One Network to Solve Them All

Solving Linear Inverse Problems using Deep Projection Models

Image Processing Problems

- Image inpainting

input

reconstruction
output

- Super resolution

input

reconstruction
output

Linear Inverse Problems

- The goal is to reconstruct an image $\boldsymbol{x} \in \mathbf{R}^d$ from a set of measurements $\boldsymbol{y} \in \mathbf{R}^m$ of the form

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$$

  where $\boldsymbol{A} \in \boldsymbol{R}^{m \times d}$ is the measurement operator and $\boldsymbol{n} \in \boldsymbol{R}^m$ is the noise.

- For example;
  - Image inpainting ⟶ $\boldsymbol{A}$ is a pixelwise mask
  - Super resolution ⟶ $\boldsymbol{A}$ is a downsampling operation
- Linear inverse problems are generally underdetermined
  - Less equations than unkowns, i.e., $m < d$
  - $\boldsymbol{A}$ has a null-space ⟶ infinite number of solutions
- How to find the «true» solutions?

Solving Linear Inverse Problems

- Using hand-designed signal priors
  - Regularizing the problem using hand-designed signal priors in a penalty form

$$\min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\phi(\boldsymbol{x})$$

    where $\phi: \mathbf{R}^d \to \mathbf{R}$ is the signal prior and $\lambda \geq 0$ is the weighting term

  - Signal priors constraining sparsity is widely studied

$$\phi(\boldsymbol{x}) = \|\boldsymbol{W}\boldsymbol{x}\|_1$$

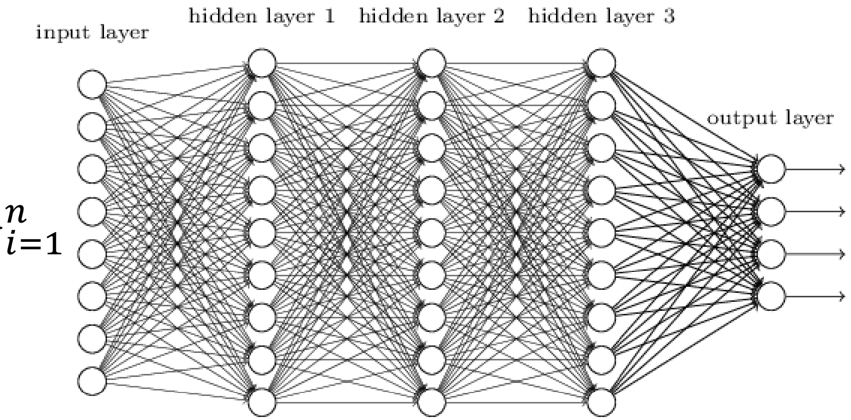
    where $\boldsymbol{W}$ is a linear operation that produces sparse features from input signal

  - For images, $\boldsymbol{W}$ is usually wavelet transformation
  - ℓ_1 norm is used
    - Forms a convex optimization problem, hence, global optimality is possible
    - Under certain assumptions, there exist many theoretical guarantees

Solving Linear Inverse Problems

- Using deep neural networks
  - Given a linear operator $\boldsymbol{A}$ and a dataset $\mathcal{M} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ the pairs $\{(\boldsymbol{x}_i, \boldsymbol{A}\boldsymbol{x}_i)\}_{i=1}^{n}$ can be used to learn an inverse mapping $f \approx \boldsymbol{A}^{-1}$ by minimazing the distance between $\boldsymbol{x}_i$ and $f(\boldsymbol{A}\boldsymbol{x}_i)$, even when $\boldsymbol{A}$ is underdetermined

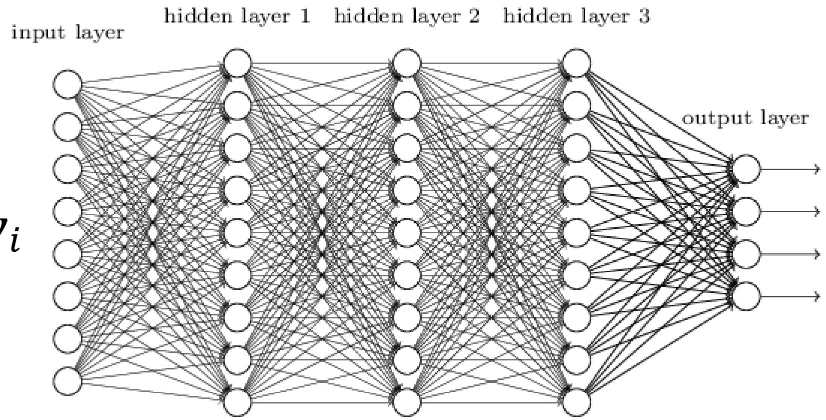
$\{\boldsymbol{A}\boldsymbol{x}_i\}_{i=1}^{n}$



$\{\boldsymbol{x}_i\}_{i=1}^{n}$

Training

$\boldsymbol{y}_i$



$\widehat{\boldsymbol{x}}_i$

Reconstruction

Problems with these solution methods

- Using hand-designed signal priors
  - Priors are usually too generic to recover the signal of interest
  - One can easily generate noise signals that have sparse wavelet coefficients
- Using deep neural networks
  - End-to-end mapping
  - Application specific
  - Even if the problems change slightly, the mapping functions (neural nets) need to be retrained
- One strategy is to use deep neural networks to learn the signal priors from the given dataset

Proposed Solution

- Based on alternating direction method of multipliers (ADMM)
  - ADMM typically separates a complicated objective into several simpler ones by variable splitting, i.e.,

$$\min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_2^2 + \lambda\phi(\boldsymbol{x})$$
$$\text{s.t.} \;\; \boldsymbol{x} = \boldsymbol{z}$$

  which is equivalent to the original problem
  - The scaled form of the augmented Lagrangian of this problem

$$L(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}) = \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_2^2 + \lambda\phi(\boldsymbol{x}) + \frac{\rho}{2} \|\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{u}\|_2^2$$

  where $\rho > 0$ is the penalty parameter of the constraint $\boldsymbol{x} = \boldsymbol{z}$ and $\boldsymbol{u} = \frac{\boldsymbol{y}}{\rho}$

ADMM Updates

- Alternatively optimizing $L(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u})$ over $\boldsymbol{x}, \boldsymbol{z}$ and $\boldsymbol{u}$, ADMM algorithm yields

$$\mathbf{x}^{(k+1)} \leftarrow \arg\min_{\mathbf{x}} \frac{\rho}{2}\left\|\mathbf{x} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}\right\|_2^2 + \lambda\,\phi(\mathbf{x})$$

$$\mathbf{z}^{(k+1)} \leftarrow \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{y} - A\mathbf{z}\|_2^2 + \frac{\rho}{2}\left\|\mathbf{x}^{(k+1)} - \mathbf{z} + \mathbf{u}^{(k)}\right\|_2^2$$

$$\mathbf{u}^{(k+1)} \leftarrow \mathbf{u}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}.$$

- Update of $\boldsymbol{z}$ is a least squares problem and can be solved efficiently
- Update of $\boldsymbol{x}$ is in the form of a proximal operator of signal prior $\phi(\boldsymbol{x})$ with penalty ρ/λ, denoted as $\mathbf{prox}_{\phi,\frac{\rho}{\lambda}}(\boldsymbol{v})$, where $\boldsymbol{v} = \boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}$
- The signal prior $\phi(\boldsymbol{x})$ and the linear operator $\boldsymbol{A}$ is seperated
- This separation enables the learning of signal priors via deep neural networks

Learning a Proximal Operator

$$\boldsymbol{x}^{(k+1)} \leftarrow \mathbf{prox}_{\phi,\frac{\rho}{\lambda}}(\boldsymbol{v})$$

- The signal prior $\phi(\boldsymbol{x})$ only appears in the form of a proximal operator
- No need to explicitly learn the signal prior and solve the proximal operator for updating $\boldsymbol{x}$
- Directly learn the proximal operator $\mathcal{P}$ such that

$$\boldsymbol{x}^{(k+1)} \leftarrow \mathcal{P}(\boldsymbol{v}) = \mathcal{P}\left(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}\right)$$

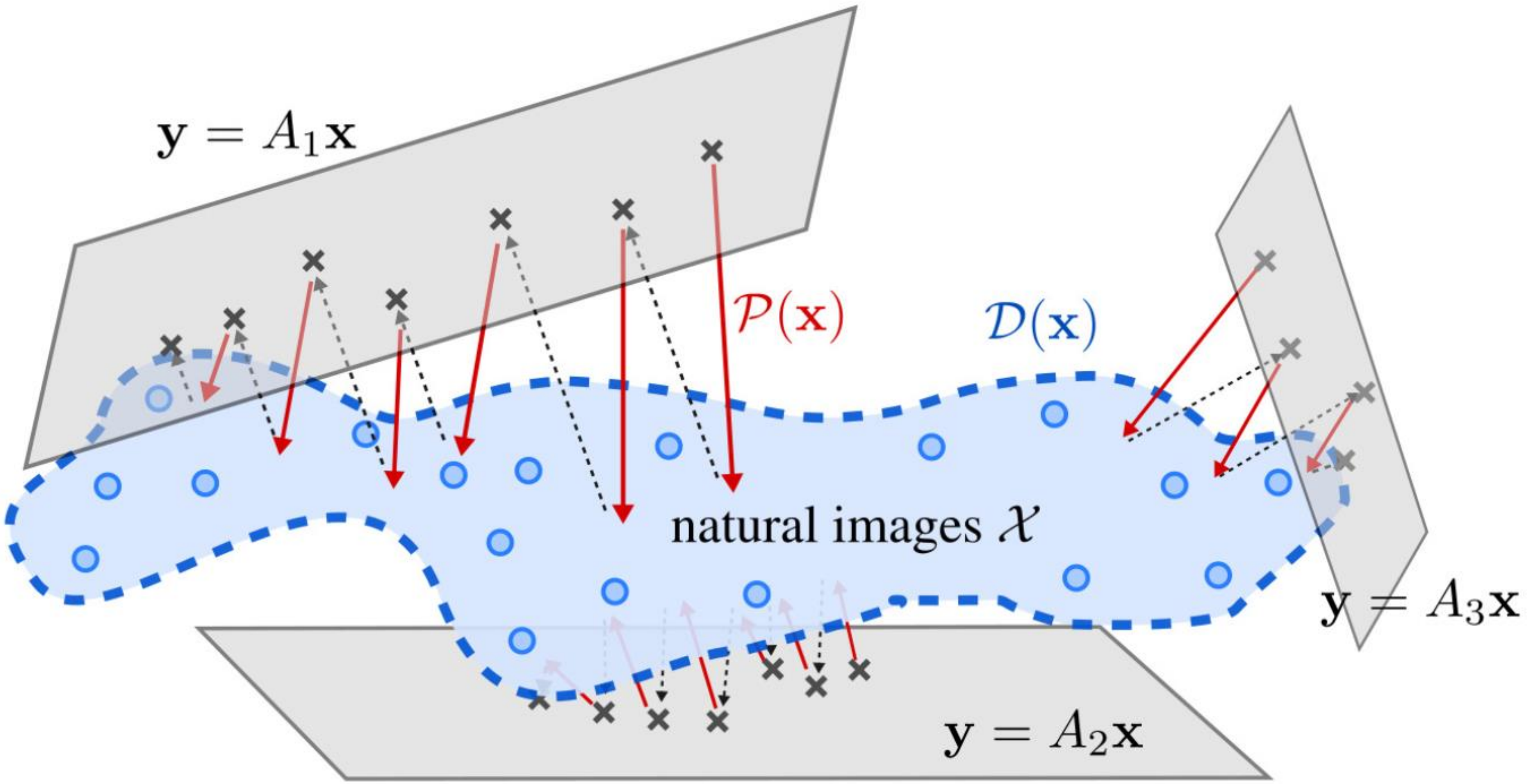
How to learn $\mathcal{P}$

- Let $\mathcal{X}$ represent the set of all natural images, i.e., solution space
- The best signal prior is the indicator function of $\mathcal{X}$, denoted by $I_{\mathcal{X}}$
- The corresponding proximal operator is

$$\mathbf{prox}_{I_{\mathcal{X}},\rho}\left(\boldsymbol{v}\right)$$

- However, we do not have $I_{\mathcal{X}}$ in practice, hence, cannot evaluate $\mathbf{prox}_{I_{\mathcal{X}},\rho}\left(\boldsymbol{v}\right)$
- Thus, train a classifier $\mathcal{D}$ to learn $I_{\mathcal{X}}$
- Based on the learned classifier $\mathcal{D}$, learn a projection function $\mathcal{P}$ that maps a signal $\boldsymbol{v}$ to the set defined by the classifier

Illustration of classifier $\mathcal{D}$ and projection $\mathcal{P}$



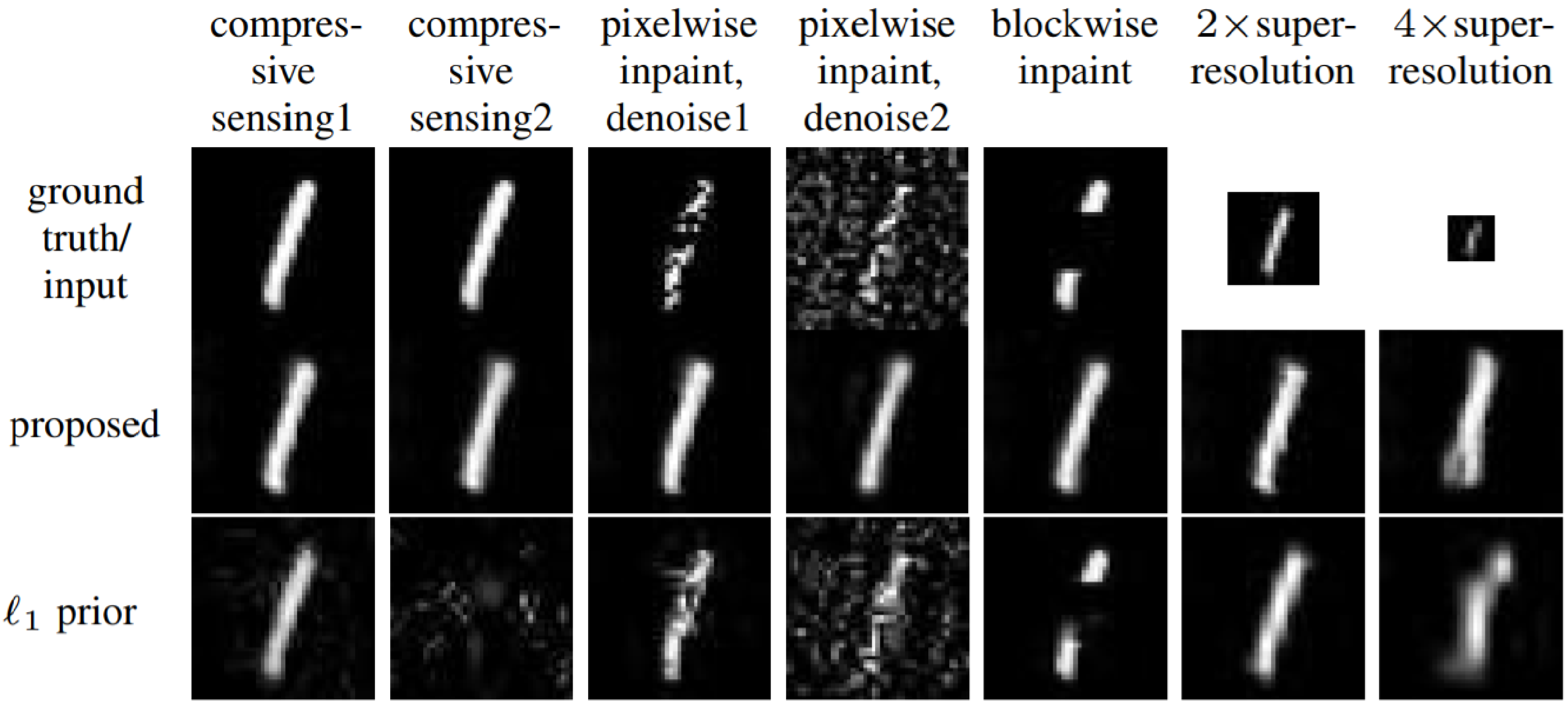
Training the networks

- Adversarial learning
  - $\mathcal{P}$ is the generative network
  - $\mathcal{D}$ is the discriminative network
  - The projector network $\mathcal{P}$ is trained to minimize

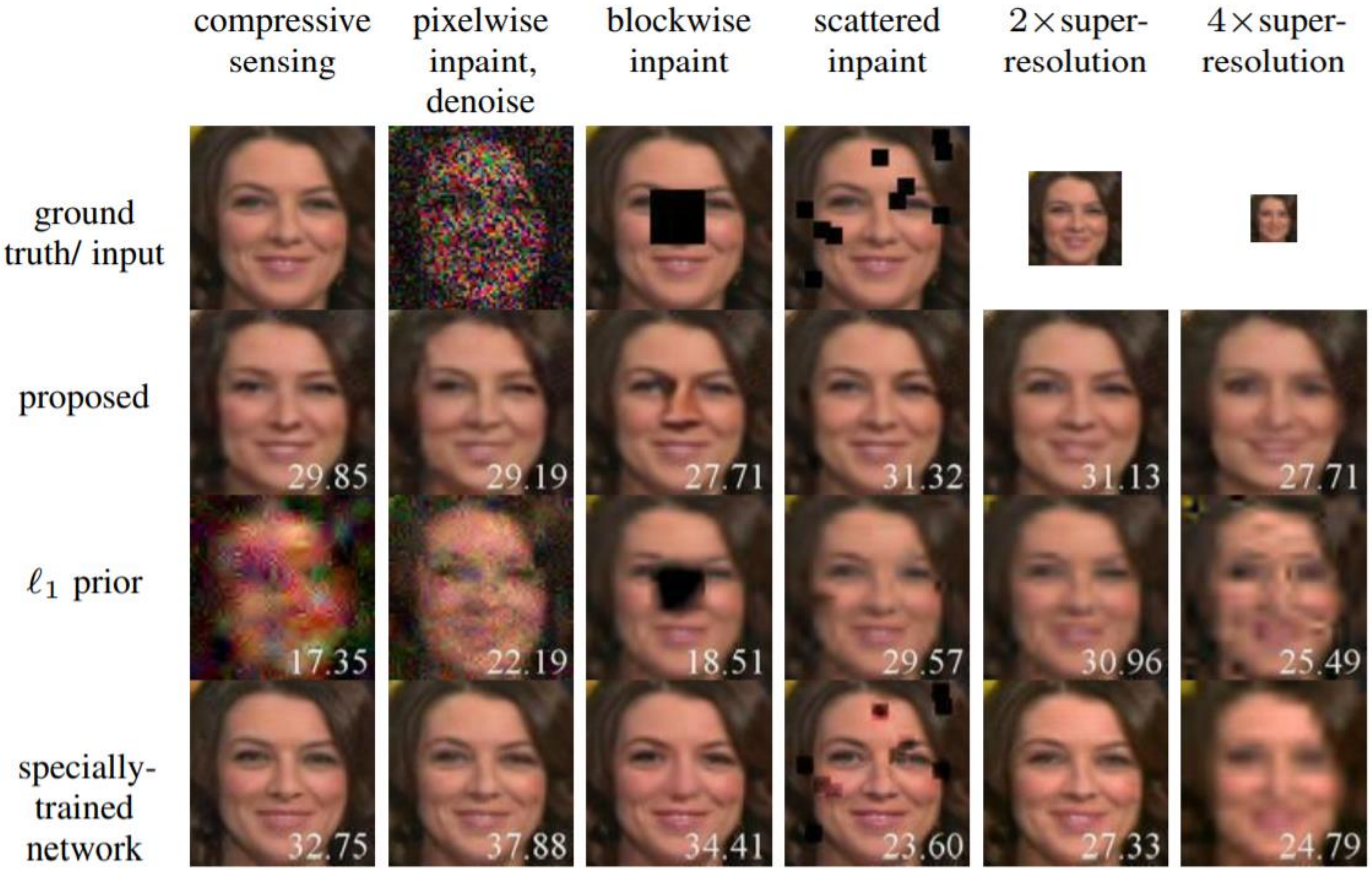
$$\arg\min_{\mathbf{x}} \frac{\rho}{2}\left\|\mathbf{x}-\mathbf{z}^{(k)}+\mathbf{u}^{(k)}\right\|_2^2+\lambda\,\phi(\mathbf{x})$$

  - Thus, $\mathcal{P}$ aims to fail the classifier $\mathcal{D}$ by generating more natural like images
  - As the projector $\mathcal{P}$ improves, $\mathcal{D}$ is updated to tighten its decision boundary

Results


compres- sive sensing1
compres- sive sensing2
pixelwise inpaint, denoise1
pixelwise inpaint, denoise2
blockwise inpaint
2×super- resolution
4×super- resolution
ground truth/ input
proposed
ℓ_1 prior

Results


compressive sensing
pixelwise inpaint, denoise
blockwise inpaint
scattered inpaint
2× super-resolution
4× super-resolution
ground truth/ input
proposed
ℓ1 prior
specially-trained network
29.85
29.19
27.71
31.32
31.13
27.71
17.35
22.19
18.51
29.57
30.96
25.49
32.75
37.88
34.41
23.60
27.33
24.79

Robustness to variations on $\boldsymbol{A}$ and noise