Rethink Autoencoders: Robust Manifold Learning

Taihui Li [1] **Rishabh Mehta** [1] **Zecheng Qian** [1] **Ju Sun** [1]

Abstract

PCA can be made robust to data corruption, i.e., robust PCA. What about the deep autoencoder, as a nonlinear generalization of PCA? This further motivates us to "reinvent" a factorization-based PCA as well as its nonlinear generalization. Focusing on sparse corruption, we model the sparsity structure explicitly using the ℓ_1 norm to obtain various robust formulations. For linear data, robust factorization performs comparably to the seminal convex formulation of robust PCA, whereas robust autoencoders provably fail. For nonlinear data, we perform careful experimental evaluation of robust deep autoencoders and robust nonlinear factorization for corruption removal on natural images. Both schemes can remove a considerable level of sparse corruption and effectively reconstruct the clean images.

1. Introduction

From robust PCA to robust manifold learning. Despite its practical popularity, the classic PCA is well known to be sensitive to data corruption and outliers (Huber & Ronchetti, 2009). Over the last decade, a number of robust formulations of PCA (Candès et al., 2011; Chandrasekaran et al., 2011; Xu et al., 2012; Lerman et al., 2014) have been proposed, taking a common geometric view: finding a best-fit subspace to data that are contaminated by gross errors.

Can we do similar things for data lying on low-dimensional manifolds, which are often considered as natural generalizations of subspaces? For example, this is relevant for visual data: visual appearance of an object is determined by only few explanatory factors (e.g., pose, deformation, illumination, etc.) and its images lie near a low-dimensional manifold (Donoho & Grimes, 2005). On the other hand, cast shadows, occlusions, and defects in imaging sensors do not correspond to any explanatory factors of interest, but lead to structured erasure and irrelevant imputation of image pixels or parts—called corruptions, gross errors, and outliers. So we indeed have a low-dimensional manifold, or perhaps manifolds, and sparse corruptions and rare outliers.

Motivation: robustness against natural corruption & perturbation. Vision systems based on deep neural networks (DNNs) can be jeopardized by maliciously constructed small perturbations (Szegedy et al., 2013; Heaven, 2019; Miller et al., 2019). While this concerns mission-critical applications, it does not account for practical corruption and perturbation: e.g., salt-and-pepper noise, motion blurring, weather conditions, camera tilting, occlusions, that are not maliciously constructed and need not be small in any sense. Numerous recent works (Hosseini et al., 2017; Dodge & Karam, 2019; Geirhos et al., 2019; Hendrycks et al., 2019b; Hendrycks & Dietterich, 2019; Xie et al., 2019) have suggested DNN-based systems are also not robust against such corruption and perturbation.

Our contribution. In this paper, we focus on an unsupervised setting and show that robust recovery of nonlinear visual data subject to sparse corruption is possible. We model the sparsity structure explicitly and derive nonlinear generalizations of PCA and the robust variants—*robust deep autoencoder* and *robust nonlinear factorization*. We highlight the efficacy of nonlinear factorization, as an attractive simplification of autoencoders. In our experiments, the proposed robust models show great promise for performing robust visual manifold learning against sparse corruption.

2. Method

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathbb{R}^n$ be a set of zero-centered data points and write $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{n \times m}$. In this paper, we focus on additive sparse corruption, i.e. $\widehat{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{C}$, where $\boldsymbol{C}$ is a sparse matrix containing very few nonzero entries. A crucial component in our formulations is to explicitly model the sparsity in $\boldsymbol{C}$. While there are many choices, we use the classic ℓ_1 norm $\|\cdot\|_1$ (Candès, 2014). Below, we always assume $p \leq n$.

2.1. PCA, Autoencoder, and Factorization

The classic PCA uses the top eigenspace of $\boldsymbol{X}\boldsymbol{X}^\intercal$, or equivalently, the top (left) singular-vector space of $\boldsymbol{X}$ to fit the

[1]Department of Computer Science and Engineering, University of Minnesota, Twin Cites, USA. Correspondence to: Taihui Li <lixx5027@umn.edu>, Ju Sun <jusun@umn.edu>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.

point set.

The autoencoder is a classic neural network model for unsupervised learning, consisting of only a linear encoder $\boldsymbol{A}$ that typically maps the input into a low-dimensional "code", and a linear decoder $\boldsymbol{B}$ that maps the "code" back to the original space, so that $\boldsymbol{X} \approx \boldsymbol{ABX}$. This is often formulated as:

$$\min_{\boldsymbol{A}\in\mathbb{R}^{p\times n},\boldsymbol{B}\in\mathbb{R}^{n\times p}} \|\boldsymbol{X}-\boldsymbol{BAX}\|_F^2 \quad \text{(autoencoder)}.$$

The seminal paper Baldi & Hornik (1989) is the first to identify a clear connection of the autoencoder with PCA: under mild nondegeneracy conditions, any $\boldsymbol{B}$ at a local minimizer recovers the top rank-p eigenspace of $\boldsymbol{XX}^\intercal$. One can hence find the desired subspace basis using simple numerical methods.

Yet another popular formulation is based on factorization:

$$\min_{\boldsymbol{B}\in\mathbb{R}^{n\times p},\boldsymbol{Z}\in\mathbb{R}^{p\times m}} \|\boldsymbol{X}-\boldsymbol{BZ}\|_F^2 \quad \text{(factorization)}.$$

Same as the autoencoder, it can be proved that the top rank-p eigenspace can be recovered from any local minimizer (proved in Theorem A.1). Therefore, from the geometric viewpoint, both the autocoder and factorization perform PCA on the given data $\boldsymbol{X}$.

2.2. Robust PCA, Robust Autoencoder, and Robust Factorization

When we observe the corrupted data $\widehat{\boldsymbol{X}}$, we want to explicitly account for the sparse corruption $\boldsymbol{C}$. A convex relaxation proposed by a couple of breakthrough papers (Candès et al., 2011; Chandrasekaran et al., 2011) is

$$\min_{\boldsymbol{X},\boldsymbol{C}} \|\boldsymbol{X}\|_* + \lambda\|\boldsymbol{C}\|_1 \text{ s. t. } \widehat{\boldsymbol{X}} = \boldsymbol{X}+\boldsymbol{C}, \text{(convex RPCA)}$$

based on the observation that $\boldsymbol{X}$ is low-rank if the data points it contains can be well approximated by a linear subspace. The norms $\|\cdot\|_*$ and $\|\cdot\|_1$ are used to promote the low-rankness and sparsity structures, respectively.

Here, we consider alternative formulations built directly on the autoencoder and factorization. This helps us to generalize easily to the manifold setting later. Now in autoencoder and factorization, we hope that $\boldsymbol{BA}\widehat{\boldsymbol{X}}$ and $\boldsymbol{BZ}$ recover the low-rank part of $\widehat{\boldsymbol{X}}$, i.e., $\boldsymbol{X}$, so that the differences $\widehat{\boldsymbol{X}} - \boldsymbol{BA}\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{X}} - \boldsymbol{BZ}$ both represent the sparse corruption $\boldsymbol{C}$. Thus, it is natural to consider

$$\min_{\boldsymbol{A}\in\mathbb{R}^{p\times n},\boldsymbol{B}\in\mathbb{R}^{n\times p}} \left\|\widehat{\boldsymbol{X}}-\boldsymbol{BA}\widehat{\boldsymbol{X}}\right\|_1 \text{(robust autoencoder)},$$
$$\min_{\boldsymbol{B}\in\mathbb{R}^{n\times p},\boldsymbol{Z}\in\mathbb{R}^{p\times m}} \left\|\widehat{\boldsymbol{X}}-\boldsymbol{BZ}\right\|_1 \text{(robust factorization)}.$$

Robust autoencoder will not work; we make a formal argument in Appendix B and direct simulation also confirms this. By contrast, robust factorization works in a regime even larger than the convex relaxation; again, see Appendix B. Robust factorization or its variants as an alternative formulation for robust PCA have been briefly discussed in (Li et al., 2018; Charisopoulos et al., 2019; Brooks et al., 2013).

2.3. Generalizations for Robust Manifold Learning

Now we move to the nonlinear world and assume the data lie on low-dimensional manifolds. The deep autoencoder is a simple generalization of the autoencoder into a nonlinear form:

$$\min_{\boldsymbol{W},\boldsymbol{V}} \|\boldsymbol{X}-g_{\boldsymbol{V}}\circ f_{\boldsymbol{W}}(\boldsymbol{X})\|_F^2 \quad \text{(deep autoencoder)}.$$

The idea is simply replacing encoder $\boldsymbol{A}$ and decoder $\boldsymbol{B}$ by nonlinear mappings $\boldsymbol{f_W}$ and $g_{\boldsymbol{V}}$ paramterized by $\boldsymbol{W}$ and $\boldsymbol{V}$, respectively. The is exactly the autoencoder that people use for nonlinear dimension reduction (Hinton & Salakhutdinov, 2006), information retrieval (Salakhutdinov & Hinton, 2009), and outlier detection (Sakurada & Yairi, 2014)—successful training of deep autoencoders in Hinton & Salakhutdinov (2006) has spawned the current resurgence of DNNs.

The same idea also leads to a nonlinear generalization of factorization:

$$\min_{\boldsymbol{V},\boldsymbol{Z}} \|\boldsymbol{X}-g_{\boldsymbol{V}}(\boldsymbol{Z})\|_F^2 \quad \text{(nonlinear factorization)}.$$

This formulation is much less popular than the deep autoencoder, although there exist few precursors (Tan & Mayrovouniotis, 1995; Fan & Cheng, 2018; Bojanowski et al., 2019; Park et al., 2019; Heckel & Hand, 2018). Arguably, it admits a more direct interpretation that deep autoencoders: it directly learns the explanatory factors $\boldsymbol{Z}$ and the generative process modeled as a neural network $g_{\boldsymbol{V}}$.

Inspired by the robust autoencoder and robust factorization, we propose natural robust extensions to deep autoencoder and nonlinear factorization:

$$\min_{\boldsymbol{W},\boldsymbol{V}} \left\|\widehat{\boldsymbol{X}}-g_{\boldsymbol{V}}\circ f_{\boldsymbol{W}}\left(\widehat{\boldsymbol{X}}\right)\right\|_1 \text{(robust deep autoencoder, RDAE)},$$
$$\min_{\boldsymbol{V},\boldsymbol{Z}} \left\|\widehat{\boldsymbol{X}}-g_{\boldsymbol{V}}(\boldsymbol{Z})\right\|_1 \text{(robust nonlinear factorization, RNLF)}.$$

In Section 2.2 we showed both theoretically and empirically that the robust autoencoder does not solve the robust PCA problem. Here, we nonetheless do not rule out RDAE, as the linear algebra argument is very specific to the linear case and things may change dramatically once we allow nonlinear mappings. In fact, as we shall see in the experiment (Section 3), RDAE and RNLF mostly perform comparably.

2.4. Generalization by Others

(Zhou & Paffenroth, 2017) proposes another robust version

of deep autoencoder as

$$\min_{\boldsymbol{W},\boldsymbol{V},\boldsymbol{X}} \|\boldsymbol{X} - g_{\boldsymbol{V}} \circ f_{\boldsymbol{W}}(\boldsymbol{X})\|_F + \lambda \|\widehat{\boldsymbol{X}} - \boldsymbol{X}\|_1,$$

where they introduce an additional variable $\boldsymbol{X}$ to account for the clean data. In our RNLF, this is conveniently represented as $g_{\boldsymbol{V}}(\boldsymbol{Z})$ and hence our formulation is much neater. (Chalapathy et al., 2017) proposes a modification to the shallow autoencoder

$$\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{S}} \left\|\widehat{\boldsymbol{X}} - \boldsymbol{A}\sigma(\boldsymbol{B}(\widehat{\boldsymbol{X}})) - \boldsymbol{S}\right\|_F + \lambda_1\left(\|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}\|_F^2\right) + \lambda_2\|\boldsymbol{S}\|_1,$$

where the $\|\boldsymbol{A}\|_F^2 + \|\boldsymbol{B}\|_F^2$ was inspired by nuclear norm as regularizer for low-rankness. Compared to both, our formulation is much cleaner.

3. Experiments

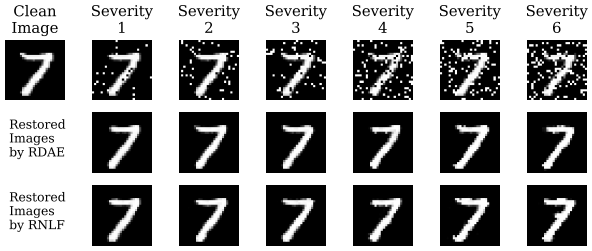

Figure 1. MNIST restored images by RDAE and RNLF for different severity levels of Impulse Noise.

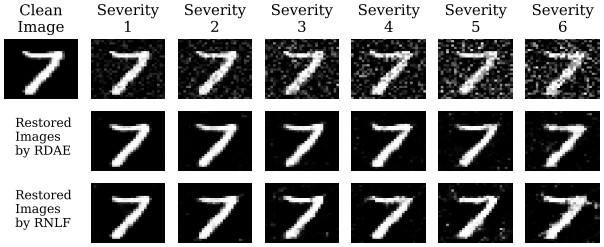

Figure 2. MNIST restored images by RDAE and RNLF for different severity levels of Gaussian Noise.

3.1. Recovery: visual results

We verify the effectiveness of *robust deep autoencoder* (RDAE) and *robust nonlinear factorization* (RNLF) through preliminary experiments on a number of image benchmark datasets: MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), CIFAR10 and CIFAR100 (Krizhevsky et al., 2009). We simulate our corrupted images by applying Salt-and-Pepper Impulse Noise

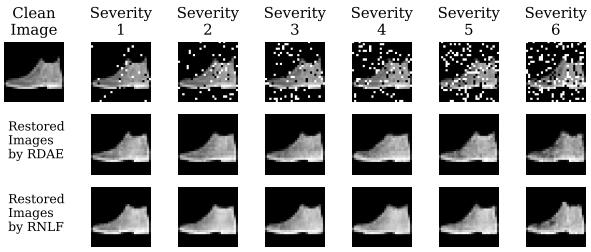

Figure 3. Fashion-MNIST restored images by RDAE and RNLF for different severity levels of Impulse Noise.

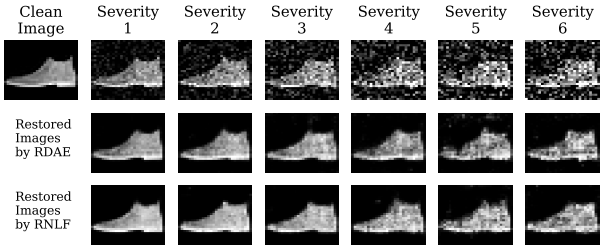

Figure 4. Fashion-MNIST restored images by RDAE and RNLF for different severity levels of Gaussian Noise.

and Gaussian noise to the test data of these benchmark datasets with different severity levels, ranging from benign to severe corruptions. Different corruption severity levels and brief explanations are included in Table 2. In Fig. 1, Fig. 2, Fig. 3, and Fig. 4, we show randomly selected examples of restored images by our models along with the corresponding clean and corrupted images. It is clear that our models are able to recover images reasonable well, discarding the sparse noise present in them.

We implement RDAE and RNLF with the deep learning framework PyTorch 1.4. We build our neural network models using convolutional neural networks (CNNs), choose sigmoid activation $\sigma(x) = \frac{1}{1+e^{-x}}$ or $\tanh$ activation $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ in the output layer depending on the normalization used in image preprocessing, learn and update the network parameters and/or the latent codes using Adam (Kingma & Ba, 2014) for all network models. We also exploit the learning rate scheduler with a decay factor of 0.1 for every 100 epochs. The detailed network architecture along with the hyper-parameters are provided in Appendix C.

3.2. Recovery measured by classification accuracy

In this experiment, we evaluate the denoising effectiveness of RDAE and RNLF by feeding the restored images to DNN-based visual recognition systems. We use the recognition performances on clean and corrupted images as baseline references to measure how much our methods can help mit-

Table 1. Object recognition performance of DNN-based models on clean, corrupted, and restored images by RDAE and RNLF. From the top to bottom: 1^{st} group: MNIST; 2^{nd} group: Fashion-MNIST; 3^{rd} group: CIFAR10; 4^{th} group: CIFAR100.

	Impulse Noise & Gaussian Noise															
Network	**Clean Image**		**Severity 1**		**Severity 2**		**Severity 3**		**Severity 4**		**Severity 5**		**Severity 6**		**Mixup**[1]	
PC[2]	98.96	98.96	92.91	89.28	72.31	73.92	49.38	54.85	30.59	34.87	19.98	20.15	14.68	13.14	53.69	54.79
RDAE + PC[2]	98.70	98.61	98.55	98.68	98.41	98.69	98.10	98.57	97.81	98.39	96.98	97.90	95.88	97.37	97.87	98.35
RNLF + PC[2]	98.54	98.32	98.45	98.55	98.38	98.57	97.94	98.40	97.14	97.82	95.73	96.85	93.61	94.27	97.69	97.82
PC[2]	90.38	90.38	68.48	81.94	46.13	71.44	32.78	57.28	26.04	44.29	21.43	34.70	18.04	28.39	44.02	58.14
RDAE + PC[2]	86.91	86.47	86.43	86.48	85.93	85.30	85.48	83.41	85.17	82.30	84.60	79.45	83.89	76.24	85.58	83.63
RNLF + PC[2]	86.10	85.81	85.76	85.26	85.65	84.24	84.80	81.55	84.80	78.93	84.39	75.53	83.03	71.24	85.23	81.73
ResNet[3]	93.71	93.71	40.05	75.19	14.60	52.35	11.90	41.78	10.95	32.48	10.62	25.60	10.41	21.54	27.55	48.49
RDAE + ResNet[3]	85.55	86.19	86.19	87.05	85.75	83.75	85.50	79.80	85.29	74.39	84.81	68.30	82.22	59.04	84.85	81.28
RNLF + ResNet[3]	81.05	83.63	80.60	84.08	80.03	79.59	81.14	77.12	79.94	72.40	79.70	70.34	78.66	64.46	81.52	77.22
ResNet[3]	70.79	70.79	13.46	38.90	5.09	23.66	2.72	19.04	1.73	15.52	1.49	12.45	1.27	10.47	14.03	27.07
RDAE + ResNet[3]	53.73	54.17	53.51	56.05	54.10	52.17	52.18	46.89	52.44	41.34	53.05	34.28	51.32	28.04	53.64	50.32
RNLF + ResNet[3]	47.94	49.19	45.87	46.91	47.74	42.72	40.99	43.52	39.65	42.25	45.61	37.62	45.06	32.52	44.61	43.88

[1] Mixup: mix up corruption ratio — randomly selecting a corruption from no corruption and severity 1-6 for each image.
[2] PC: a pretrained classifier trained on clean MNIST (Fashion-MNIST) as a DNN-based model for MNIST (Fashion-MNIST) test data.
[3] Resnet with 56 layers (He et al., 2016).

Table 2. The numbers indicate severity levels of noise. For Impulse Noise, these numbers correspond to the fraction of pixels being corrupted. For Gaussian Noise, these numbers indicate the standard deviation of Gaussian distribution with zero mean.

	MNIST/Fashion-MNIST & CIFAR10/100											
	Severity 1		**Severity 2**		**Severity 3**		**Severity 4**		**Severity 5**		**Severity 6**	
Impulse Noise	0.05	0.05	0.10	0.10	0.15	0.15	0.20	0.20	0.25	0.25	0.30	0.30
Gaussian Noise	0.10	0.04	0.15	0.06	0.20	0.07	0.25	0.08	0.30	0.09	0.35	0.10

igate the effects of corruption on classification. For each corruption type, Impulse Noise and Gaussian Noise, we conduct experiments on each individual corruption severity and mixup corruption severity, respectively. Table 1 shows that our methods can reconstruct both clean and corrupted images reasonable well, without loss of any prime information. As the corruption severity increases, the classification accuracy of DNN-based models on the corrupted images drop dramatically, while we can see graceful degradation of the performance on the restored images by RDAE and RNLF—which is always substantially better than the corrupted images, even when the corruption is severe.

4. Related Work

Our focus here is on robustness for manifold learning, which is intrinsically an unsupervised learning problem. Several public datasets have been constructed to foster research on average-case robustness in the supervised visual recognition tasks (Hendrycks et al., 2019b; Hendrycks & Dietterich, 2019; Xie et al., 2019). The corruptions and perturbations considered in these datasets are far broader than the sparse corruptions we consider here. The recent works (Geirhos et al., 2018; Lopes et al., 2019; Hendrycks et al., 2019a) addressing the robustness issue on these datasets have almost exclusively hinged on data augmentation and model expansion, which is plausible and promising but seems unsustainable. Practical corruptions and perturbations tend to much less amenable to low-dimensional models than the informative visual contents themselves, raising the concern if data augmentation can ultimately tame the curse of dimensionality. Here, instead of the data-driven approach, we advocate a model-based approach to explicitly model the structures in the corruptions and perturbations.

Our corruption removal task here may remind of denoising autoencoders (Vincent et al., 2010). They are disparate, as in training denoising autoencoders, clean data are required and multiple iid noisy versions of the same data point are seen—it is a supervised setting. Here, only one corrupted version of each data point is seen and no clean data are available—an unsupervised setting.

Nonlinear factorization can be traced back at least to (Tan & Mayrovouniotis, 1995), and recently has been deployed for deep nonlinear matrix factorization (Fan & Cheng, 2018), generative models (Bojanowski et al., 2019), and 3D shape representation (Park et al., 2019). Certain aspects of the modeling and computational advantages have just started to emerge. It is also closely related to the intriguing deep generative priors (Ulyanov et al., 2018; Bora et al., 2017; Heckel & Hand, 2018). We are the first to extend the idea for robust representation learning.

Acknowledgments

The authors acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this paper.

References

Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, jan 1989. doi: 10.1016/0893-6080(89)90014-2.

Bojanowski, P., Joulin, A., Lopez-Paz, D., and Szlam, A. Optimizing the latent space of generative networks. *arXiv:1707.05776*, 2019.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 537–546. JMLR. org, 2017.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis, 2018.

Brooks, J., Dulá, J., and Boone, E. A pure $l1$-norm principal component analysis. *Computational Statistics & Data Analysis*, 61:83–98, may 2013. doi: 10.1016/j.csda.2012.11.007.

Candès, E. J. Mathematics of sparsity (and a few other things). In *Proceedings of the International Congress of Mathematicians, Seoul, South Korea*, volume 123. Citeseer, 2014.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, may 2011. doi: 10.1145/1970392.1970395.

Chalapathy, R., Menon, A. K., and Chawla, S. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 36–51. Springer, 2017.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, apr 2011. doi: 10.1137/090761793.

Charisopoulos, V., Chen, Y., Davis, D., Díaz, M., Ding, L., and Drusvyatskiy, D. Low-rank matrix recovery with composite optimization: good conditioning and rapid convergence. *arXiv:1904.10020*, 2019.

Dodge, S. and Karam, L. A study and comparison of human and deep learning recognition performance under visual distortions. *arXiv:1705.02498*, 2019.

Donoho, D. L. and Grimes, C. Image manifolds which are isometric to euclidean space. *Journal of Mathematical Imaging and Vision*, 23(1):5–24, jul 2005. doi: 10.1007/s10851-005-4965-4.

Fan, J. and Cheng, J. Matrix completion by deep matrix factorization. *Neural Networks*, 98:34–41, feb 2018. doi: 10.1016/j.neunet.2017.10.007.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Geirhos, R., Janssen, D. H. J., Schütt, H. H., Rauber, J., Bethge, M., and Wichmann, F. A. Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv:1706.06969*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Heaven, D. Why deep-learning ais are so easy to fool. *Nature*, 574 (7777):163, 2019.

Heckel, R. and Hand, P. Deep decoder: Concise image representations from untrained non-convolutional networks. *arXiv:1810.03982*, 2018.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019a.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. *arXiv:1907.07174*, 2019b.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Hosseini, H., Xiao, B., and Poovendran, R. Google's cloud vision api is not robust to noise. *arXiv:1704.05051*, 2017.

Huber, P. J. and Ronchetti, E. M. *Robust Statistics*. John Wiley & Sons, Inc., jan 2009. doi: 10.1002/9780470434697.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lerman, G., McCoy, M. B., Tropp, J. A., and Zhang, T. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, sep 2014. doi: 10.1007/s10208-014-9221-0.

Li, X., Zhu, Z., So, A. M.-C., and Vidal, R. Nonconvex robust low-rank matrix recovery. *arXiv:1809.09237*, 2018.

Lopes, R. G., Yin, D., Poole, B., Gilmer, J., and Cubuk, E. D. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

Miller, D. J., Xiang, Z., and Kesidis, G. Adversarial learning in statistical classification: A comprehensive review of defenses against attacks. *arXiv:1904.06292*, 2019.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Sakurada, M. and Yairi, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pp. 4–11, 2014.

Salakhutdinov, R. and Hinton, G. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, jul 2009. doi: 10.1016/j.ijar.2008.11.006.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tan, S. and Mayrovouniotis, M. L. Reducing data dimensionality through optimizing neural network inputs. *AIChE Journal*, 41 (6):1471–1480, jun 1995. doi: 10.1002/aic.690410612.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xie, Q., Hovy, E., Luong, M.-T., and Le, Q. V. Self-training with noisy student improves imagenet classification. *arXiv:1911.04252*, 2019.

Xu, H., Caramanis, C., and Sanghavi, S. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, may 2012. doi: 10.1109/tit.2011.2173156.

Zhou, C. and Paffenroth, R. C. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674, 2017.

A. Factorization as PCA

Theorem A.1. *Assume that $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ (with $m \geq n$) is full rank and that $\boldsymbol{X}\boldsymbol{X}^\intercal$ has distinct eigenvalues. Then, at any local minimizer of the optimization problem*

$$\min_{\boldsymbol{B}\in\mathbb{R}^{n\times p},\boldsymbol{Z}\in\mathbb{R}^{p\times m}} \|\boldsymbol{X}-\boldsymbol{B}\boldsymbol{Z}\|_F^2, \quad p \leq n, \tag{1}$$

$\boldsymbol{B}$ spans the top rank-p eigenspace of $\boldsymbol{X}\boldsymbol{X}^T$. In fact, all these local minimizers are also global minimizers.

Proof. The first-order optimality condition is

$$\boldsymbol{B}^\intercal\boldsymbol{X} = \boldsymbol{B}^\intercal\boldsymbol{B}\boldsymbol{Z}, \tag{2}$$
$$\boldsymbol{X}\boldsymbol{Z}^\intercal = \boldsymbol{B}\boldsymbol{Z}\boldsymbol{Z}^\intercal. \tag{3}$$

By Eq. (2), at any first-order critical point,

$$\begin{aligned}\boldsymbol{Z} &= (\boldsymbol{B}^\intercal\boldsymbol{B})^\dagger\boldsymbol{B}^\intercal\boldsymbol{X} + \big(\boldsymbol{I}-\boldsymbol{B}^\dagger\boldsymbol{B}\big)\boldsymbol{L}\\ &= \boldsymbol{B}^\dagger\boldsymbol{X} + \big(\boldsymbol{I}-\boldsymbol{B}^\dagger\boldsymbol{B}\big)\boldsymbol{L}\end{aligned} \tag{4}$$

for arbitrary $\boldsymbol{L}$'s with a compatible dimension, where $(\cdot)^\dagger$ denotes the matrix pseudoinverse. Thus, at any first-order critical point,

$$\boldsymbol{B}\boldsymbol{Z} = \boldsymbol{B}\boldsymbol{B}^\dagger\boldsymbol{X} + \boldsymbol{B}\big(\boldsymbol{I}-\boldsymbol{B}^\dagger\boldsymbol{B}\big)\boldsymbol{L} = \boldsymbol{B}\boldsymbol{B}^\dagger\boldsymbol{X}, \tag{5}$$

where we have simplified the expression using $\boldsymbol{B} = \boldsymbol{B}\boldsymbol{B}^\dagger\boldsymbol{B}$. On the other hand, by Eq. (3), we have

$$\boldsymbol{X}\boldsymbol{Z}^\intercal\boldsymbol{B}^\intercal = \boldsymbol{B}\boldsymbol{Z}\boldsymbol{Z}^\intercal\boldsymbol{B}^\intercal. \tag{6}$$

Substituting Eq. (5) into the last and write $\boldsymbol{\Sigma} \doteq \boldsymbol{X}\boldsymbol{X}^\intercal$, we obtain

$$\boldsymbol{\Sigma}\boldsymbol{B}\boldsymbol{B}^\dagger = \boldsymbol{B}\boldsymbol{B}^\dagger\boldsymbol{\Sigma}\boldsymbol{B}\boldsymbol{B}^\dagger, \tag{7}$$

where we have simplified the expression using that $\boldsymbol{B}\boldsymbol{B}^\dagger$ is symmetric. Since $\boldsymbol{B}\boldsymbol{B}^\dagger\boldsymbol{\Sigma}\boldsymbol{B}\boldsymbol{B}^\dagger$ is symmetric, we conclude that

$$\boldsymbol{\Sigma}\boldsymbol{B}\boldsymbol{B}^\dagger = \boldsymbol{B}\boldsymbol{B}^\dagger\boldsymbol{\Sigma}. \tag{8}$$

Henceforth we will write $\mathcal{P}$ to mean the orthoprojector, and $\mathcal{P}_{\boldsymbol{B}} \doteq \boldsymbol{B}\boldsymbol{B}^\dagger$ simply means orthoprojector onto the column space of $\boldsymbol{B}$. For any orthogonal matrix $\boldsymbol{U} \in O_n$, it can be easily verified by expressing $\boldsymbol{B}$ in its SVD form that

$$\mathcal{P}_{\boldsymbol{B}} = \boldsymbol{U}\mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}}\boldsymbol{U}^\intercal. \tag{9}$$

Substituting the last into Eq. (8) and write the eigendecomposition of $\boldsymbol{\Sigma}$ as $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\intercal$, we have that

$$\begin{aligned}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\intercal\boldsymbol{U}\mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}}\boldsymbol{U}^\intercal &= \boldsymbol{\Sigma}\boldsymbol{B}\boldsymbol{B}^\dagger\\ &= \boldsymbol{B}\boldsymbol{B}^\dagger\boldsymbol{\Sigma}\\ &= \boldsymbol{U}\mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}}\boldsymbol{U}^\intercal\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\intercal,\end{aligned} \tag{10}$$

which implies

$$\boldsymbol{\Lambda}\mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}} = \mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}}\boldsymbol{\Lambda}. \tag{11}$$

Obviously, $\mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}}$ must be a diagonal matrix, since $\boldsymbol{\Lambda}$ is diagonal with distinct values. Since it is an orthoprojector and hence $\mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}} = \mathcal{P}^2_{\boldsymbol{U}^\intercal\boldsymbol{B}}$, we conclude that any diagonal element of $\mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}}$ is either 1 or 0.

So at any first order critical point, if $\operatorname{rank}(\boldsymbol{B}) = r$ (with $r \leq p$), there is an index set $J \subset \{1,\dots,n\}$ so that

$$\mathcal{P}_{\boldsymbol{B}} = \boldsymbol{U}\mathcal{P}_{\boldsymbol{U}^\intercal\boldsymbol{B}}\boldsymbol{U}^\intercal = \boldsymbol{U}_J\boldsymbol{U}_J^\intercal, \tag{12}$$

where $\boldsymbol{U}_J$ consists of columns of $\boldsymbol{U}$ indexed by J. Obviously, $\operatorname{range}(\boldsymbol{B}) = \operatorname{range}(\boldsymbol{U}_J)$. So $\boldsymbol{B}$ can be written in the form

$$\boldsymbol{B} = [\boldsymbol{U}_J, \boldsymbol{0}_{n\times(p-r)}]\boldsymbol{C} \tag{13}$$

for a certain invertible $\boldsymbol{C} \in \mathbb{R}^{p\times p}$, implying that

$$\boldsymbol{B}^\dagger = \boldsymbol{C}^{-1}[\boldsymbol{U}_J^\intercal; \boldsymbol{0}_{(p-r)\times n}]. \tag{14}$$

Substituting this into Eq. (4), we obtain that

$$\begin{aligned}\boldsymbol{Z} &= \boldsymbol{C}^{-1}\begin{bmatrix}\boldsymbol{U}_J^\intercal \\ \boldsymbol{0}_{(p-r)\times n}\end{bmatrix}\boldsymbol{X} \\ &\quad + (\boldsymbol{I} - \boldsymbol{C}^{-1}\begin{bmatrix}\boldsymbol{U}_J^\intercal \\ \boldsymbol{0}_{(p-r)\times n}\end{bmatrix}[\boldsymbol{U}_J, \boldsymbol{0}_{n\times(p-r)}]\boldsymbol{C})\boldsymbol{L} \\ &= \boldsymbol{C}^{-1}\begin{bmatrix}\boldsymbol{U}_J^\intercal\boldsymbol{X} \\ \boldsymbol{0}_{(p-r)\times n}\end{bmatrix} + \boldsymbol{C}^{-1}\begin{bmatrix}\boldsymbol{0} & \\ & \boldsymbol{I}_{p-r}\end{bmatrix}\boldsymbol{C}\boldsymbol{L} \\ &= \boldsymbol{C}^{-1}\begin{bmatrix}\boldsymbol{U}_J^\intercal\boldsymbol{X} \\ \text{last } p-r \text{ rows of } \boldsymbol{C}\boldsymbol{L}\end{bmatrix}.\end{aligned}$$

We now discuss two cases separately.

- $r < p$. When $r < p$, we can perturb $\boldsymbol{L}$ slightly (remember $\boldsymbol{L}$ can be arbitrary) to obtain a full-rank matrix $\widehat{\boldsymbol{Z}}$. It is easy to verify that $\boldsymbol{B}\boldsymbol{Z} = \boldsymbol{B}\widehat{\boldsymbol{Z}}$. Now when $\widehat{\boldsymbol{Z}}$ is full-rank, $f(\boldsymbol{B}, \widehat{\boldsymbol{Z}})$ is strongly convex in $\boldsymbol{B}$. Hence we can find a $\widehat{\boldsymbol{B}}$ so that for all $\overline{\boldsymbol{B}} = (1-\varepsilon)\boldsymbol{B} + \varepsilon\widehat{\boldsymbol{B}}$ with $\varepsilon \in (0,1)$,

$$f(\overline{\boldsymbol{B}}, \widehat{\boldsymbol{Z}}) < f(\boldsymbol{B}, \widehat{\boldsymbol{Z}}) = f(\boldsymbol{B}, \boldsymbol{Z}). \tag{15}$$

  Since ε can also be made arbitrarily small and $\overline{\boldsymbol{B}} \to \boldsymbol{B}$ as $\varepsilon \to 0$, $(\boldsymbol{B}, \boldsymbol{Z})$ is a saddle point.

- $r = p$. In this case, we have

$$\boldsymbol{B} = \boldsymbol{U}_J\boldsymbol{C} \tag{16}$$
$$\boldsymbol{Z} = \boldsymbol{C}^{-1}\boldsymbol{U}_J^\intercal\boldsymbol{X} \tag{17}$$

  for an index set $J \subset \{1,\dots,n\}$ with $|J| = p$ and an invertible matrix $\boldsymbol{C} \in \mathbb{R}^{p\times p}$. There are $\binom{n}{p}$ possible choices for J, corresponding to $\binom{n}{p}$ first-order critical points.

  Next we show that when $J \neq \{1,\dots,p\}$, the corresponding critical point is not a local minimizer. First note that when $\boldsymbol{B}$ is full-rank, $\boldsymbol{Z}$ is uniquely determined by Eq. (2). So we have

$$\begin{aligned}f(\boldsymbol{B}, \boldsymbol{Z}(\boldsymbol{B})) &= \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{Z}(\boldsymbol{B})\|_F^2 \\ &= \left\|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{B}^\dagger\boldsymbol{X}\right\|_F^2 \\ &= \|\boldsymbol{X} - \mathcal{P}_{\boldsymbol{B}}\boldsymbol{X}\|_F^2 \\ &= \|\boldsymbol{X}\|_F^2 - \langle \boldsymbol{U}_J\boldsymbol{U}_J^\intercal, \boldsymbol{X}\rangle \\ &= \|\boldsymbol{X}\|_F^2 - \langle \boldsymbol{I}_J, \boldsymbol{\Lambda}\rangle \\ &= \|\boldsymbol{X}\|_F^2 - \sum_{j\in J}\lambda_j.\end{aligned}$$

  Now since $J \neq \{1,\dots,p\}$, there exists an index $\ell \in \{1,\dots,p\}$ but $\ell \notin J$ and there also exists an index $k \in J$ so that $k \geq \ell$. Now consider an ε-perturbation to $\boldsymbol{u}_k$ (i.e., the k-th column in $\boldsymbol{U}$, where we assume that the columns in $\boldsymbol{U}$ are ordered corresponding to the descending order of the eigenvalues λ_i's): $\widehat{\boldsymbol{u}_k} = (\boldsymbol{u}_k + \varepsilon\boldsymbol{u}_\ell)/\sqrt{1+\varepsilon^2}$ and replace $\boldsymbol{u}_k$ in $\boldsymbol{U}_J$ by $\widehat{\boldsymbol{u}_k}$ to form $\widehat{\boldsymbol{U}}_J$. It is obvious we still have $\widehat{\boldsymbol{U}}_J^\intercal\widehat{\boldsymbol{U}}_J = \boldsymbol{I}_J$. Now let $\widehat{\boldsymbol{B}} = \widehat{\boldsymbol{U}}_J\boldsymbol{C}$ and $\widehat{\boldsymbol{Z}} = \boldsymbol{C}^{-1}\widehat{\boldsymbol{U}}_J^\intercal\boldsymbol{X}$. The perturbed objective

$$f(\widehat{\boldsymbol{B}}, \widehat{\boldsymbol{Z}}) = \|\boldsymbol{X}\|_F^2 - \left\langle \widehat{\boldsymbol{U}}_J\widehat{\boldsymbol{U}}_J^\intercal, \boldsymbol{X}\right\rangle \tag{18}$$
$$= \|\boldsymbol{X}\|_F^2 - \left\langle \boldsymbol{U}^\intercal\widehat{\boldsymbol{U}}_J\widehat{\boldsymbol{U}}_J^\intercal\boldsymbol{U}, \boldsymbol{\Lambda}\right\rangle \tag{19}$$
$$= \|\boldsymbol{X}\|_F^2 - \sum_{j\in J,\, j\neq k}\lambda_j - \frac{1}{1+\varepsilon^2}\lambda_k - \frac{\varepsilon^2}{1+\varepsilon^2}\lambda_\ell \tag{20}$$
$$= \|\boldsymbol{X}\|_F^2 - \sum_{j\in J}\lambda_j + \frac{\varepsilon^2}{1+\varepsilon^2}(\lambda_k - \lambda_\ell). \tag{21}$$

  Since $\lambda_k - \lambda_\ell < 0$, we have that for all $\varepsilon > 0$

$$f(\widehat{\boldsymbol{B}}, \widehat{\boldsymbol{Z}}) < f(\boldsymbol{B}, \boldsymbol{Z}). \tag{22}$$

  Since ε can be arbitrarily small, $(\boldsymbol{B}, \boldsymbol{Z})$ is a saddle point.

  When $J = \{1,\dots,p\}$, the objective value is $\|\boldsymbol{X}\|_F^2 - \sum_{i=1}^p\lambda_i = \sum_{i=p+1}^n\lambda_i$. We know that $\boldsymbol{B}\boldsymbol{Z}$ has a rank no larger than p, and the best rank-p approximation to $\boldsymbol{X}$ leads to the lowest objective value, which is $\sum_{i=p+1}^n\lambda_i$. So this case corresponds to a local minimizer which is also a global minimizer.

By the above argument, when $J = \{1,\dots,p\}$, $\boldsymbol{B} = \boldsymbol{U}_{\{1,\dots,p\}}\boldsymbol{C}$ for a certain invertible matrix $\boldsymbol{C}$. So $\boldsymbol{B}$ spans the top rank-p eigenspace of $\boldsymbol{X}\boldsymbol{X}^\intercal$, completing the proof. ■

B. Robust Autoencoder vs. Robust Factorization

We first argue that robust autoencoder will not work. For simplicity, we assume $\boldsymbol{X}$ is exactly rank-p. Now if $\boldsymbol{B}\boldsymbol{A}\widehat{\boldsymbol{X}}$ were to recover the clean data $\boldsymbol{X}$, then $\mathrm{col}(\boldsymbol{B}\boldsymbol{A}\widehat{\boldsymbol{X}}) = \mathrm{col}(\boldsymbol{X})$ and $\mathrm{row}(\boldsymbol{B}\boldsymbol{A}\widehat{\boldsymbol{X}}) = \mathrm{row}(\boldsymbol{X})$, where $\mathrm{col}(\cdot)$ and $\mathrm{row}(\cdot)$ denote the column and row spaces, respectively. From $\mathrm{col}(\boldsymbol{B}\boldsymbol{A}\widehat{\boldsymbol{X}}) = \mathrm{col}(\boldsymbol{X})$ we know that $\boldsymbol{B}$ must be column full-rank, and so $\mathrm{row}(\boldsymbol{B}\boldsymbol{A}\widehat{\boldsymbol{X}}) = \mathrm{row}(\boldsymbol{A}\widehat{\boldsymbol{X}}) = \mathrm{row}(\boldsymbol{X})$. The last equality implies that $\boldsymbol{A}$ must be row full-rank, and so $\mathrm{row}(\widehat{\boldsymbol{X}}) = \mathrm{row}(\boldsymbol{X})$, which is generally not possible for $\widehat{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{C}$ with sparse but substantial $\boldsymbol{C}$. Direct simulation in Fig. 5 confirms this argument and robust autoencoder almost always fails. By contrast, robust factorization works in a larger regime than the convex relaxation,

and continues to work reasonably even if we over-specify the rank p.

C. Network Architecture

Table 3. Robust Deep Autoencoder network architecture for MNIST and Fashion-MNIST datasets.

Hyperparameters	**Model learning rate**	**Batch size**
	0.0001	100
	Code dimension	
	30	
Input	Corrupted Image	(C, H, W): (1, 28, 28)
Model	**Layer**	**Output Dimension**
Encoder	Convolution ((4,4), 64, stride 2, padding 1)	(64, 14, 14)
	Batch Normalization, LeakyReLU	(64, 14, 14)
	Convolution ((4,4), 128, stride 2, padding 1)	(128, 7, 7)
	Batch Normalization, LeakyReLU	(128, 7, 7)
	Convolution ((7,7), Code dimension, stride 1, padding 0)	(Code dimension, 1, 1)
Decoder	Deconvolution ((7,7), Code 128, stride 1, padding 0)	(128, 7, 7)
	Batch Normalization, LeakyReLU	(128, 7, 7)
	Deconvolution ((4,4), 64, stride 2, padding 1)	(64, 14, 14)
	Batch Normalization, LeakyReLU	(64, 14, 14)
	Deconvolution ((4,4), 1, stride 2, padding 1)	(1, 28, 28)
	Sigmoid	(1, 28, 28)
Output	Restored Image	(C, H, W): (1, 28, 28)

Table 4. Robust Nonlinear Factorization network architecture for MNIST and Fashion-MNIST datasets.

Hyperparameters	**(Model, Code) learning rate**	**Batch size**
	(0.0001,10)	100
	Code dimension	
	30	
Input	Latent Code & Corrupted Image	Dimensions: (Code dimension)
Model	**Layer**	**Output Dimension**
Factorization	Reshape	(Code dimension, 1, 1)
	Deconvolution ((7,7), 128, stride 1, padding 0)	(128, 7, 7)
	Batch Normalization, LeakyReLU	(128, 7, 7)
	Deconvolution ((4,4), 64, stride 2, padding 1)	(64, 14, 14)
	Batch Normalization, LeakyReLU	(64, 14, 14)
	Deconvolution ((4,4), 1, stride 2, padding 1)	(1, 28, 28)
	Sigmoid	(1, 28, 28)
Output	Restored Image	(C, H, W): (1, 28, 28)

Table 5. Robust Deep Autoencoder network architecture for CIFAR-10 and CIFAR-100 datasets. Residual blocks architecture has been adopted from (Brock et al., 2018).

Hyperparameters	**Model learning rate**	**Batch size**
	0.0005	100
	Code dimension	
	1024	
Input	Corrupted Image	(C, H, W): (3, 32, 32)
Model	**Layer**	**Output Dimension**
Encoder	Convolution ((3,3), 128, stride 1, padding 1)	(128, 32, 32)
	ResBlock down	(256, 16, 16)
	ResBlock down	(512, 8, 8)
	ResBlock down	(1024, 4, 4)
	Reshape & Linear	(Code dimension)
Decoder	Linear & Reshape	(1024, 4, 4)
	ResBlock up	(512, 8, 8)
	ResBlock up	(256, 16, 16)
	ResBlock up	(128, 32, 32)
	Batch Normalization, ReLU	(128, 32, 32)
	Convolution ((3,3), 128, stride 1, padding 1)	(3, 32, 32)
	Tanh	(3, 32, 32)
Output	Restored Image	(C, H, W): (3, 32, 32)

Table 6. Robust Nonlinear Factorization network architecture for CIFAR-10 and CIFAR-100 datasets. Residual blocks architecture has been adopted from (Brock et al., 2018).

Hyperparameters	**(Model, Code) learning rate**	**Batch size**
	(0.0005, 10)	100
	Code dimension	
	1024	
Input	Latent Code & Corrupted Image	Dimensions: (Code dimension)
Model	**Layer**	**Output Dimension**
Factorization	Linear & Reshape	(1024, 4, 4)
	ResBlock up	(512, 8, 8)
	ResBlock up	(256, 16, 16)
	ResBlock up	(128, 32, 32)
	Batch Normalization, ReLU	(128, 32, 32)
	Convolution ((3,3), 128, stride 1, padding 1)	(3, 32, 32)
	Tanh	(3, 32, 32)
Output	Restored Image	(C, H, W): (3, 32, 32)

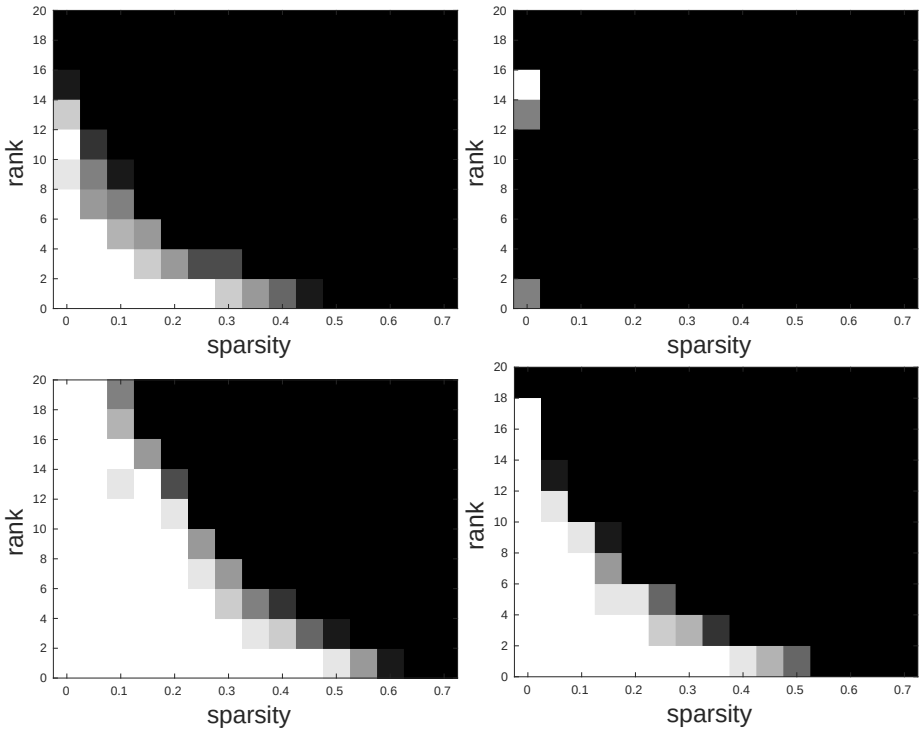

Figure 5. Phase transition plots of the various robust PCA methods. From left to right: convex relaxation, robust autoencoder, robust factorization, overparametrized robust factorization.