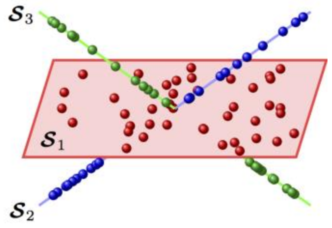
A Comparison of Classical and Deep Learning-based Techniques for Compressing Signals in a Union of Subspaces

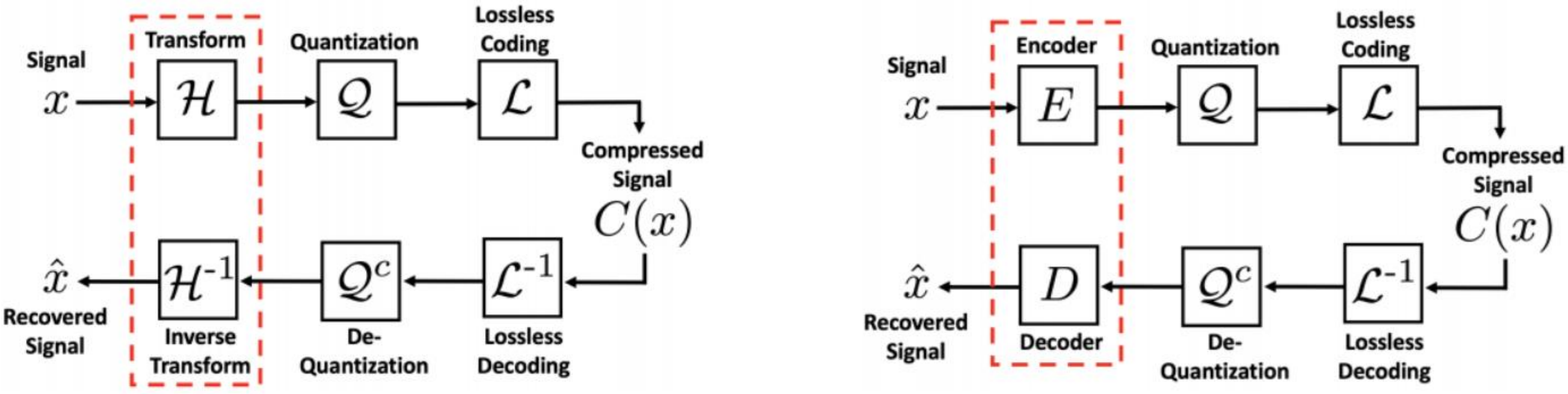
Sriram Ravula
The University of Texas at Austin

Swayambhoo Jain
Interdigital Inc.



- **Our Scenario:** Data lie in a union of subspaces
  - $\mathbf{D^*} \in \mathbb{R}^{\mathbf{d \times r}}, (\mathbf{r} \geq \mathbf{d})$ - overcomplete basis
  - $\mathbf{z} \in \mathbb{R}^{\mathbf{r}}$ - selection vector with s < d nonzero components
  - $\mathbf{x} = \mathbf{D^* z}$ - **signal we wish to compress**
- **Question:** How can we exploit the union-of-subspace structure to compress the signal while maintaining high level of fidelity?
- Traditional compression pipelines transform data into basis that is more useful for low-dimensional representations
  - e.g. block-wise DCT for images in JPEG
  - combine with quantization and lossless coding

- **Idea:** exploit learned models to find best low-dimensional representation of the data
    - replace transform and corresponding inverse transform with classical or deep learning-based encoder and decoder



(a) traditional compression pipeline

(b) our proposed compression pipeline

- Test three encoder-decoder combinations:
  - deep autoencoder (AE)
  - compressed sensing with a learned dictionary (CSLD)
  - compressed sensing with generative models (CSGM)
- Each one consists of an encoder that reduces the dimension of the signal and a decoder that recovers the signal

- **Deep Autoencoder:** jointly optimize encoder and decoder networks to reconstruct signal
- Training procedure:

$$\theta_E^*, \theta_D^* = \arg\min_{\theta_E, \theta_D} \sum_{i=1}^{N} \|\mathbf{x}_i - D_{\theta_D}(E_{\theta_E}(\mathbf{x}_i))\|_2^2.$$

- Encoder: $E_{\theta_E}(\mathbf{x})$
- Decoder: $D_{\theta_D}(\mathbf{y})$

- **Compressed Sensing with a Learned Dictionary**: learn a dictionary to represent training data, then reconstruct a signal from random Gaussian projections using the dictionary
- Training Procedure:

$$\hat{\mathbf{D}}, \mathbf{S} = \arg\min_{\mathbf{D},\mathbf{S}} \|\mathbf{X} - \mathbf{D}\mathbf{S}\|_F^2 + \alpha||\mathbf{S}||_1$$

- Encoder: $E(\mathbf{x}) = \mathbf{A}\mathbf{x}$

$$\mathbf{A}_{i,j} \sim \mathcal{N}(0, \tfrac{1}{m})$$

- Decoder: $D(\mathbf{y}) = \hat{\mathbf{D}}\hat{\mathbf{v}}$

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} ||\mathbf{y} - \mathbf{A}\hat{\mathbf{D}}\mathbf{v}||_2^2 + \lambda||\mathbf{v}||_1$$

- **Compressed Sensing with a Generative Model:** train a generative adversarial network (GAN) on training data, then reconstruct a signal from random Gaussian projections using the range of the generator
- Training Procedure:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

- Encoder: $E(\mathbf{x}) = \mathbf{A}\mathbf{x}$

$$\mathbf{A}_{i,j} \sim \mathcal{N}(0, \tfrac{1}{m})$$

- Decoder: $D(\mathbf{y}) = G_\phi(\mathbf{z}^*)$

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} ||\mathbf{y} - \mathbf{A}G_\phi(\mathbf{z})||_2^2 + \beta||\mathbf{z}||_2^2$$

- Comparison of the methods:

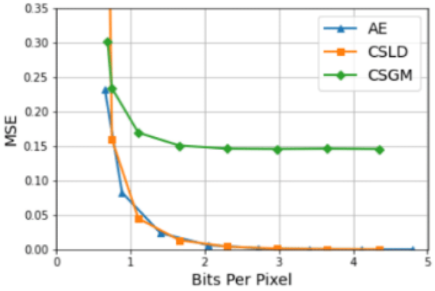
Method	Nonlinear		Learned		Iterative		Memory Use	
	Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec
AE	True	True	True	True	False	False	High	High
CSLD	False	False	False	True	False	True	Low	Low
CSGM	False	True	False	True	False	True	Low	High

- **Data:**
    - 100k train, 30k val, 20k test points generated from union of subspaces
    - MNIST handwritten images
- **Implementation Details:**
    - Input $\mathbf{x}$ and Output $\tilde{\mathbf{x}}$ for pipeline:

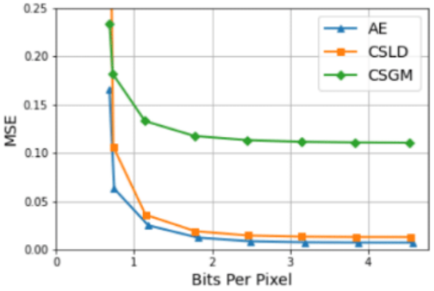
$$\mathbf{x} \to \mathbf{y} = \mathrm{Encoder}(\mathbf{x}) \to \hat{\mathbf{y}} = \mathrm{Quant}(\mathbf{y}) \to$$
$$\tilde{\mathbf{y}} = \mathrm{DeQuant}(\hat{\mathbf{y}}) \to \tilde{\mathbf{x}} = \mathrm{Decoder}(\tilde{\mathbf{y}})$$

    - uniform quantization with 1,2,...,8 bits
    - compare reconstruction quality (mean squared error) vs compression level (bits per pixel)

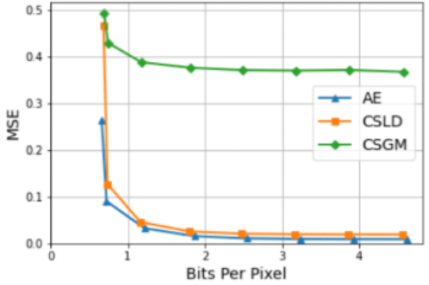
Results - Synthetic Data



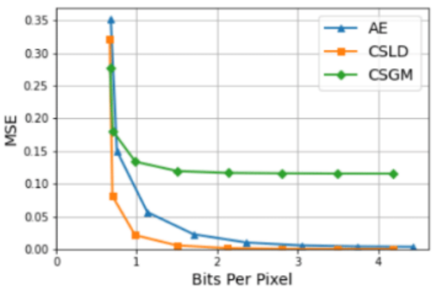
(a) $d = 20, r = 100, s = 1$



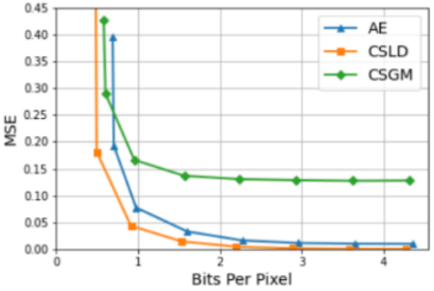
(b) $d = 20, r = 100, s = 5$



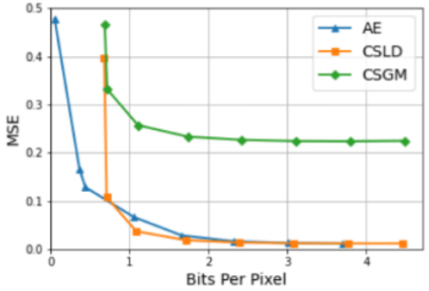
(c) $d = 20, r = 100, s = 10$



(d) $d = 100, r = 1000, s = 1$

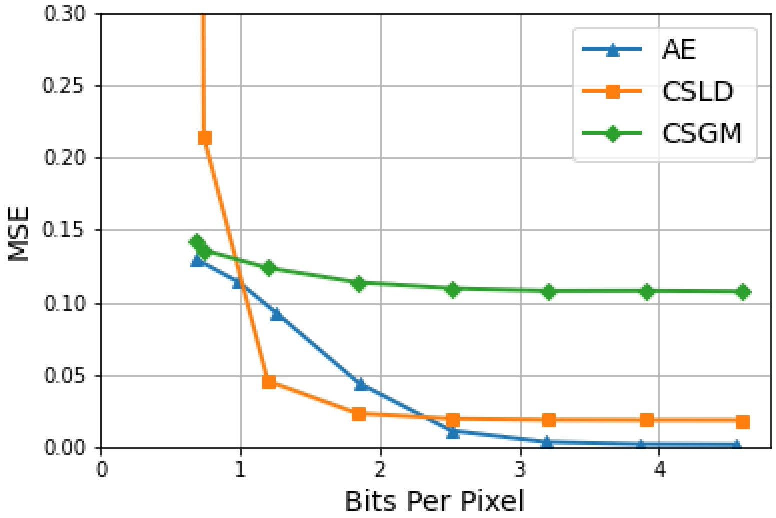


(e) $d = 100, r = 1000, s = 5$



(f) $d = 100, r = 1000, s = 10$

Results - MNIST



- **Conclusions:**
  - AE and CSLD both good for compressing signals from union-of-subspaces
    - Dictionary-based methods have advantage of solid theoretical foundation
    - AE performs better on data with less sparsity - deep learning succeeds where sparsity priors cannot
  - CSGM performs poorly - quantization effects?