Robust Variational Autoencoders: Generating Noise-Free Images from Corrupted Images

Huimin Ren[1], Yun Yue[1], Chong Zhou[2], Randy C. Paffenroth[1],Yanhua Li[1], Matthew L. Weiss[1]
[1]Worcester Polytechnic Institute, [2]Microsoft Corporation
[1]{hren,yyue,rcpaffenroth,yli15,mlweiss}@wpi.edu, [2]chozh@microsoft.com

ABSTRACT

Generative models, including Variational Autoencoders, aim to find mappings from easily sampled latent spaces to intractable observed spaces. Such mappings allow one to generate new instances by mapping samples in the latent space to points in the high dimensional observed space. However, in many real-world problems, pervasive noise is commonplace and these corrupted measurements in the observed spaces can lead to substantial corruptions in the latent space. Herein, we demonstrate a novel extension to Variational Autoencoders, which can generate new samples *without access to any clean noise-free training data and pre-denoising stages.* Our work arises from Robust Principal Component Analysis and Robust Deep Autoencoders, and we split the input data into two parts, $X = L + S$, where S contains the noise and L is the noise-free data which can be accurately mapped from the latent space to the observed space. We demonstrate the effectiveness of our model by comparing it against standard Variational Autoencoders, Generative Adversarial Neural Networks, and other pre-trained denoising models.

KEYWORDS

Denoising, Variational Autoencoder, Robust Generative Model

ACM Reference Format:
Huimin Ren[1], Yun Yue[1], Chong Zhou[2], Randy C. Paffenroth[1],Yanhua Li[1], Matthew L. Weiss[1]. 2018. Robust Variational Autoencoders: Generating Noise-Free Images from Corrupted Images. In *AdvML '20: Workshop on Adversarial Learning Methods for Machine Learning and Data Mining, August 24, 2020, San Diego, CA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Generative models have been successfully applied to many application domains including image and text generation, semi-supervised learning, and domain adaption [17, 20]. Some advanced applications of generative models have been proposed such as generating plausible images from human-written descriptions [21], and recovering photo-realistic textures from heavily down-sampled images [15]. Building good generative models of realistic images is also a fundamental requirement of current AI systems[7].

In particular, there are two primary generative models, Generative Adversarial Networks (GANs) [10] and Variational Autoencoders (VAEs) [13]. A GAN trains a generator and discriminator at the same time until they reach Nash Equilibrium [10]. A VAE assumes that a collection of latent variables generates all the observations [13]. Recently, various flavors of GANs and VAEs have been proposed, which have achieved compelling results in the image generation area [2, 20].

Most generation models depend on clean noise-free input. However, anomalies and noise are commonplace and high-quality data is not always available in many cases [26]. Recently proposed research with generative models either focus on removing noise from corrupted input [6] or generating new images from available cleaned data, which can be obtained from existing off-the-shelf denoising methods [4]. This raises the question: can we combine the denoising and generation abilities of neural networks to create clean images from corrupted input data directly? It may seem intuitive to denoise first, then generate new data from denoising output. However, the final generation depends highly on the denoising, which cannot be guaranteed to pass clear images to the generative step.

To bridge the research gap of creating realistic images from noisy input data directly, we propose a novel denoising generative model, Robust Variational Autoencoder (RVAE), where an enhanced VAE takes corrupted images and generates noise-free images. Our main contributions are summarized as follows:

- We propose an extension of VAEs to robust cases where no clean, noise-free data is available. Such an extension allows denoising and inferring new instances at the same time, which, to the best of our knowledge, is a novel combination of robust models and generative models.
- Instead of separating the denoising and generation processes, our model integrates them. The denoising part offers clear inputs to the generative part and the generative part provides potential corrupted points to the denoising part.
- We demonstrate the robustness of our proposed method using different data sets such as MNIST, fashion-MNIST, and CelebA, where the input images are corrupted by different noise types, including Gaussian noise and the salt-and-pepper noise.

2 OVERVIEW AND RELATED WORK

In this section, we outline some of the key ideas from RPCA, RDA, and VAE. RPCA [5, 18] assumes observed instances and features are linearly correlated, with the exception of noise and outliers. Such a model offers a framework that can be extended and generalized from linear feature learning to non-linear, as shown in RDA [28]. VAEs, which recently gained popularity, are generative models that learn a mapping from a latent variable z to the observations X.

AdvML '20, August 24, 2020, San Diego, CA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

And In the next section, we provide technical details of our novel contribution to the above-mentioned problems, more specifically, by allowing a VAE to be embedded into the denoising framework of RPCA and RDA.

2.1 From Robust Principal Component Analysis to Robust Deep Autoencoders

RPCA is a generalization of Principal Component Analysis (PCA) [8] that attempts to alleviate the effects of grossly corrupted observations, which are unavoidable in real-world data. In particular, RPCA assumes a given data matrix X is comprised of an unknown low-rank matrix L and an unknown sparse matrix S, with the goal of discovering both L and S simultaneously.

In the literature, there exist commonly used approaches in which RPCA can be treated using a tractable convex optimization as follows [5, 18]:

$$\begin{aligned} &\min_{L,S} \|L\|_* + \lambda\|S\|_1 \\ &\text{s.t. } \|X - L - S\|_F^2 = 0, \end{aligned} \tag{1}$$

where $\|\cdot\|_*$ is the nuclear norm which is the sum of non-zero singular values of a matrix, $\|L\|_* = \sum_{i=1}^{n} \sigma_i$, $\|\cdot\|_1$ is ℓ_1 norm which is the sum of absolute values, $\|S\|_1 = \sum_{i,j} \|S_{i,j}\|$, and $\lambda > 0$ is a regularization parameter to balance L and S.

RDA maintains a deep autoencoder's ability to discover high-quality non-linear features in data but also uses the principles of RPCA to remove outliers and noise from data. The key insight of RDA is that noise and outliers are substantially incompressible and therefore cannot adequately be projected to a low-dimensional hidden layer by an autoencoder. Similar to RPCA, an RDA also splits the input data X into two parts, L and S. Here L represents the portion of the input data that is well represented by an autoencoder hidden layer and S contains noise and outliers which are difficult to reconstruct. By removing noise and outliers from X, the autoencoders can more accurately recover the remaining L. In particular, the objective function for RDA is given by the following [28]

$$\begin{aligned} &\min_{\theta,S} \|L - D_\theta(E_\theta(L))\| + \lambda\|S\|_1 \\ &\text{s.t. } X - L - S = 0, \end{aligned} \tag{2}$$

where S is the anomalous data, L is a low dimension manifold which can be accurately reconstructed by an encoder map E and a decoder map D, and λ is a parameter that tunes the level of sparsity in S. The first term is the objective function for a standard autoencoder, where L is the input, and $D_\theta(E_\theta(L))$ is the reconstruction of L.

2.2 Variational Autoencoders

A VAE assumes all the observed instances X are generated from a latent variable z but the distribution of X, $p(X)$, is intractable to compute with a limited number of observations. $p(z)$ is the prior distribution of the latent variable, $q(z|x)$ is the approximate inference mapping and $p(x|z)$ is the generative mapping. The VAE parameterizes $q(z|x)$ and $p(x|z)$ by neural networks as

$$\begin{aligned} q(z|x) &= E_{\theta_1}(x) \\ p(x|z) &= D_{\theta_2}(z), \end{aligned} \tag{3}$$

where E_{θ_1} and D_{θ_2} are two neural networks parameterized with θ_1 and θ_2 respectively. In analogy to autoencoders, E_{θ_1} is called the encoder and D_{θ_2} is called the decoder. Building on ideas from [13], the commonly used optimization function for VAE training is:

$$\min_{\theta_1,\theta_2} \|X - D_{\theta_2}(E_{\theta_1}(X))\| + KL(E_{\theta_1}(X) \mid \mathcal{N}(0,1)), \tag{4}$$

where KL represents Kullback-Leibler divergence (KL divergence) and the first term, $\|X - D_{\theta_2}(E_{\theta_1}(X))\|$ represents the standard autoencoder reconstruction error.

3 METHODOLOGY

In this section, we provide details of our model, RVAE, which builds an anomaly filter into a standard VAE. The key idea of RVAE is that noisy and clean data essentially arise from different distributions, and therefore the generation of both noisy and clean data from the same latent variables is highly unlikely. In particular, a VAE assumes all instances are generated from simple, low-dimensional distributions, but noise and anomalies share little information with clean data. This results in large errors if one tries to infer noise from generative mappings which are optimal on clean data.

We depict the structure of an RVAE in Figure 1, where denoising and inferring new instances are implemented at the same time. The noisy input is split into two parts, L and S. L represents desired clean data, and it is passed to a standard VAE that includes latent variables z, inference mapping $p(z|L)$ and generative mapping $q(L|z)$. Therefore, a VAE uses ℓ_1 norm to separate data into outliers, represented by S and nominal data, represented by L. We also provide a training algorithm for the splitting of L and S, which is a non-differentiable and non-convex problem. The denoising and generation stages finish simultaneously, as they share the same parameters from the decoder.

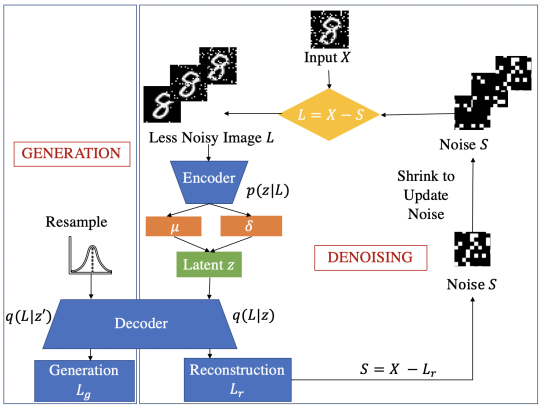


Figure 1: Structure of RVAE

3.1 Robust Variational Autoencoders

Since noisy and clean data essentially arise from different distributions, the distributions of the clean data and the noise in the latent space are also different. This key difference allows us to isolate noise from the clean data by augmenting the VAE with a filter layer. Similar to RDA in eq.2), an RVAE splits the input data X into two parts $X = L + S$, where L represents the part of normal data that is well represented by a Gaussian distribution in the latent space, and S contains noise and outliers. To achieve this property, we pose the following RVAE optimization problem:

$$\min_{\theta_1,\theta_2,L,S} \|L - D_{\theta_2}(E_{\theta_1}(L))\|_2 + KL(E_{\theta_1}(L)|\mathcal{N}(0,1)) + \lambda\|S\|_1 \quad \text{s.t. } X - L - S = 0, \tag{5}$$

where E_{θ_1} and D_{θ_2} represent inference mapping $q(z|x)$ and generative mapping $p(x|z)$ respectively, $\|L - D_{\theta_2}(E_{\theta_1}(L))\|_2$ represents a reconstruction error of the VAE part, and $KL(E_{\theta_1}(L)|\mathcal{N}(0,1))$ represents the KL-divergence, measuring differences between the distribution of the latent variables and a Gaussian distribution. The $\|S\|_1$ represents the ℓ_1 norm of S and λ controls the level of penalization on S and thus tunes the amount of data to be isolated as noise. A small λ encourages more data to be isolated as noise, and a large λ discourages data to be filtered into S.

3.2 Algorithm Training

In this section, we provide the details of our algorithm for solving the optimization problem in eq.5). The entire objective in eq.5) can be split into three parts, $\|L - D_{\theta_2}(E_{\theta_1}(L))\|_2$, $KL(E_{\theta_1}(L)|\mathcal{N}(0,1))$, and $\lambda\|S\|_1$. Each part only relies on either L or S. As a result, applying the Alternating Direction Method of Multipliers (ADMM) [19] to eq.5) is more efficient than directly training it with back-propagation[22]. In particular, the first two terms in eq.5) rely only on L and taken together are the optimization function for a standard VAE, which has been well implemented and has off-the-shelf methods readily available. In addition, back-propagation is a typical training algorithm for deep learning, and it is the method we use for optimizing the VAE. In other words, we train the first two terms, which only rely on L, together using the standard back-propagation algorithm. The S part is non-differentiable but can be phrased as a proximal gradient problem, where the solution is offered in [3]. Also, we are inspired by the training algorithm utilized in RDA that makes use of ADMM. We iteratively optimize the L part, which is a standard VAE, and the S part, which is phrased as a proximal gradient problem. Both L and S training are interspersed with projections onto the constraint manifold.

The training algorithm is provided in Algorithm 1, where $min||L - D_{\theta_2}(E_{\theta_1}(L))||_2 + KL(E_{\theta_1}(L)|N(0,1))$ is trained by back-propagation, $S = prox_{\lambda,\ell_1}(S)$ is the proximal methods, and both $L = X - S$ and $S = X - L$ are alternating projection steps. c_1 and c_2 are designed to check the convergence of the algorithm.

4 EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of our proposed RVAE on the MNIST [14], Fashion MNIST [25] and CelebA[16] data sets. We corrupt original images with salt-and-pepper noise [23]. We also justify the robustness of the model by introducing Gaussian noise. In the following section, we compare the RVAE with other benchmark models and demonstrate the robustness of RVAE by applying images with different noise ratio.

To evaluate the denoising and generation abilities of the model from corrupted images, we compare our model with two well-known generative models: traditional VAE and GAN. In addition, we compared our model with three naive approaches which remove the noise first by some well-known models such as a Low-Pass Filter (LPF), RPCA and RDA, and then generate images from the preprocessed images with a traditional VAE.

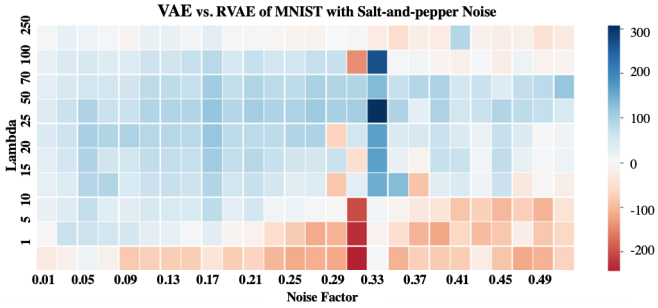


Figure 2: Comparison between VAE and RVAE

4.1 Results

4.1.1 Evaluation Metrics. To measure the quality of images generated by the model, we use the "Fréchet Inception Distance" (FID) score, which is computed by considering the similarity of two distributions X_1 and X_2 [11]. FID score has been proven as an effective measure, which correlates well with human's visual inspection[11]. Mathematically, FID assumes that instances follow a continuous multivariate Gaussian, and its formula is:

$$\|\mu_1 - \mu_2\|^2 + Tr(\sigma_1 + \sigma_2 - 2\sqrt{\sigma_1 * \sigma_2}),$$

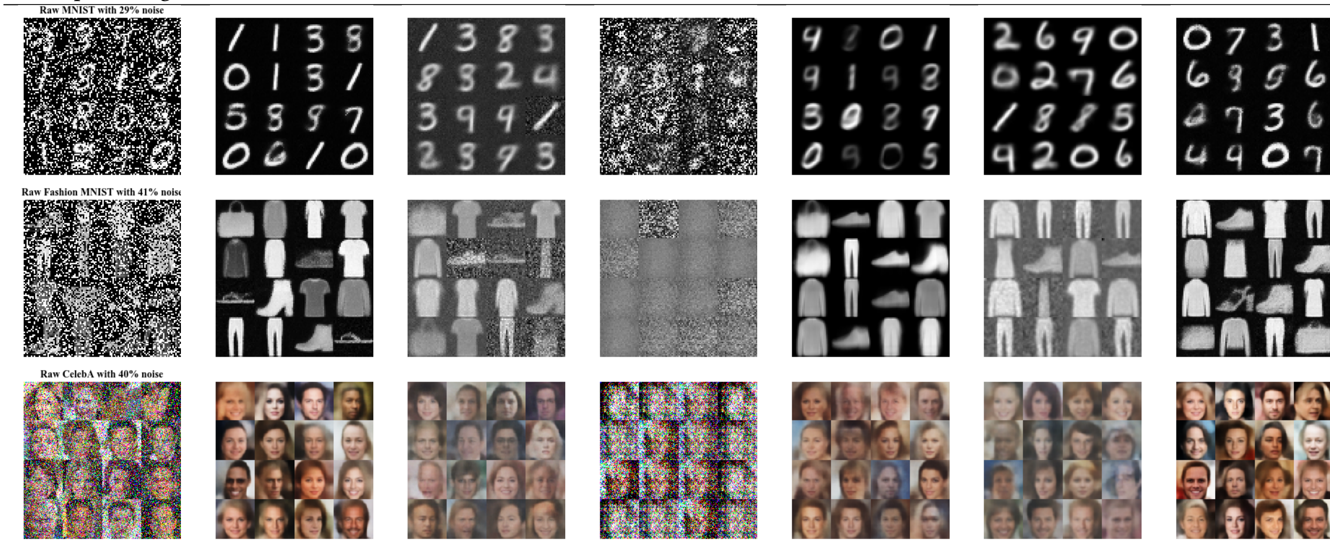
where (μ_1, σ_1) and (μ_2, σ_2) are the sample mean and covariance of X_1 and X_2. FID ranges from 0 to ∞, where a **small** FID score indicates a **high** similarity between X_1 and X_2 [11]. We calculate FID scores based on generated images and original noise-free images to measure the closeness between clean images and generated images where small FID scores indicate successful generations.

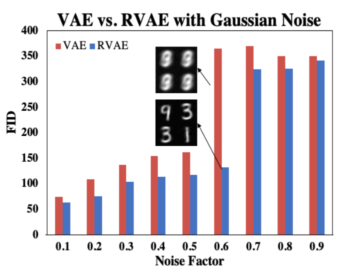
4.1.2 MNIST. Figure 2 shows the quantitative comparison of generation ability between the VAE and the RVAE, where the input contains the same level of salt-and-pepper noise. In Figure 2, a small λ indicates a large number of pixels are isolated as noise, while a large λ means only a small part of data belongs to S. Each cell of the heatmap represents the difference in the FID score between the VAE and the RVAE. Since the FID score has a negative correlation with visual quality check for the generated images [11], a *large* difference between the VAE and RVAE FID scores indicates that the generation of the RVAE is *better* than the VAE. The blue areas show that the RVAE can generate higher quality images than the VAE due to the ℓ_1 norm of S. On average, our model shows **42.47%** improved image generation when the corruption ratio ranges from 3% to 27%, and λ value varies from 10 to 100. The best result of our model shows **74.11%** improved image generation when the 33% of the pixels are corrupted. The generated examples of all the models when 29% of the pixels of the raw image are corrupted are shown in the first row at Table 1.

To evaluate the robustness of our model, we also corrupt the images with Gaussian noise. As can be seen from Figure 3a, the RVAE achieves a smaller FID score than the VAE when the noise is not too large and there is a large difference between the VAE and the RVAE with 60% corruption. Two examples are provided to show the clear and blurry generation of the RVAE and the VAE respectively. As the noise increases to 70% and more, both the RVAE and the

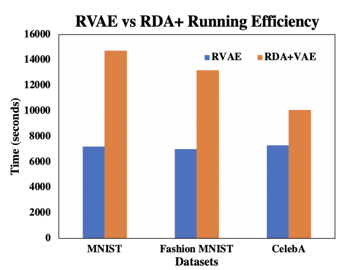
Table 1: Generated examples from the salt-and-pepper corrupted inputs of RVAE and other benchmark methods

Corrupted Image	**RVAE**	VAE	GAN	RPCA+VAE	LPF+VAE	RDA+VAE





(a) Gaussian Noise



(b) Running Efficiency Comparison

Figure 3: Results

VAE cannot generate high-quality images since excess corruption makes the input images unrecognizable.

4.1.3 Fashion MNIST. To evaluate the generative capability of the RVAE, we test our model with another data set, Fashion MNIST. Generated examples from 41% corrupted data are shown in the middle row of Table 1. These pictures show that RVAE has a strong capability to generate high visual quality images, while generative models (VAE and GAN) are unable to isolate noise and thus fail to generate clean and realistic images. Comparing with two-stage models, the quality of generation from RVAE is better than RPCA+VAE and LPF+VAE, i.e. the edge of each fashion product is much sharper and clearer in RVAE's generation while the generated images from RPCA+VAE and LPF+VAE are blurry and unrealistic. Although RVAE and RDA+VAE have similar generative quality, RVAE requires less training time to reach similar performance as the RDA+VAE, shown in Figure 3b.

4.1.4 CelebA. In addition, to evaluate our method with non-gray-scale images, we implement our model on an RGB multi-channel data set, CelebA. The last row in Table 1 shows the generated examples of our model and benchmark models when 40% of the pixels of the raw image are corrupted. Similar to the results of MNIST and Fashion MNIST, the RVAE shows strong capabilities to generate better visual-quality images from CelebA data set, even with highly corrupted inputs. One may notice that the quality of generation from RVAE and RDA+VAE is about the same. However, the RDA+VAE requires much longer training time to reach similar performance as the RVAE, shown in Figure 3b. On average, the RVAE is 38.11% faster than the RDA+VAE.

5 CONCLUSION

In this paper, we bridge the research gap between denoising and generation and show that the RVAE can generate high-quality images in the case where no clean, noise-free data is available. We introduce $X = L + S$ to split noise and clean data, where L is the input to a standard VAE and S is regularized by the ℓ_1 norm. Our training algorithm is inspired by ADMM [19], back-propagation [22] and proximal methods [3]. We evaluate the effectiveness of our model with different data sets. The experiments show that the RVAE is faithful to its name, "robust", which, with a wide range of λ selection, generates reasonable images from corrupted data with varying amounts and types of corrupting noise. Also, we show that our integrated denoising-generative model has superior performance over separated denoising and generation models.

Our future work will include both experimental and theoretical directions. Testing our robust generation model in areas other than images, such as voice and text. In our future work we also plan to extend our work to include other generative models such as GANs.

6 ACKNOWLEDGEMENTS

Huimin Ren and Yanhua Li were partly supported by NSF grants IIS-1942680 (CAREER), CNS-1657350 (CRII), and CMMI-1831140.

Algorithm 1 Training algorithm for RVAE

Input: $X \in R^{N \times n}$, ϵ

Initialize $L \in R^{N \times n}$, $S \in R^{N \times n}$ to be zero matrices,
$L + S = X$, and the variational autoencoder $D_{\theta_2}(E_{\theta_1}(\cdot))$
with randomly initialized parameters.
while $c_1 > \epsilon$ and $c_2 > \epsilon$ and iter < max_ister **do**
  $L = X - S$
  $min||L - D_{\theta_2}(E_{\theta_1}(L))||_2 + KL(E_{\theta_1}(L)|N(0,1))$
  $L = D_{\theta_2}(E_{\theta_1}(L))$
  $S = X - L$
  $S = prox_{\lambda,\ell_1}(S)$
  $c_1 = ||X - L - S||_2 / ||X||_2$
  $c_2 = ||LS - L - S|| / ||X||_2$
  $LS = L + S$
end while
return L and S

REFERENCES

[1] F. Agostinelli, M. R. Anderson, and H. Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *Advances in Neural Information Processing Systems*, pages 1493–1501, 2013.

[2] D. Berthelot, T. Schumm, and L. Metz. Began: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[5] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[6] J. Chen, J. Chen, H. Chao, and M. Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2018.

[7] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[8] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

[9] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1715–1723, 2017.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[12] D. J. Im, S. Ahn, R. Memisevic, Y. Bengio, et al. Denoising criterion for variational auto-encoding framework. In *AAAI*, pages 2059–2065, 2017.

[13] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

[15] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114. IEEE, 2017.

[16] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[17] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pages 698–707, 2018.

[18] R. Paffenroth, P. du Toit, R. Nong, L. Scharf, A. P. Jayasumana, and V. Bandara. Space-time signal processing for distributed pattern detection in sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):38–49, 2013.

[19] P. M. Pardalos. Convex optimization theory. *Optimization Methods and Software*, 25(3):487–487, 2010.

[20] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[21] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[23] S. J. Sangwine and R. E. Horne. *The colour image processing handbook*. Springer Science & Business Media, 2012.

[24] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

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

[26] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, pages 341–349, 2012.

[27] D. Zhao, B. Guo, J. Wu, W. Ning, and Y. Yan. Robust feature learning by improved auto-encoder from non-gaussian noised images. In *Imaging Systems and Techniques (IST), 2015 IEEE International Conference on*, pages 1–5. IEEE, 2015.

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

A ALGORITHM

B EXPERIMENT SETTING AND MORE RESULTS

B.1 Datasets

MNIST is a standard benchmark dataset with an integer value between 0 and 9, while Fashion MNIST is comprised of 10-class fashion products. Both of these data sets consist of 70, 000 instances with 784 features each. CelebA is a large-scale real-world face image data set which contains more than 200, 000 celebrity images. Rather than focusing on generating any specific classes of images, we instead utilize all training samples without label information.

We corrupt original images with salt-and-pepper noise [23]. We also justify the robustness of the model by introducing Gaussian noise. The salt-and-pepper noise corruption works by changing some random pixels to 0 if the original values are larger than 0.5, and 1 if the values are smaller than 0.5. As for Gaussian noise, we add a value drawn from a Gaussian distribution with a scaling factor to every pixel. Each pixel is then clipped to [0, 1]. In the following section, we compare the RVAE with other benchmark models and demonstrate the robustness of RVAE by applying images with different noise ratio.

B.2 Implementation Details

All the generative models for the same data set share the same number of parameters. The layer sizes of the RVAE are 784, 512, 256, 49, 256, 512, and 784. For MNIST and Fashion MNIST, Sigmoid is used as an activation function and the batch size of each model is 200. We train the VAE inside the RVAE for 20 epochs, with 30 iterations per epoch, to alternate projecting onto L and S, thus the total number of iterations is 600. The benchmark methods are also trained with 600 iterations. We use Adam optimizer to optimize our model with $1e$-3 as a learning rate. Since CelebA contains real color images, we add convolutional layers to the VAE model. Specifically, we use nine hidden layers which project the input data of CelebA from $64 \times 64 \times 3$ to $32 \times 32 \times 64$, $16 \times 16 \times 128$, $8 \times 8 \times 256$, $4 \times 4 \times 512$ and 100 dimensions with convolutional layers and decode it back to $64 \times 64 \times 3$ dimensions.

As mentioned above, we also include a two-stage pipeline where we apply some well-known denoising model to remove the noise

Table 2: FID scores for MNIST

Noise	**RVAE**	VAE	GAN	RPCA+	LPF+	RDA+
1%	**28.93**	111.25	158.10	95.33	127.62	47.05
9%	**43.34**	141.32	324.35	101.98	142.79	49.96
17%	80.87	203.51	351.79	115.88	168.40	**63.37**
25%	90.15	186.20	342.96	127.98	178.87	**74.57**
33%	**102.51**	413.81	357.46	143.93	187.41	132.90
41%	172.81	313.42	406.24	**141.76**	335.77	189.35
49%	230.50	335.41	385.01	**170.10**	413.02	275.38

Table 3: FID scores for Fashion MNIST

Noise	**RVAE**	VAE	GAN	RPCA+	LPF+	RDA+
1%	**84.99**	93.80	147.98	174.84	172.16	128.13
9%	68.70	122.55	255.34	137.48	146.17	**66.08**
17%	82.43	144.22	280.94	144.75	164.76	**77.20**
25%	93.62	159.41	302.00	151.92	182.61	**85.61**
33%	102.42	170.86	323.75	161.19	193.26	**91.60**
41%	111.24	189.72	372.97	165.00	208.20	**102.12**
49%	120.78	213.99	405.98	172.54	216.86	**110.02**

Table 4: FID scores for CelebA

Noise	**RVAE**	VAE	GAN	RPCA+	LPF+	RDA+
10%	**63.11**	65.90	372.30	87.79	76.37	63.85
20%	64.46	68.68	300.84	128.64	79.21	**62.66**
30%	65.71	76.99	338.42	137.59	83.63	**64.57**
40%	66.55	89.97	416.20	87.70	86.51	**64.26**
50%	70.41	115.88	341.93	175.97	93.78	**65.87**

first, then use a standard VAE to generate images from the pre-processed images. In particular, we pick three representative denoising approaches: LPF, RPCA and RDA. LPF is a standard denoising method and widely used in image-processing, while RPCA and RDA are the inspirations of our RVAE model. As we introduced in Section 2.1, both RPCA and RDA filter noise into the S part and the remaining part L, therefore contains less noise. In our experiment, L is used as the cleaned image to generate new images. Our code is available on Github at https://github.com/huiminren/RobustVAE.

B.3 Statistical Results

Table 2, 3, 4 show the quantitative comparison results with the baseline models on all MNIST, Fashion-MNIST and CelebA. RVAE outperforms in most experiments. One may notice that the FID scores of the RVAE are slightly larger than RDA+VAE at Fashion-MNIST and CelebA, the differences are small and not visually perceptible in the generated examples shown in Table 1. In addition, the RDA+VAE requires much longer training time to reach similar performance as the RVAE, shown in Figure 3b.

C MORE RELATED WORK

Generative Models and **Denoising Models** are hot topics related to our study in the literature. Though briefly introduced in Section 1, we re-summarize some of the existing works here in an organized way.

Generative Models. Generative models have achieved attractive results in multiple areas including image generation [17, 20]. There are two main approaches in generative models, GANs [10] and VAEs [13], which generate synthetic but realistic samples from noise-free inputs. Although some works address the "denoising", such as Sønderby et. al [24] who introduced noise to the input images to improve the GANs' stability and Im et. al [12] who injected noise both in input and in the stochastic latent layer of VAE to modify training criterion as an improved objective function, the noise-free and clean inputs are still critical to their objectives.

Denoising Models. In many real problems, especially in the area of audio, image or signal processing, the importance of denoising methods is never underestimated [26, 27]. In the recent literature, some deep learning works pay attention to removing noise from the noisy inputs [26]. Agostinelli et. al [1] proposed an adaptive multi-column stacked sparse denoising autoencoder, which is robust to variation in noise types. Fu et. al [9] used a deep convolutional neural network to remove rain from a single image. Our work goes one step further than these denoising-only networks by allowing the generation of new realistic samples from the cleaned images.