

$$\min_{U,V} \|X - UV^T\|_F \quad \text{s.t.} \quad [V]_{.,k} \in \{e_1, \ldots, e_n\} \qquad (k\text{-means}) \tag{3}$$

where e_k is the k^{th} Euclidean canonical basis vector, and $[V]_{.,k}$ is the k^{th} column of V. Applications of low-rank approximation are extremely diverse ranging from denoising [26, 27], compression [28], clustering for anomaly detection [29], and forecasting [30].

The Achilles' heel of low-rank approximation is non-Gaussian signal and/or noise statistics. There are been many extensions and variants of the SVD algorithms, such as Robust PCA [12] that improves robustness in learning the low-rank matrices against outliers in the data. Similarly, other non-Gaussian noise settings, different metrics and constraints could be employed to obtain the most adapted U, V decomposition to solve the task at hand. This can be well understood based on the generative models corresponding to low-rank approximation techniques such as Probabilistic PCA [31] from which it is clear that PCA is optimal under a Gaussian noise model, and in the presence of say a Laplacian noise, an ℓ_1 reconstruction loss should be used instead. Prior work identified approaches to tackling tensor decomposition with unknown noise statistics via Bayesian optimization [32, 33], modeling noise distribution as a mixture of Gaussian [34], and using a decomposition inspired by Kronecker product [35, 36]. While the approaches are promising for several tensor decompo: none of the previous works leverage indu

There has been some research in e DNs for matrix factorization especially nonnegative matrix factorization [37], m: imaging (MRI) denoising [19], and tenso The key idea is that the statistics of train regularize the inverse problem of matrix f techniques however suffer from dataset b very large pools of data to be effective.

Deep networks as implicit regulari emerged very rapidly from classificatio applications where they have reached su mance across a wide range of datasets an recently, the use of DNs has diversified and important example for this paper is Prior (DIP) model [21]. In this setting, a DN f is used as a constrained projection of a random noise vector z to fit a target sample x as follows

$$\min_{\Theta} \|f_{\Theta}(z) - x\|_2^2. \tag{4}$$

When the architecture of the DN is carefully picked, the estimation of the input x is denoised, and well reconstructed even in the presence of missing values. From the implicit regularization field it is understood that the above problem is equivalent to some problem

$$\min_{W} \|Wz - x\|_2^2 + R(W), \tag{5}$$

where $R(W)$ is a regularization term that directly depends on the choice of the DN architecture [39]. The key result that we will leverage throughout this paper is that *searching for the DN parameters producing the desired result in (4) is equivalent to searching in the space of regularizers in (5).*

Related work combining DNs and low-rank decomposition. Applying DNs in a self-supervised manner for matrix decomposition has received surprisingly little attention. The closest work is by Aittala et al. [40], which regularized a matrix factorization for a specific video imaging problem using generative networks. The video sequence was represented by a generative network equipped with 3D convolutional kernels, while the light transport matrix was represented as a linear combination of the input video sequence multiplied by another generative network equipped by 2D convolutional kernels. DeepTensor is in many ways inspired by the factorization idea proposed by Aittala et al. [40] but goes beyond light transport matrices and can be applied to a wide variety of problem settings.

A related but different approach to DeepTensor is the work by Bacca et al. [41], who identified that hyperspectral images (which are modeled as 3D tensors) can be represented as the output of a single generative network equipped with 3D convolutional kernels. The work by Bacca et al. [41] is a promising framework for solving inverse problems in hyperspectral imaging, but are not aimed at low-rank matrix factorization – which is the focus of this paper. The key difference between their work and DeepTensor is that the input to their 3D network is a low-rank Tucker tensor; in contrast we *output* a low-rank Tucker tensor.

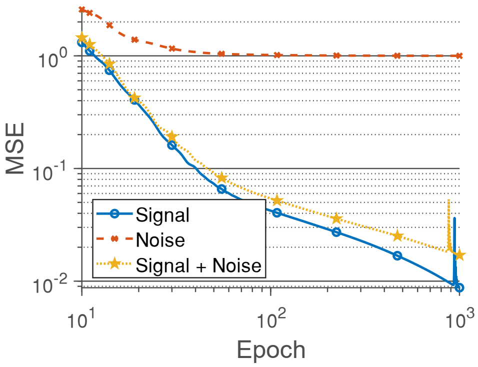


Fig. 3: **Implicit bias of DeepTensor rejects noise.** The inductive bias due to the network architecture fits better to signal and rejects noise. To test this, we performed a rank-20 decomposition of a hyperspectral image [42], iid Gaussian noise with a standard deviation of 0.01 units, and a noisy signal obtained by adding the hyperspectral image and the Gaussian noise. We observe that through the training epochs, the signal is better fit by the network, while the noise has a slower fit.

3 DEEPTENSOR DECOMPOSITION

3.1 Low-rank decomposition with deep network prior

Matrix decomposition. Consider the low-rank decomposition in (1). If we include regularizers for U and V, we obtain the following optimization function

$$\min_{U,v} \|X - UV^{\top}\|^2 + R(U,V), \tag{6}$$

where $R(U,V)$ is a regularizer for U,V. Instead of having explicit regularizers on the left and right matrices, we model U,V as the outputs of generative networks f_U, f_V, which yields

$$\min_{\theta_U,\theta_V} \|X - f_U(z_U)f_V(z_V)^{\top}\|^2, \tag{7}$$

where z_U and z_V are randomly initialized inputs to the networks f_U, f_V respectively that have parameters θ_U, θ_V. The networks'output are of the same shape as the desired U and V matrices from (6). We note here that there is no further regularizer on the matrices – any regularization comes from the implicit prior of the DN itself, which makes it an appealing choice to solve a diverse type of signals and noise settings. Figure 3 visualizes the regularization offered by DNs for the task of rank-20 decomposition. Over 100 iterations, the error for signal reduces by two orders of magnitude, while the error reduces by less than one order for noise. This slow fitting to noise is a result of implicit bias of DNs which is leveraged by DeepTensor.

Tensor decomposition. The task of tensor decomposition finds numerous applications and is an active area of research, where the major difficulty rises from defining an appropriate regularizer/basis constraint (recall (3)). Any constraint on the factor matrices can be expressed as regularization functions. Hence, given the following general decomposition problem

$$\min_{U,V,\ldots,W} \|X - \underbrace{U \otimes V \otimes \cdots \otimes W}_{k \text{ times}}\|^2 + R(U,V,\ldots,W),$$

with X a k-dimensional tensor, one needs to specify the correct regularizer (R) such that the produced decomposition is adapted for the task at hand. This search is mostly

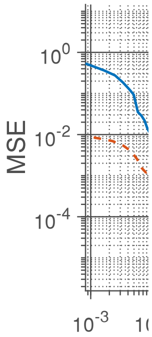


Fig. 4: **Separable conv**
benefit of deep low ran
are separated. Hence f
and one 1D network fo
resulting in two orders

understood in narrov
noise and Gaussian la
seeks to solve the the
decomposition

$$\min_{\theta_U,\theta_V,\dots,\theta_W} \|X - f_U$$

where $f_U, f_V, \dots, f_W$
tors $z_U, z_V, \times, z_W$ wi
Note that this decom
factor analysis (PAR
Tucker representation

Dimensionality s
sacrificing performa
the matrix itself via a
network instead of tw
is to study the expone
by separating the dim
tensor products of mu
(8) versus doing the t
then using as done f
that naturally general

$$f_{U,V,\dots,W}(z_U \otimes z_V \otimes \cdots \otimes z_W). \tag{9}$$

To understand the benefits of those two different cases we approximate a 3D magnetic resonance imaging (MRI) volume of size 128 × 128 × 150 with three types of networks — one 2D and 1D that is a natural parametrization that captures the dependence of the first two dimensions $f_{U,V}(z_{U,V}) \otimes f_W(z_W)$, three 1D (8), and one 3D networks (9). Figure 4 shows the plot of mean squared error (MSE) as a function of time for the three approaches. We note that using three 1D networks is faster than using a 2D and 1D network, and both are two orders of magnitude faster than using one 3D network. Additionally, our (separate) parametrization has the benefit of keeping interpretability since one has access to each generated low-rank matrix that combine to form the observed data matrix X. The choice between using networks equipped with 2D and 1D convolutions, and networks equipped with 1D convolutions is specific to the task at hand. Tensors such as videos and hyperspectral images benefit from networks equipped with 2D and 1D convolutions. In contrast, tensors from multi-dimensional face databases [43] or computed tomographic (CT) images benefit from a full tensor decomposition.

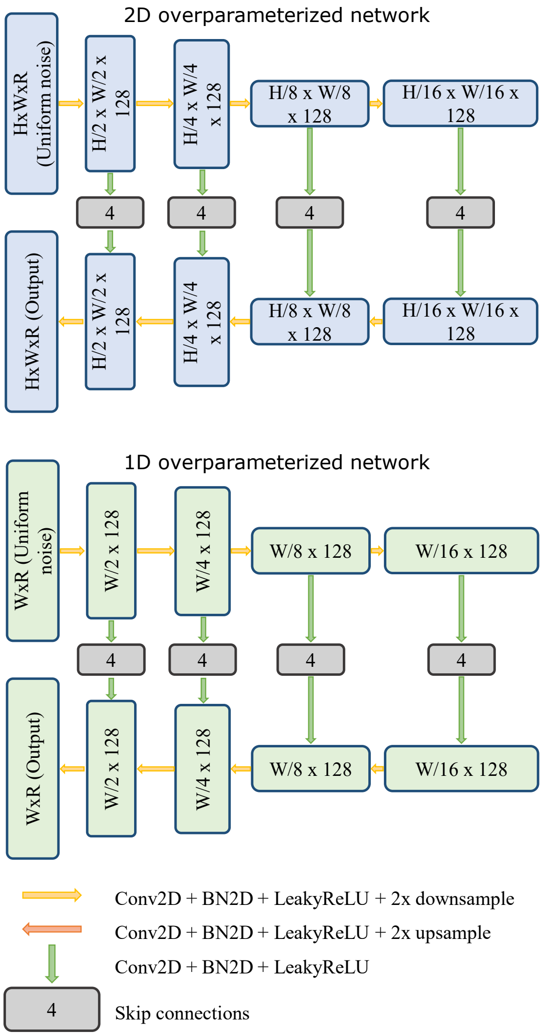


Fig. 5: **Architectures for various experiments in the paper.** We used modified 2 dimensional (top) and 1 dimensional (bottom) versions of the networks proposed in Deep Image Prior [21].

Unless explicitly specificed, we utilize overparameterized networks for the factor matrices, similar to DIP [21] (see Fig. 5). However, it is possible to use underparameterized networks instead, similar to the deep decoder architecture [22]. The advantage of the latter is that the learned parameters can be used as the compressed version of the tensor being decomposed. In contrast, for pure data imputation and denoising the DIP version should be preferred. We compare the two choices in the upcoming section.

Optimization procedure. We optimized for the parameters of the networks that output the factor matrices using stochastic gradient descent. Specifically, we employed the ADAM optimizer [44] and implemented the optimization using PyTorch framework [45]. For all experiments, we optimized an ℓ_2 loss function between the data and the product of outputs of DNs. The primary purpose of choosing a simple loss function was to emphasize on the regularization capabilities of the DNs. In practice, it is possible to choose a

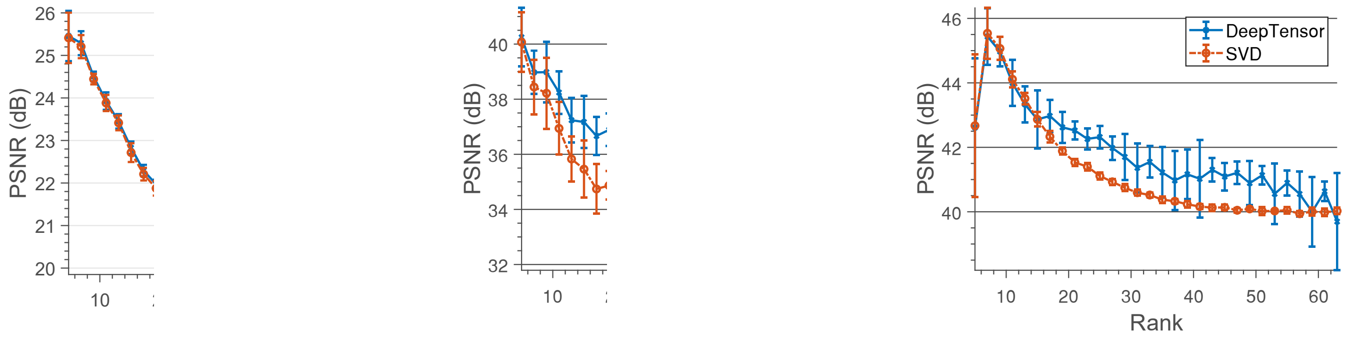


Fig. 6: **DeepTensor is robust to a range of signal and noise distributions**. We generated low-rank matrices with various signal and noise distributions and then performed a low-rank de

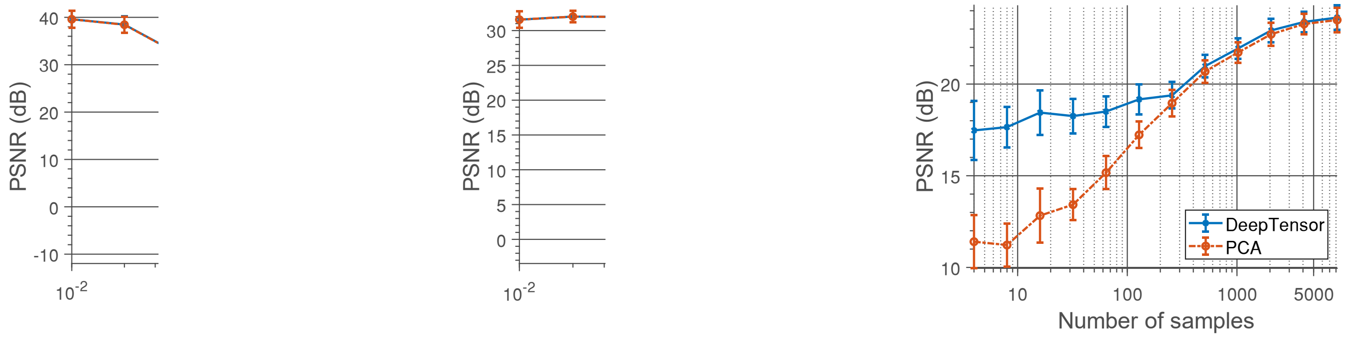


Fig. 7: **DeepTensor is a more robust for PCA under low SNR or small sample size.** The SVD is the core computation of PCA. We generated data with varying number of samples drawn from a Gaussian distribution with varying standard deviation. We then computed PCA components via SVD and DeepTensor decomposition. We observe that DeepTensor is particularly advantageous when the data is corrupted by a large noise, or the number of samples are limited.

more appropriate loss function for each specific problem.

3.2 Validation experiments

Low-rank completion. We generated random 64×64 dimensional matrices with rank varying from 10 to 60. The left and right matrices were generated as either iid Gaussian values (with a standard deviation of 1 unit), or random, piecewise constant signals, emulating visual signals. We then added one of the three types of noise:

- iid Gaussian noise with standard deviation of 0.1 units: $Y = X + \mathcal{N}(0, 0.1)$
- Poisson noise with mean at each entry of the matrix lying in the range $[0, 1000]$: $Y = \mathcal{P}(1000X)$, which is common in visual signals
- Rician noise with standard deviation of 0.02 units: $Y = \sqrt{(X + \mathcal{N}(0, 0.02))^2 + \mathcal{N}(0, 0.02)^2}$ that is common in MRI measurements.

Figure 6 shows a plot of peak signal-to-noise ratio (PSNR) as a function of rank for various signal and noise types averaged over 10 realizations. We make two observations. First, when the noise is Gaussian, DeepTensor has similar performance to SVD/PCA. This is expected, since SVD/PCA is known to be the optimal low-rank decomposition for white Gaussian matrices [11]. Second, for other noise settings, such as Poisson or Rician, DeepTensor has a far superior performance across all rank values. This empirically establishes that DeepTensor is well suited for a range of non-Gaussian noise models.

Principal Component Analysis. The performance of SVD-based PCA degrades under noisy conditions or when samples are limited. To verify how DeepTensor can benefits PCA, we generated variable number of data points with a known intrinsic matrix generated as iid Gaussian random variables with mean 0 and standard deviation of 1. The feature dimension was 64 and the intrinsic dimension (rank) was 10. We generated data via a linear combination of the columns where the weights were drawn from an iid Gaussian distribution with mean 0 and standard devation. We then added iid Gaussian noise to the data matrix with varying levels of standard deviation. Figure 7 plots the accuracy of estimating the PCA components for varying noise levels and number of samples averaged over 10 realizations. We observe that with low noise or a large number of samples, SVD-based PCA and DeepTensor have similar performance. However, with high noise or a limited number of samples, DeepTensor significantly outperforms SVD-based PCA.

TABLE 1: **Effect of learning rate schedule.** The table presents the best achievable learning rate for low-rank approximation of a toy matrix. We repeated each experiment five times. SVD accuracy was 14.4 dB. The choice affects the final achievable accuracy – fixed scheduling with a high learning rate performs better than other choices.

Scheduler Max learning rate	Fixed	Step	Exponential	Cosine annealing
10^{-2}	17.6 ± 0.7	17.1 ± 0.4	16.6 ± 0.4	16.5 ± 0.3
10^{-3}	17.5 ± 0.6	17.1 ± 0.5	16.6 ± 0.4	16.6 ± 0.3
10^{-4}	17.6 ± 0.6	17.1 ± 0.4	16.6 ± 0.4	16.5 ± 0.3
10^{-5}				

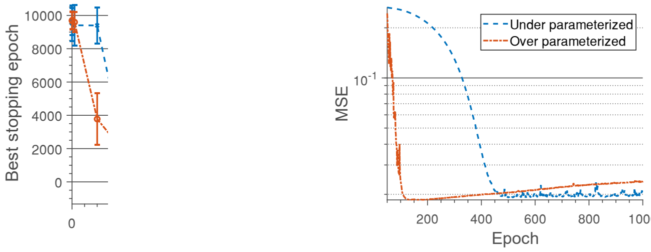
3.3 Sensitivity to learning parameter tures

We now discuss how the choice of learn ing and the network architecture affect tl obtained by DeepTensor.

Learning rate scheduling. The choice and its scheduling directly affects the max accuracy. To gain empirical insight into ho scheduling affect the training process, we performed a rank-16 low-rank decomposition of a 64×64 matrix. The entries of the matrix were drawn from iid Gaussian distribution. We then varied the learning rates from 10^{-5} to 10^{-2}, and chose four learning rate schedulers, namely fixed with no change in learning rate, step scheduler where the learning rate was multiplied by 0.99 after every 2,000 epochs, exponential scheduler with a multiplication factor of 0.9999, and cosine annealing-based scheduling [46].

Table 1 shows results for varying learning rate and its schedule. We note that a fixed scheduler results in higher PSNR across all learning rates. Moreover, learning rates ranging from 10^{-4} to 10^{-2} are equivalent choices – a very low learning rate of 10^{-5} resulted in poorer PSNR. The two observations above imply that DeepTensor does not require complex learning rate scheduling and is robust to the learning rates.

Stopping criterion. The stopping criterion for optimal approximation accuracy is a function of input noise distribution and network architecture. To demonstrate this dependence, we performed a rank-16 low-rank decomposition of 64×64 matrices with entries drawn from iid Gaussian distribution. We then utilized an under-parameterized, and over-parameterized network to estimate the left and right factor matrices with varying levels of noise. Figure 8 compares the results with the two types of networks. We note in Fig. 8(a) that both networks require fewer iterations with increasing noise level for least approximation error. Fig. 8(b) shows approximation error over epochs. Over-parameterized networks achieve lower approximation error than under-parameterized networks. However,the error for over-parameterized networks increases rapidly after optimal stopping epoch (100), whereas the error increases more gently for under-parameterized networks after the optimal stopping epoch (500). Ultimately, the stopping criteria and the network architecture depend on the exact application and prior knowledge about noise levels in the signal.



(a) Best functior

(b) Training epochs for over and under-parameterized networks.

Fig. 8: **Stopping criterion depends on input noise level and network architecture.** The optimal stopping epoch (a) reduces with increasing noise standard deviation. (b) shows the approximation error when the noise standard deviation is 0.2. The exact stopping epoch is less important for under-parameterized networks [22], whereas over-parameterized networks [21] are more sensitive to the stopping epoch, but achieve lower approximation error.

3.4 Modeling Known Decomposition Constraints with the Last Layer Activation

Different constraints on U and/or V lead to different known low-rank decompositions such as nonnegativity on U and V leads to NMF and nonnegativity on U alone leads to semi-NMF [47], which is an ubiquitous form of clustering [48]. In our case, U and V are outputs of a DN (recall Sec. 8) and are thus without range constraints. Hence we need to impose nonnegativity constraints on the output.

Nonnegativity can be achieved with various activation functions including ReLU, softplus, and element-wise absolute value, and the exact choice affects the achievable accuracy. To gain an empirical insight into the the effect of the activation function, we considered NMF on MNIST [49] and CIFAR [50] datasets. In both cases, we used 2048 images for training. We added a rician noise to the images of the form $\mathbf{n} = 0.3(0.3\mathbf{z}_1 + \mathbf{z}_2^2)$ where $\mathbf{z}_1$ and $\mathbf{z}_2$ are iid Gaussian random variables with zero mean and unit variance. The input PSNR was evaluated to be 4.8 dB. We then performed NMF on the resultant images with various techniques. Comparisons are tabulated in Table 2. We employed a combination of ℓ_2 loss for data fidelity, and ℓ_1 penalty for the factor matrices output from the DNs. DeepTensor, particularly when combined with the ReLU activation function achieves higher approximation accuracy compared to baseline NMF algorithm [51].

TABLE 2: **DeepTensor is a robust alternative to NMF.** We performed NMF on MNIST and CIFAR10 datasets by enforcing nonnegativity on U, V through different activation functions. The table below shows the average PSNR over 10 runs. DeepTensor with ReLU as the final activation function outperformed standard NMF [51], underscoring the efficacy for matrix factorization with positivity constraints.

	MNIST		
Act. func.	softplus	abs. value	ReLU
DeepTensor	7.4 ±0.03	7.3±0.01	$\mathbf{7.6 \pm 0.15}$
NMF [51]	7.5		
	CIFAR10		
Act. func.	softplus	abs. value	ReLU
DeepTensor	8.4 ±0.08	8.3±0.06	$\mathbf{8.8 \pm 0.17}$
NMF [51]	8.2		

4 Applications

We now showcase the breadth of DeepTensor's applicability in several real-world applications.

Training details. Unless otherwise specified, we used overparameterized networks with skip connections (similar to DIP [21]), and trained with a learning rate of 10^{-3}. We implemented our training process in Python using the Pytorch framework [45]. All our experiments were run on a Linux machine equipped with a hexa-core Intel Core i7-6850K CPU, 128GB RAM, and three NVIDIA GeForce GTX 1080 Ti. The network architectures were a modified version of the one used by Ulyanov et al. [21]. Specifically, we changed the number of output channels to be equal to the rank for all factor matrices. For tensor decomposition with 1D fibers, we changed the 2D convolutional architecture in [21] to a 1D architecture. We optimized for both inputs and network parameters, which provided faster convergence; however, there was no other significant difference if we did not optimize for the inputs.

4.1 Linear inverse problems in computer vision

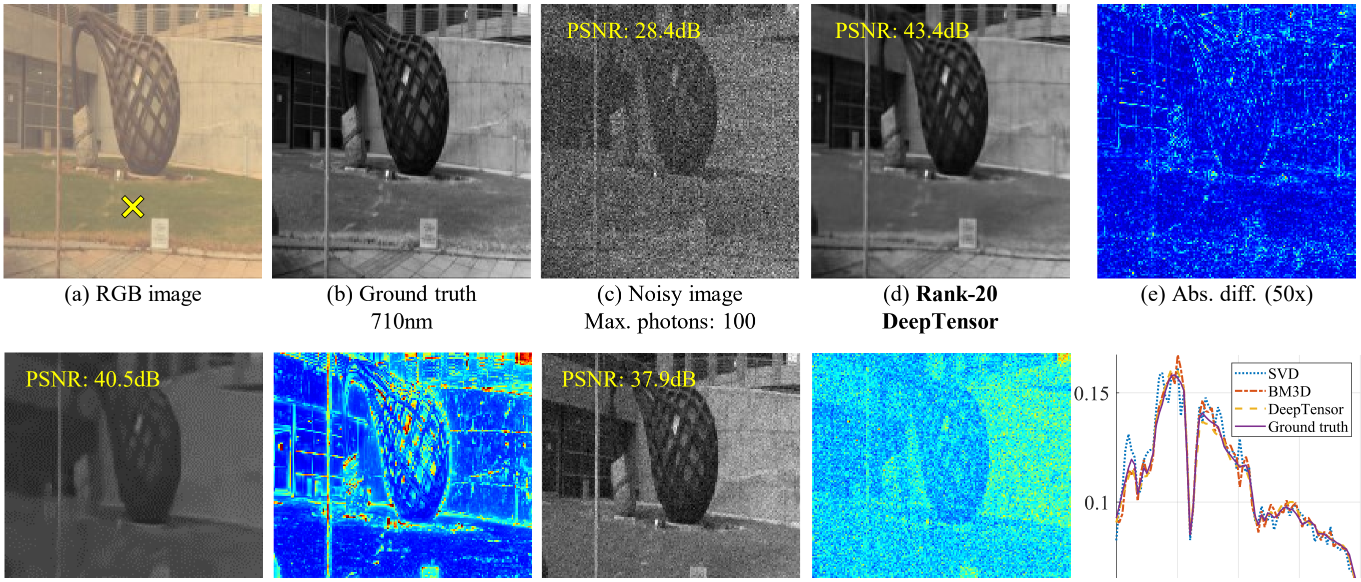
Low-rank model finds use in numerous applications in computer vision including sensing of light transport matrices [52], hyperspectral imaging [53], video compressive sensing [54], magnetic resonance imaging (MRI), and positron emission tomography (PET). Most inverse problems involve collection of limited data samples and/or highly noisy samples. In both cases, we expect the DeepTensor to be very effective. We showcase three specific examples here.

Hyperspectral image denoising. Low-rank models offer a concise representation of hyperspectral images (HSI) and are used in compression, sensing [53], and dimensionality reduction. HSIs involve imaging the scene across several hundreds of spectral bands, resulting in high photon noise (highly non-Gaussian). Typically, a HSI of dimension $N_x \times N_y \times N_\lambda$ is converted to a matrix of dimension $N_xN_y \times N_\lambda$ which is then approximated using a low-rank model. We denoised a $348 \times 327 \times 260$ HSI from Arad and Ben-Shahar [42] by simulating Poisson noise equivalent to a maximum of 100 photons per spatio-spectral voxel, and a readout noise of 2 photons – settings corresponding to a dull overcast outdoor scene. We then performed a rank-20 decomposition with SVD and DeepTensor. We also compared DeepTensor against the BM3D denoising algorithm [55]. We ran the DeepTensor optimization process for a total of 5000 iterations. Figure 6 shows the results with both techniques. DeepTensor outperforms SVD by 6dB, and BM3D by 3dB and produces visually pleasing results.

Tensor denoising. Low dimensional tensor decomposition is a promising approach to tensor denoising, especially with gross outliers such as salt and pepper noise. We performed a 3-way PARAFAC decomposition of a subset of faces from the Yale face database (B) [56] consisting of 160, 192×168-dimensional images. We then added 30%, and 60% salt and pepper noise for a fair comparison against the Kronecker Decomposition-based approach (KDRSDL) which is the state-of-the-art in tensor decomposition [35]. We found that an ℓ_1 penalty worked better than ℓ_2 for DeepTensor, as the noise was spatially sparse. In both cases, we picked a rank of 2000 for PARAFAC decomposition. Results are shown in Fig. 10. Evidently, DeepTensor outperforms KDRSDL, especially in extremely noisy settings.

Video compressive sensing. DeepTensor can also be used for full-rank decomposition, which is apt for signals such as videos. To test this, we used our framework on recovery of video frames from spatially multiplexed images. We relied on the setup from Hitomi et al. [54] where each pixel was sampled at an arbitrary time frame across multiple frames. We combined 8 frames into one single coded image and then used DeepTensor to solve the linear inverse problem. For comparison, we recovered video by solving the linear inverse problem with a 2D TV over spatial images as a regularizer. We trained DeepTensor for a total of 10,000 epochs. Figure 11 shows coded image as well as the 8 recovered images for an example video. DeepTensor not only has higher accuracy than TV, but the recovered images look visibly similar to ground truth.

3D reconstruction from CT images. DeepTensor is well suited for tasks such as recovery of 3D volume from CT scans. To demonstrate the advantages of using a tensor representation, we rely on PARAFAC decomposition. We approximated a $256 \times 256 \times 56$ PET CT scan from Clark et al. [57] with a rank-1000 PARAFAC tensor. Note that although the rank is much larger than any single dimension, the effective number of parameters are only 15% of the number of elements in the 3D volume. We added Poisson and readout noise of maximum of 100 photons per voxel and 2 photons respectively. We then simulated 40 tomographic measurements for each slice. 2D TV results were obtained by a TV penalty on each slice along z-direction. Bacca et al. [41] results were obtained by representing input as a rank-1000 PARAFAC tensor and using an untrained 2D network which output 56 channels. 2D TV + PARAFAC results were obtained with self-supervised learning by representing the volume via rank-1000 PARAFAC decomposition, and a TV penalty on each slice along z-axis. Algorithmic reconstruction results were obtained by solving the linear inverse problem without any other priors. FBPConvNet [58] results were obtained by using the output of algorithmic reconstruction and then denoising with a pre-trained model. Finally, the results by [59] were obtained with a self-supervised learning by using a 3D convolutional neural network. While [59] utilized an MRI



(a) RGB image

(b) Ground truth 710nm

(c) Noisy image Max. photons: 100

(d) **Rank-20 DeepTensor**

(e) Abs. diff. (50x)

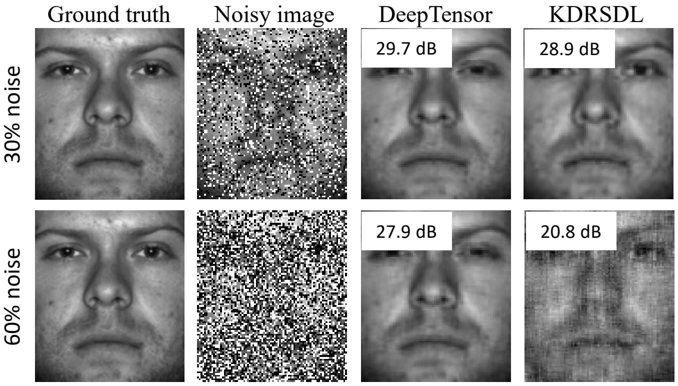


Fig. 10: **Tensor denoising.** DeepTensor enables denoising via low-rank tensor representation that is often better than state-of-the-art denoising techniques [35].

image as input for the network, we used uniform random noise. Results are visualized in Fig. 12. DeepTensor achieves better reconstruction accuracy than other approaches in both PSNR and SSIM. We note that pre-trained approaches such as FBPConvNet Jin et al. [58] can outperform DeepTensor if trained with the appropriate data.

4.2 Classification via low-dimensional projection

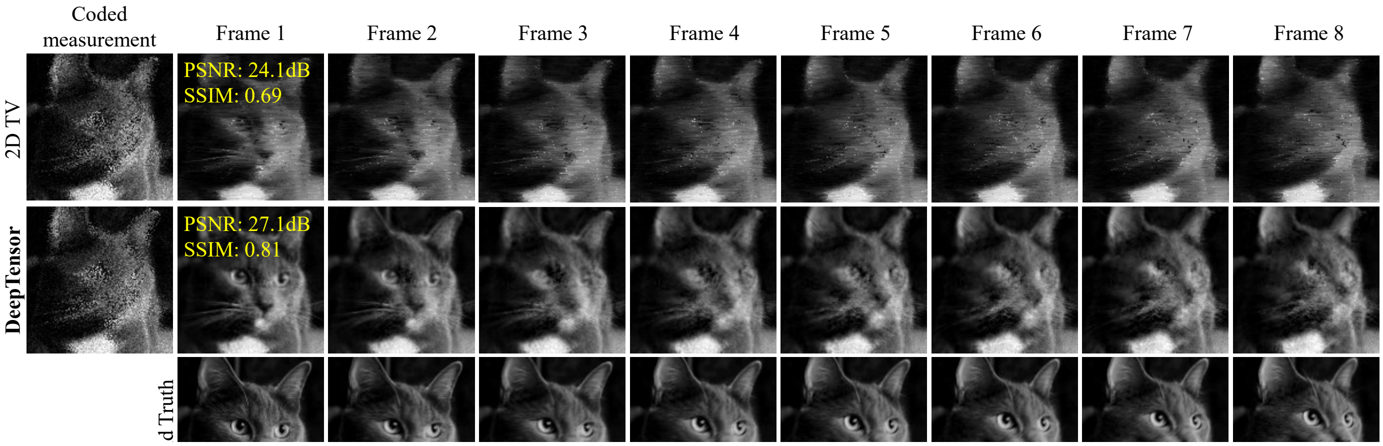
A robust low-rank approximation also affects downstream tasks such as classification. To test this, we worked with an Eigenfaces [20] example for facial classification. We took 840 images across 28 subjects from the Weizmann dataset [60]. We used 25% of the data for training an 84-dimensional subspace via PCA, Independent Component Analysis (ICA) [61], and running DeepTensor on the sample covariance matrix. To emulate noisy conditions, we added poisson noise with variable amounts of noise. We then converted the images to the 84-dimensional space and trained linear and kernel support vector machine with radial basis function. We cross-validated to choose penalty that maximized classification accuracy for each individual classifier. Finally, we evaluated the resultant linear subspace along with SVM on the test data to compute average accuracy. Figure 2 visualizes the learned basis vectors with the three approaches, and the average classification accuracy as a function of input data PSNR. We notice that the basis learned from DeepTensor is smoother and better representative of the underlying data. In contrast, PCA and ICA overfit to the noise in data, resulting in a reduction in classification accuracy.

4.3 Time-frequency decomposition

As discussed in section 3.4, DeepTensor can be combined with additional constraints such as nonnegativity on U, V. We validated the advantages of DeepTensor by performing an NMF on speech data obtained from the GOOGLECOMMANDS dataset [62]. We first computed spectrograms on all speech data with a window size of 1024 samples and a hop size of 32, resulting in a 512×512 dimensional time-frequency image. We then added noise as described in section 3.4, resulting in an input SNR of 4.6 dB. The approximation accuracies with DeepTensor, and a baseline NMF [47] NMF algorithm in Table 3. DeepTensor performs significantly better than the baseline. This ability of DeepTensor to perform better in high noise settings is of particular significance when speech is recorded in noisy environments.

4.4 Choice of rank

As with all matrix and tensor decomposition approaches, the rank of the decomposition is an important parameter for



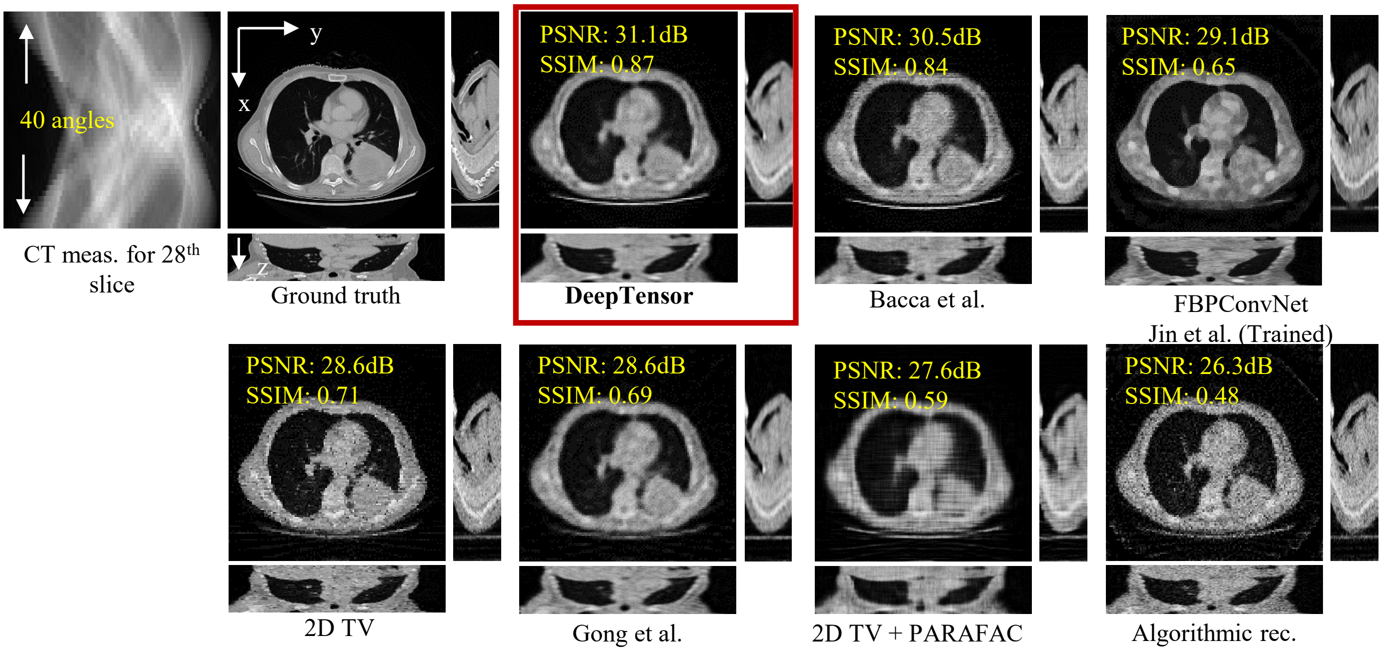


Fig. 12: **Reconstruction from 3D CT measurements.** We utilized DeepTensor to recover 3D PET volume from limited, noisy CT images. We expressed the 3D volume as a rank-1000 PARAFAC tensor and then computed CT images per each slice along 40 angles. DeepTensor resulted in higher SSIM compared to other approaches.

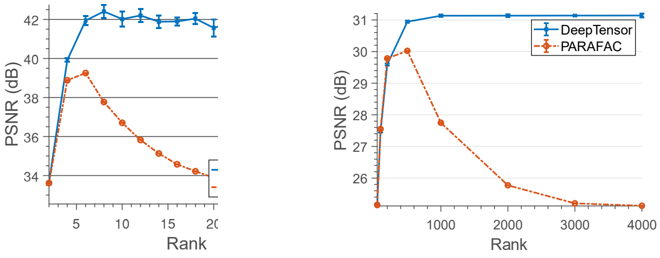
DeepTensor. However, we now demonstrate that DeepTensor is less sensitive to the choice of rank than the SVD in two different settings. In the first experiment, we truncated a hyperspectral image [63] to rank-20 and swept the rank from 1 to 30. In the second experiment, we truncated a subset of the Yale-B dataset [56] using rank-1000 PARAFAC decomposition and then swept the decomposition rank from 10 to 4000. In both cases, we added Poisson ($\lambda_{\max} = 100$) and Gaussian noise ($\sigma = 2$) equivalent to a 25 dB measurement PSNR. Figure 13 shows plots of PSNR as a function of rank for both the cases. When the rank is under-estimated, DeepTensor and SVD achievely similarly low accuracy. However, when the rank is *over-estimated*, the achievable accuracy with SVD reduces, while the accuracy with DeepTensor stays approximately the same. The exact choice of rank is less important for DeepTensor compared to the SVD, which is significantly advantageous when the appropriate rank to use is unknown.

5 Conclusions

Discussion. We have demonstrated that self-supervised learning is effective for solving low-rank tensor and matrix decomposition. Across the board, we see that DeepTensor is a superior option compared to SVD/PCA when the input SNR is low, the matrix/tensor values are non-Gaussian distributed, or the inverse problem is ill-conditioned such as in the case of PCA with limited samples, or linear inverse

TABLE 3: **NMF on speech data.** Average and standard deviation (over 10 runs) of the PSNR (dB) for GOOGLECOMMANDS [62] in the NMF (nonnegativity of U, V) and semi-NMF (nonnegativity of U alone) low-rank decomposition setting for different activation functions enforcing nonnegativity. DeepTensor with ReLU performs better than standard NMF [51].

	GOOGLECOMMANDS NMF			GOOGLECOMMANDS SEMI-NMF		
Act. func.	softplus	abs. value	ReLU	softplus	abs. value	ReLU
DeepTensor	9.0 ±0.03	8.7±0.01	8.7±0.03	8.7±0.09	8.8±0.02	8.8±0.02
		.6			4.7	



(a) Matrix decomp

(b) PARAFAC tensor decomposition

Fig. 13: **Effect of choice of rank.** We truncated a hyperspectral image to rank-20 and a tensor to rank-1000 (PARAFAC) and added shot noise. We then swept rank of decomposition. We observe that DeepTensor is less sensitive to rank compared to SVD and PARAFAC.

problems. Moreover, our separable approximation approach results in faster approximation of 3D tensors compared to DNs with 3D convolutional filters.

Future directions. Our experiments relied mostly on 1D or 2D convolutional filters, whose inductive biases are better suited for signals such as time series and images. However, the DNs can be chosen specific to the task at hand, such as fully connected or recurrent networks. DeepTensor can also be used in non-linear representations, such as local low-rank, or even non-local low-rank. Such settings increase the range of problems that can be tackled effectively and are exciting future directions.

Limitations. The current bottleneck of DeepTensor, as opposed to techniques such as SVD, lies in the training computational complexity. While a single pass through a DN is much faster than the training of SVD or NMF, the need to repeatedly iterate between the forward passes and back propagation inherent to gradient based learning slows DeepTensor training. This opens up interesting research directions aimed at discovering simpler networks that are faster to train yet maintain high performance and developing specialized training algorithms that leverage the low-rank decomposition structure.

6 Acknowledgments

This work was supported by NSF grants CCF-1911094, IIS-1838177, and IIS-1730574; ONR grants N00014-18-12571, N00014-20-1-2534, and MURI N00014-20-1-2787; AFOSR grant FA9550-22-1-0060; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047

References

[1] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966. 1

[2] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[3] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *ACM Symposium on Theory of Computing*, 2013, pp. 665–674.

[4] E. J. Candes and B. Recht, "Exact low-rank matrix completion via convex optimization," in *Allerton Conf. Communication, Control, and Computing*, 2008.

[5] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Machine Learning*, vol. 56, no. 1, pp. 9–33, 2004. 1

[6] A. I. Zecevic and D. D. Siljak, "Global low-rank enhancement of decentralized control for large-scale systems," *IEEE Trans. Automatic Control*, vol. 50, no. 5, pp. 740–744, 2005. 1

[7] P. Benner and T. Breiten, "Low-rank methods for a class of generalized lyapunov equations and related issues," *Numerische Mathematik*, vol. 124, no. 3, pp. 441–470, 2013. 1

[8] X. Liang, X. Ren, Z. Zhang, and Y. Ma, "Repairing sparse low-rank texture," in *IEEE European Conf. Computer Vision (ECCV)*, 2012. 1

[9] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low-rank matrix completion," in *IEEE Comp. Vision and Pattern Recognition (CVPR)*, 2010.

[10] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Trans. Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 4729–4743, 2013. 1

[11] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936. 1, 5

[12] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," *arXiv preprint arXiv:1010.4237*, 2010. 1, 2

[13] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Analysis and Applications*, vol. 35, no. 1, pp. 225–253, 2014.

[14] A. E. Waters, A. C. Sankaranarayanan, and R. G. Baraniuk, "SpaRCS: Recovering low-rank and sparse matrices from compressive measurements." in *Adv. Neural Info. Processing Systems*, 2011. 1

[15] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, and G. Liu, "Tensor completion using total variation and low-rank matrix factorization," *Info. Sciences*, vol. 326, pp. 243–257, 2016. 1

[16] W. He, H. Zhang, L. Zhang, and H. Shen, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration," *IEEE Trans. Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 178–188, 2015.

[17] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X.-L. Zhao, and D. Meng, "Hyperspectral image restoration via total variation regularized low-rank tensor decomposition," *IEEE J. Selected Topics in Appl. Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1227–1243, 2017. 1

[18] X.-L. Zhao, W.-H. Xu, T.-X. Jiang, Y. Wang, and M. K. Ng, "Deep plug-and-play prior for low-rank tensor completion," *Neurocomputing*, vol. 400, pp. 137–149, 2020. 1, 3

[19] Z. Ke, W. Huang, J. Cheng, Z. Cui, S. Jia, H. Wang, X. Liu, H. Zheng, L. Ying, Y. Zhu *et al.*, "Deep low-rank prior in dynamic mr imaging," *arXiv preprint arXiv:2006.12090*, 2020. 1, 3

[20] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991. 1, 8

[21] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *IEEE Comp. Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 4, 6, 7

[22] R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," *arXiv preprint arXiv:1810.03982*, 2018. 2, 4, 6

[23] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and J. Science*, vol. 2, no. 11, pp. 559–572, 1901. 2

[24] W. H. Lawton and E. A. Sylvestre, "Self modeling curve resolution," *Technometrics*, vol. 13, no. 3, pp. 617–633, 1971. 2

[25] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, 1967, pp. 281–297. 2

[26] D. W. Tufts and A. A. Shah, "Estimation of a signal waveform from noisy data using low-rank approximation to a data matrix," *IEEE Trans. Signal Processing*, vol. 41, no. 4, pp. 1716–1721, 1993. 2

[27] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising based on low-rank and sparse representations," in *IEEE Intl. Geoscience and Remote Sensing Symposium (IGARSS)*, 2016. 2

[28] Z. Yuan and E. Oja, "Projective nonnegative matrix factorization for image compression and feature extraction," in *Scandinavian Conf. Image Analysis*, 2005. 2

[29] Y. Xu, Z. Wu, J. Li, A. Plaza, and Z. Wei, "Anomaly detection in hyperspectral images based on low-rank and sparse representation," *IEEE Trans. Geoscience and Remote Sensing*, vol. 54, no. 4, pp. 1990–2000, 2015. 2

[30] S. Barratt, Y. Dong, and S. Boyd, "Low-rank forecasting," *arXiv preprint arXiv:2101.12414*, 2021. 2

[31] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999. 2

[32] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian cp factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1751–1763, 2015. 2

[33] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S.-I. Amari, "Bayesian robust tensor factorization for incomplete multiway data," *IEEE Trans. Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 736–748, 2015. 2

[34] X. Chen, Z. Han, Y. Wang, Q. Zhao, D. Meng, and Y. Tang, "Robust tensor factorization with unknown noise," in *IEEE Comp. Vision and Pattern Recognition (CVPR)*, 2016. 2

[35] M. Bahri, Y. Panagakis, and S. Zafeiriou, "Robust kronecker component analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2365–2379, 2018. 2, 7, 8

[36] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 40, no. 8, pp. 1888–1902, 2017. 2

[37] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, "A deep semi-nmf model for learning hidden representations," in *Intl. Conf. Machine Learning*, 2014. 3

[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. 3

[39] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro, "Implicit bias of gradient descent on linear convolutional networks," *arXiv preprint arXiv:1806.00468*, 2018. 3

[40] M. Aittala, P. Sharma, A. Yedidia, L. Murmann, W. Freeman, G. Wornell, and F. Durand, "Computational mirrors: Blind inverse light transport by deep matrix factorization," in *Adv. Neural Info. Processing Systems*, 2019. 3

[41] J. Bacca, Y. Fonseca, and H. Arguello, "Compressive spectral image reconstruction using deep prior and low-rank tensor representation," *Appl. Optics*, vol. 60, no. 14, pp. 4197–4207, 2021. 3, 7

[42] B. Arad and O. Ben-Shahar, "Sparse recovery of hyperspectral signal from natural rgb images," in *IEEE European Conf. Computer Vision (ECCV)*, 2016. 3, 7

[43] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *IEEE European Conf. Computer Vision (ECCV)*, 2002. 4

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Intl. Conf. Learning Representations*, 2015. 4

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Info. Processing Systems*, 2019. 4, 7

[46] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," *arXiv preprint arXiv:1704.00109*, 2017. 6

[47] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2008. 6, 8, 10

[48] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *SIAM Intl. Conf. Data Mining*, 2005. 6

[49] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 6

[50] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009. 6

[51] A. Cichocki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Trans. Fundamentals of Electronics, Communications and Comp. Sciences*, vol. 92, no. 3, pp. 708–721, 2009. 6, 7, 10

[52] M. O'Toole and K. N. Kutulakos, "Optical computing for fast light transport analysis." *ACM Trans. Graphics*, vol. 29, no. 6, pp. 164:1–12, 2010. 7

[53] V. Saragadam and A. Sankaranarayanan, "KRISM—Krylov subspace-based optical computing of hyperspectral images," *ACM Trans. Graphics*, vol. 38, no. 5, pp. 148:1–14, 2019. 7

[54] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary," in *IEEE Intl. Conf. Computer Vision (ICCV)*, 2011. 7, 9

[55] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007. 7, 8

[56] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997. 7, 9

[57] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle *et al.*, "The cancer imaging archive (tcia): maintaining and operating a public information repository," *J. Digital Imaging*, vol. 26, no. 6, pp. 1045–1057, 2013. 7

[58] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in

imaging," *IEEE Trans. Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017. 7, 8

[59] K. Gong, C. Catana, J. Qi, and Q. Li, "PET image reconstruction using deep image prior," *IEEE Trans. Medical Imaging*, vol. 38, no. 7, pp. 1655–1665, 2018. 7

[60] W. I. of Sciences, "Computer vision databases," http://www.wisdom.weizmann.ac.il/~vision/FaceBase/. 8

[61] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994. 8

[62] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018. 8, 10

[63] I. Choi, D. S. Jeon, G. Nam, D. Gutierrez, and M. H. Kim, "High-quality hyperspectral reconstruction using a spectral prior," *ACM Trans. Graphics*, vol. 36, no. 6, pp. 218:1–13, 2017. 9