Generative Models for Low-rank Video Representation and Reconstruction from Compressive Measurements

Rakib Hyder and M. Salman Asif

Electrical and Computer Engineering , University of California, Riverside

Introduction

- **Compressive Sensing:** A sensing and reconstruction framework that allows us to recover a structured signal from a small number of linear/nonlinear measurements.
- **Examples:** Inpainting, denoising, super-resolution, spatial compression
- **General Problem Formulation:** Suppose we are given noisy compressive measurements of a video sequence as

$$\mathbf{y_t} = \mathbf{A_t x_t} + \mathbf{e_t}$$

where $\mathbf{x_t} \in \mathbb{R}^n$ is unknown t^{th} frame, $\mathbf{A_t} \in \mathbb{R}^{m \times n}$ is the measurement operator, $\mathbf{y_t} \in \mathbb{R}^m$ is measurement vector, and $\mathbf{e_t} \in \mathbb{R}^m$ is measurement noise for t^{th} frame of the video sequence.

- **Aim:** To recover the unknown video sequence $\mathbf{x_t}$ given the $\boldsymbol{y}_\mathbf{t}$ and $\mathbf{A_t}$.

Generative Model for Representation

- An under-determined system has infinitely many possible solutions.
- To recover the unknown signal we must restrict the solution space to a set $\boldsymbol{S} \subset \mathbb{R}^n$ that captures some known structure $\mathbf{x_t}$ is expected to obey.
- In a generative prior setup, we assume that the target image lies in the range of a trained generative model. Generative model, $G(\cdot)$ is a function that maps a latent variable $\mathbf{z} \in \mathbb{R}^k$ to the image $\mathbf{x} \in \mathbb{R}^n$.
- The compressive sensing problem can then be formulated as the following constrained optimization problem [1]:

$$\min_{\mathbf{z_t}} loss(\mathbf{y_t}, \mathbf{A_t x_t}) \quad s.t. \quad \mathbf{x_t} = G_\gamma(\mathbf{z_t})$$

where γ denotes the parameters of the generator.

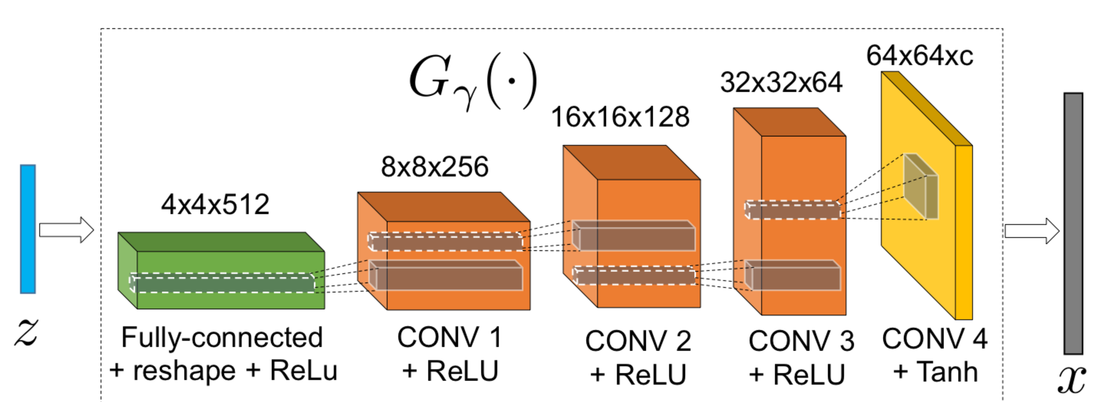

Figure 1. DCGAN [5] generator structure used in our experiments.

References

1. A. Bora, A. Jalal, E. Price, and A. Dimakis, "Compressed sensing using generative models," Proc. Int. Conf. Machine Learning, 2017.
2. C. Li, W. Yin, H. Jiang, and Y. Zhang, "An efficient augmented Lagrangian method with applications to total variation minimization," Computational Optimization and Applications, 2013.
3. R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," Proc. Int. Conf. Learning Representations (ICLR), 2018.
4. Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Deep image prior," in Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR), 2018.
5. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," Proc. Int. Conf. Learning Representations (ICLR), 2016.

Trained vs Untrained Model

- In generative prior approach we often assume that we have a trained generator which can well approximate the target image. But we cannot find trained generator for every application in practice.
- Recent research shows that convolutional generative structures alone provide good prior for reconstructing natural images [4].
- Based on this finding, we use untrained generator as a prior for solving video compressive sensing by optimizing over latent codes and network weights.

$$\min_{\mathbf{z_1},\mathbf{z_2},\ldots,\mathbf{z_T};\gamma} \sum_{t=1}^{T} ||\mathbf{y_t} - \mathbf{A_t} G_\gamma(\mathbf{z_t})||_2{}^2$$

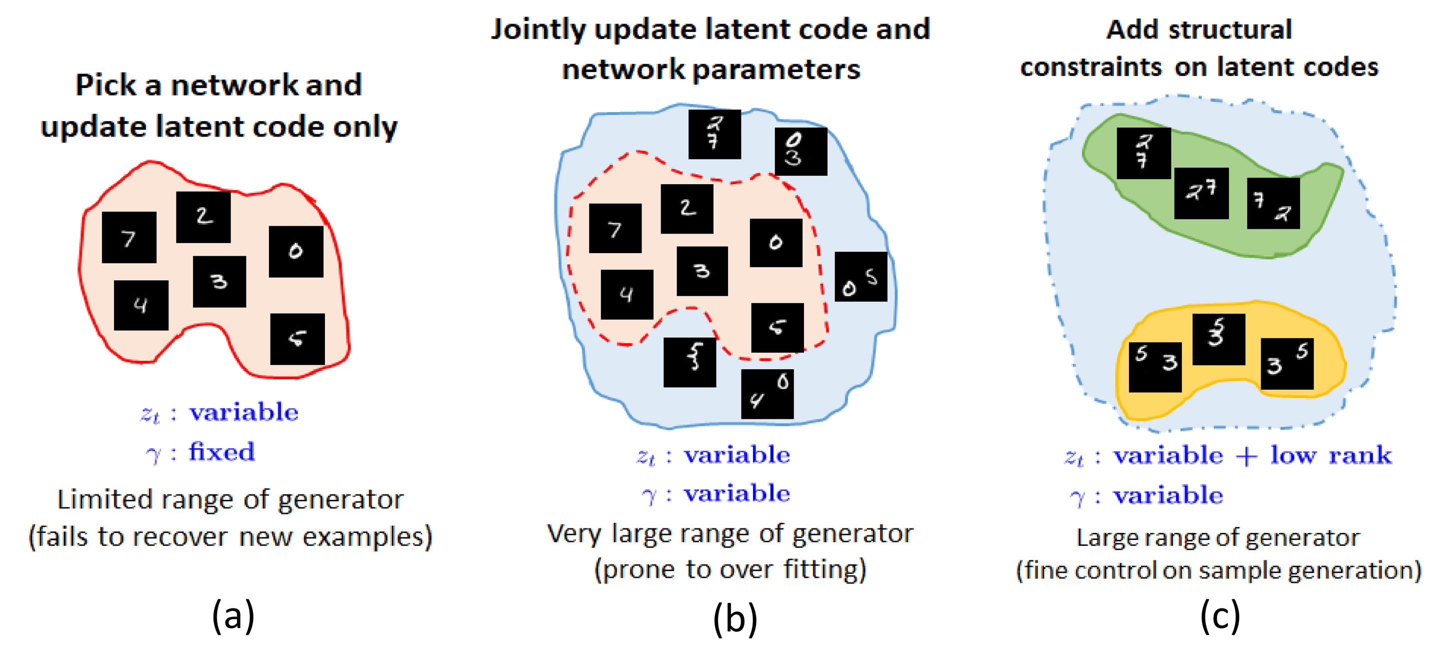

Figure 2. An illustration of different generative priors (a) Optimizing $\mathbf{z_t}$ of a trained generator. (b) Jointly optimizing $\mathbf{z_t}$ and γ enables recovery of a larger range of images. (c) Joint Optimization + Low-rank constraint potentially explain other structures in data.

Joint Optimization with Low-rank Constraint

- As the generator is usually a continuous function, joint optimization will allow latent codes to reflect the visual similarity of the video frames.
- We can further impose low-rank constraint on latent codes to represent the latent codes corresponding to the video sequence more concisely.

$$\min_{Z;\gamma} \sum_{t=1}^{T} ||\mathbf{y_t} - \mathbf{A_t} G_\gamma(\mathbf{z_t})||_2{}^2 \quad s.t. \quad rank(\mathbf{Z}) = r, \mathbf{Z} = [\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_T}]$$

Algorithm pseudocode: *Generative model for low-rank representation and reconstruction of videos*

Input: Measurements y_t, measurement matrices A_t, A generator structure $G_\gamma(\cdot)$
Initialize the latent codes z_t and generator weights γ randomly.
repeat
  Compute gradients w.r.t. z_t via backpropagation.
  Update latent code matrix $Z = [z_1 \ \cdots \ z_T]$.
  Threshold Z to a rank-r matrix via SVD or PCA.

  Compute gradients w.r.t. γ via backpropagation.
  Update network weights γ.
until convergence or maximum epochs
Output: Latent codes: $z_1, \ldots, z_T$ and network weights: γ

Experimental Setup

- **Datasets:** Different video sequences from KTH dataset resized to 64×64 and UCF101 dataset resized to 256×256.
- **Latent code dimension:** k = 256 for 64×64 and k = 512 for 256×256 video sequences. **Rank=4** as low-rank.
- **Optimizer:** Gradient descent for latent code update, Adam for network parameter update.
- **Generator:** We used generator architecture from DCGAN [5].

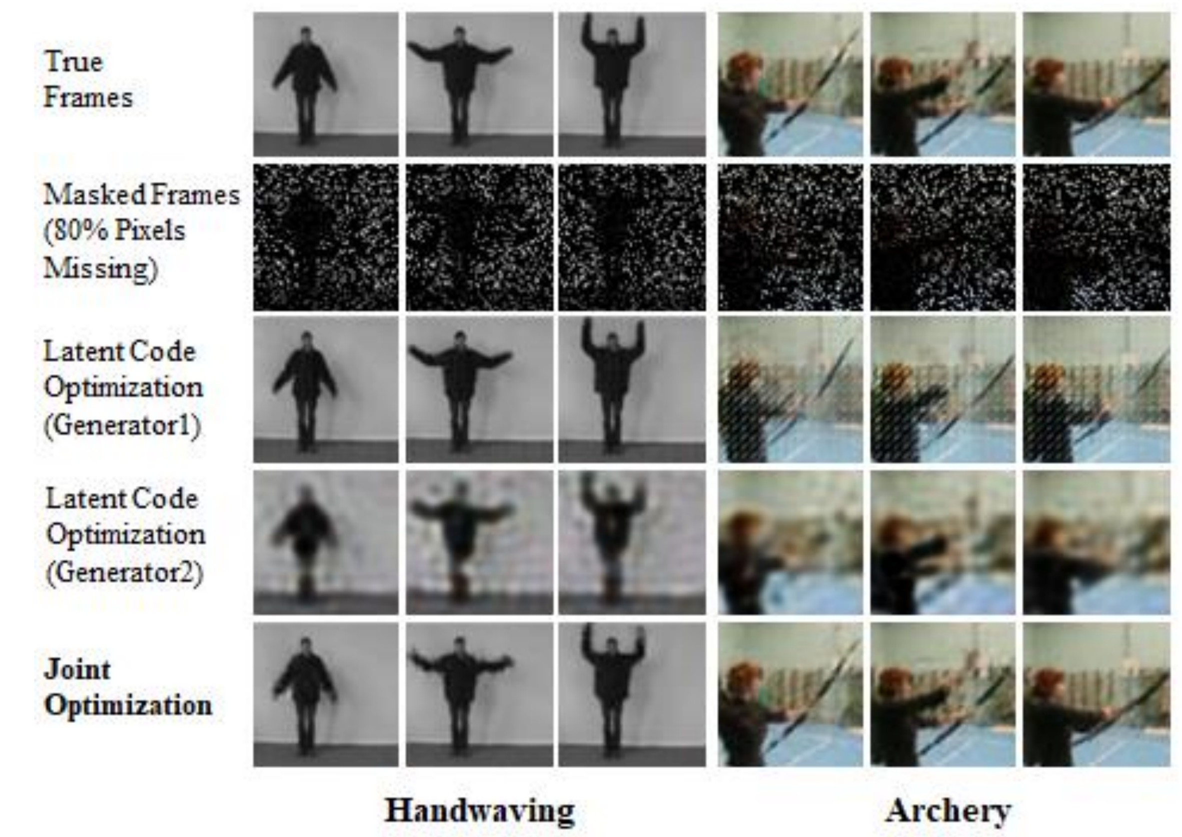

Figure 3. Joint optimization (untrained generator) vs latent code optimization (trained generators: Generator1 and Generator2). Generator1 is trained on the same dataset as the test set, Generator2 is trained on CIFAR10. Frame size is 64×64.

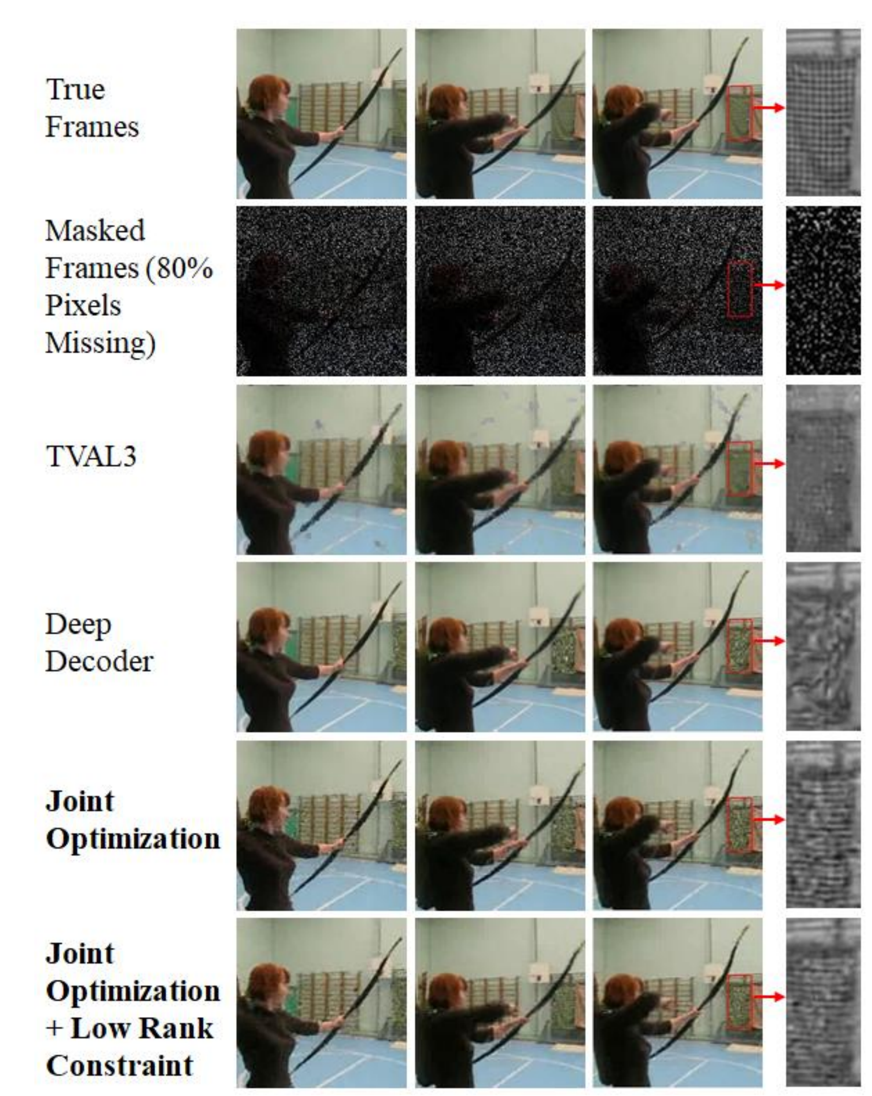

Figure 4. Reconstruction (Archery) for different algorithms (joint opt., joint opt +low rank., TVAL3 [2] and deep decoder [3]) for inpainting problem. Frame size is 256×256.

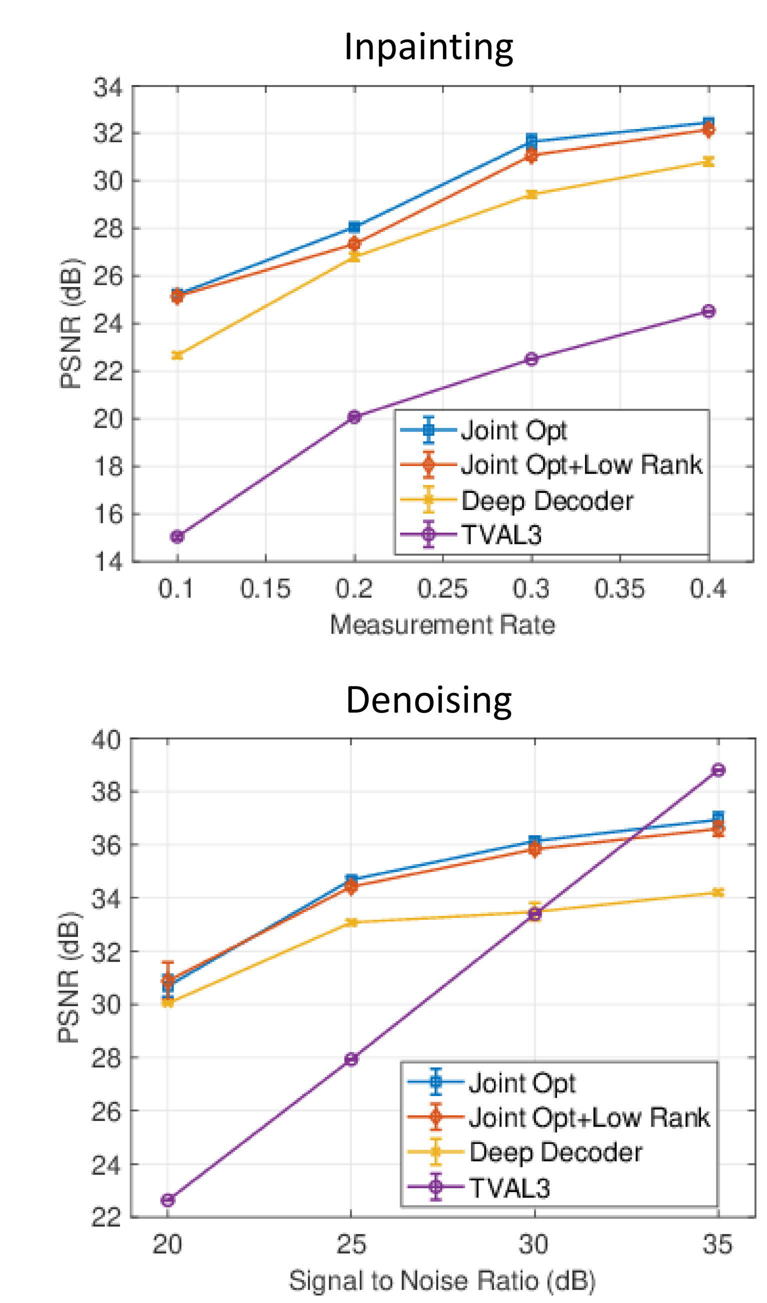

Figure 5. Reconstruction performance comparison for different algorithms for Handwaving sequence.