Computational Imaging with Plug-and-play priors: Leverage the Power of Deep Learning

Xiaojian Xu
Supervised by Dr. Ulugbek Kamilov
Washington University in St. Louis

ISPG-Seminar – Dec 9 2020

Imaging is everywhere

Washington University in St.Louis

imaging in microscopy

imaging in photography

imaging in astronomy

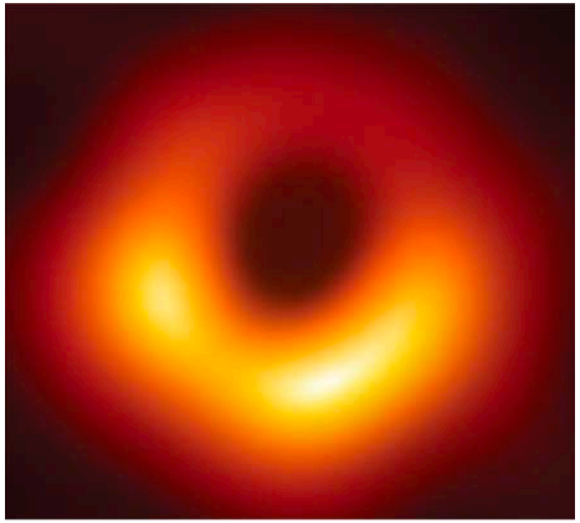
Big shift in imaging

- Imaging not by taking pictures but by computing pictures

Computational imaging applications

- In computational imaging, we don't have direct access to the thing we want

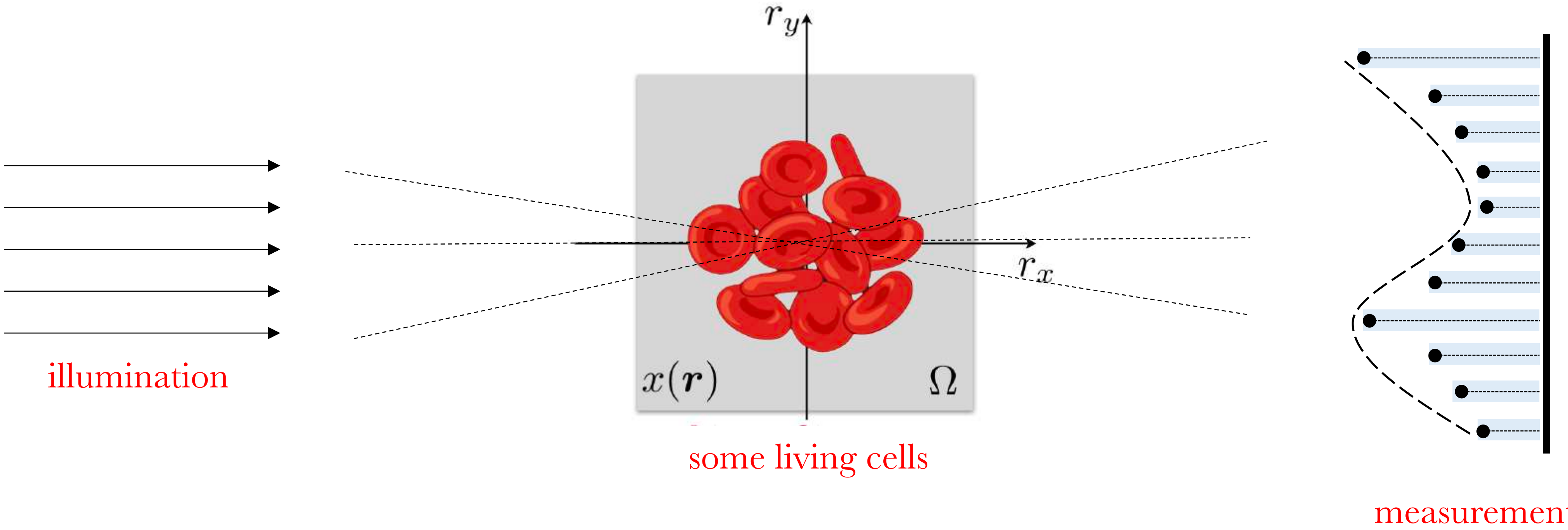
imaging skeleton



imaging blackhole

imaging infant

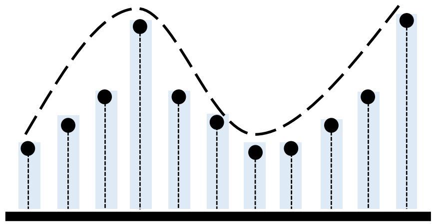
A simple microscopy imaging problem (Acquisition)



soure: [Ulugbek Kamilov]

A simple microscopy imaging problem (Reconstruction)

Washington University in St. Louis



measurements

Computational Imaging

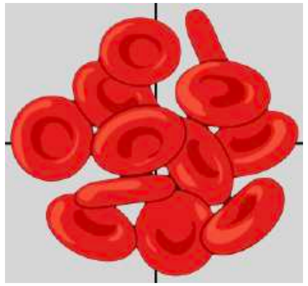
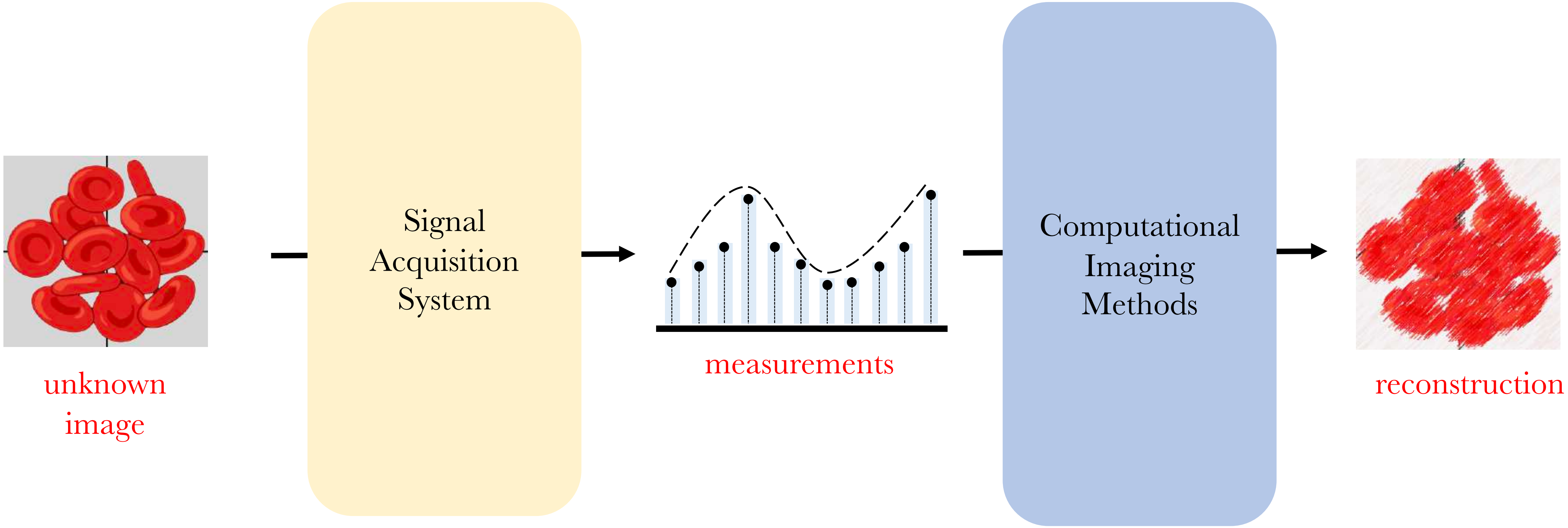


image of cells

Computational Imaging is the process of **indirectly forming images** from **measurements** using algorithms that rely on a significant amount of computing.

soure: https://en.wikipedia.org/wiki/Computational_imaging

Imaging needs two procedures



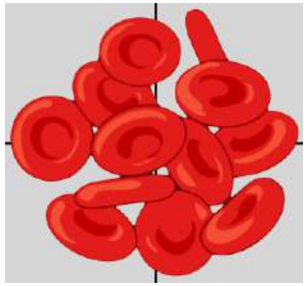
Imaging as an inverse problem

Washington University in St. Louis

Signal Acquisition Procedure

Forward problem: generate $\boldsymbol{y}$ from $\boldsymbol{x}$

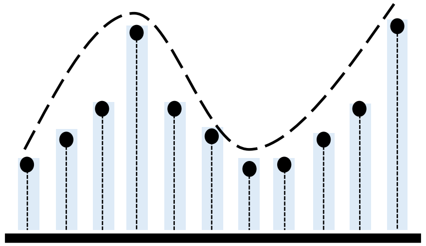
unknown (target)	known	unknown	known
$\boldsymbol{x}$	$\mathbf{H}(\cdot)$	$\boldsymbol{e}$	$\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$



unknown image

forward model

noise



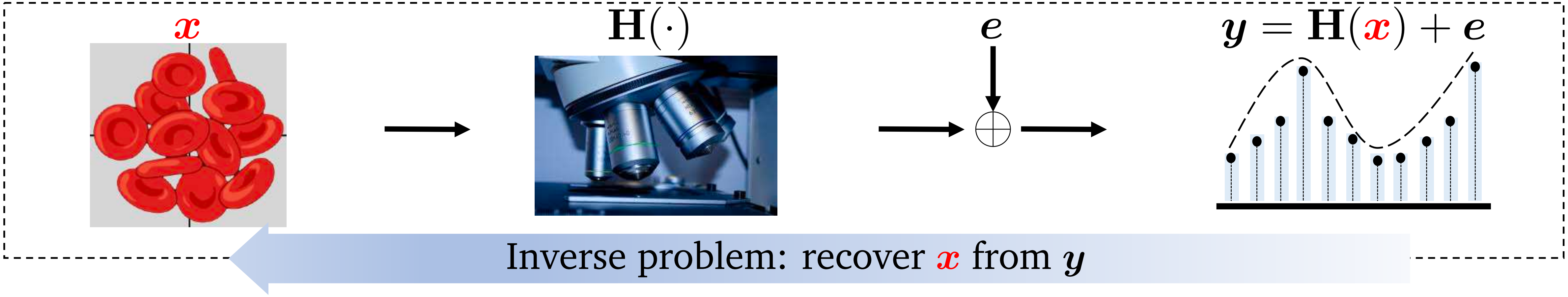
measurements

Inverse problem: recover $\boldsymbol{x}$ from $\boldsymbol{y}$

Image Reconstruction Procedure

Imaging inverse problems are challenging

Washington University in St. Louis



What makes imaging inverse problems challenging?

- Solution is not unique
- Measurements are noisy
- Image x can be high-dimensional

Computation imaging methods

Washington University in St. Louis

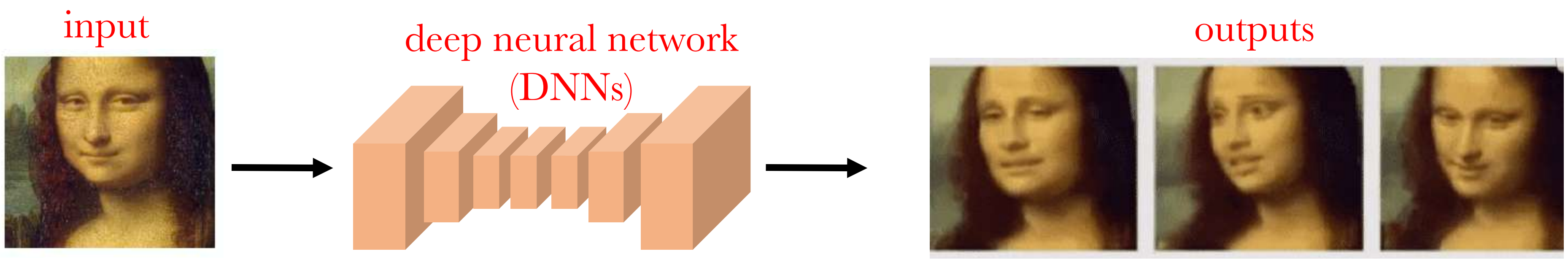
- Formulate it as a **regularized optimization** task (model-based optimization)

data-fidelity, prior/regularizer

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{ g(\boldsymbol{x}) + h(\boldsymbol{x}) \}$$

example: $$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{ \frac{1}{2} \| \mathbf{H}(\boldsymbol{x}) - \boldsymbol{y} \|^2 + h(\boldsymbol{x}) \}$$

- Learn an **end-to-end mapping**



soure: [Zakharov'19]

- Combining **optimization & learning**!

Unrolling | RED | PnP | …

Computation imaging methods

Washington University in St. Louis

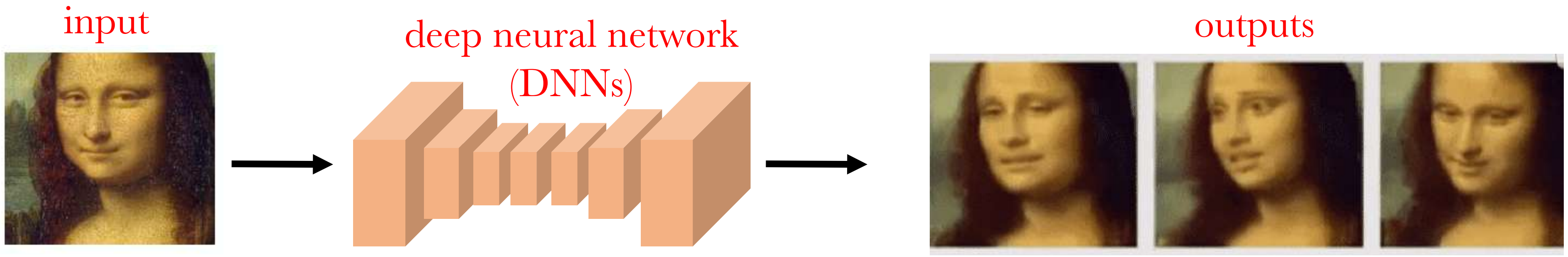
- Formulate it as a **regularized optimization** task (model-based optimization)

data-fidelity prior/regularizer

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{ g(\boldsymbol{x}) + h(\boldsymbol{x}) \}$$

example: $$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{ \frac{1}{2} \|\mathbf{H}(\boldsymbol{x}) - \boldsymbol{y}\|^2 + h(\boldsymbol{x}) \}$$

- Learn an **end-to-end mapping**



soure: [Zakharov'19]

- Combining **optimization & learning**!

Unrolling | RED | PnP | …

Today we will talk about

Washington University in St. Louis

1. Introduction to Plug-and-Play Priors (PnP)

- Denoiser strength selecting challenge
- Optimization interpretation and convergence analysis challenge

2. Denoiser scaling technique

- [Xu'20(1)] X. Xu et.al. Boosting the Performance of Plug-and-Play Priors via Denoiser Scaling

3. Optimization interpretation and convergence analysis of PnP with MMSE denoisers

- [Xu'20(2)] X. Xu et.al. Provable Convergence of Plug-and-Play Priors With MMSE Denoisers,

Imaging as a regularized optimization task

- Recall our regularized optimization task

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\arg\min}\{\underbrace{g(\boldsymbol{x})}_{\text{data-fidelity}} + \underbrace{h(\boldsymbol{x})}_{\text{prior/regularizer}}\}$$

Example: Fast iterative shrinkage/thresholding algorithm (FISTA) [Nesterov'13] & Alternating direction method of multipliers (ADMM) [Boyd'10]

FISTA

$$\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma\nabla g(\boldsymbol{s}^{k-1})$$
$$\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma h}(\boldsymbol{z}^k)$$
$$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

ADMM

$$\boldsymbol{z}^k \leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$$
$$\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma h}(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$$
$$\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$$

Proximal algorithms

Washington University in St. Louis

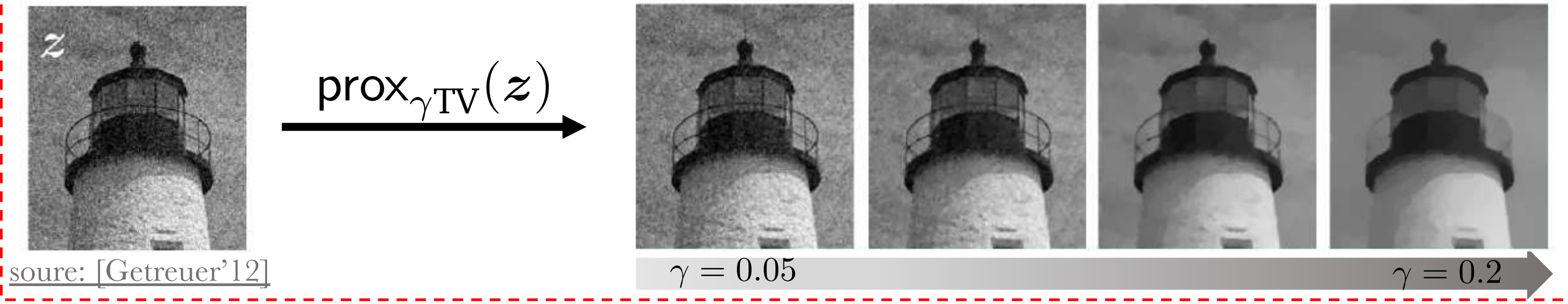
model prior

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{ g(\boldsymbol{x}) + h(\boldsymbol{x}) \}$$

- Let's take a closer look at these two proximal algorithms

FISTA		ADMM
$\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$	increase data consistency	$\boldsymbol{z}^k \leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$
$\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma h}(\boldsymbol{z}^k)$	reduce noise	$\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma h}(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$
$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$		$\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$

Definition: $\mathsf{prox}_{\gamma h}(\boldsymbol{z}) := \arg\min_{\boldsymbol{x}} \{ \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \gamma h(\boldsymbol{x}) \}$ image denoiser for AWGN



soure: [Getreuer'12]

Proximal algorithms

Washington University in St. Louis

model prior
$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{g(\boldsymbol{x}) + h(\boldsymbol{x})\}$

- Let's take a closer look at these two proximal algorithms

FISTA		ADMM
$\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$	increase data consistency	$\boldsymbol{z}^k \leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$
$\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma h}(\boldsymbol{z}^k)$	reduce noise	$\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma h}(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$
$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$		$\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$

Plug and Play Prior (PnP) [Venkat'13]:
simply replace the proximal map with other denoisers D_σ!

$$\mathsf{prox}_{\gamma h} \Rightarrow \mathsf{D}_\sigma$$

where $\sigma \geq 0$ refers to denoising strength.

PnP: Incorporating a denoiser in the optimization

Washington University in St. Louis

- Plug-and-Play (PnP) embraces off-the-shelf image denoisers

PnP-FISTA

$$\boldsymbol{z}^k = \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$$

$$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k)$$

$$\boldsymbol{s}^k = \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

any off-the-shelf image denoiser

PnP-ADMM

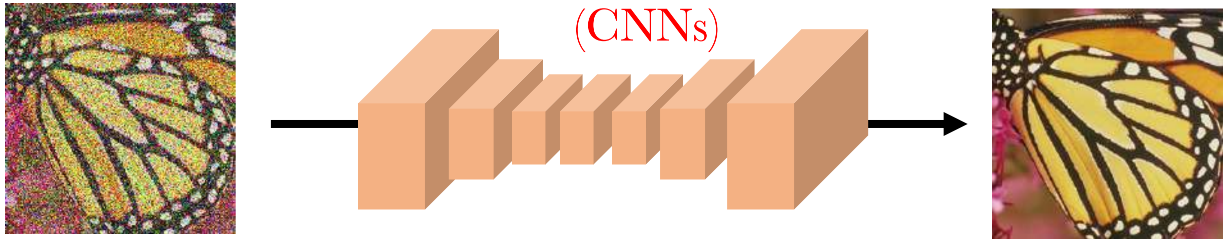
$$\boldsymbol{z}^k = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$$

$$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$$

$$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$$

Example: D_σ could be a neural network

Convolutional Neural Networks (CNNs)



PnP: Incorporating a denoiser in the optimization

Washington University in St. Louis

- Plug-and-Play (PnP) embraces off-the-shelf image denoisers

PnP-FISTA		PnP-ADMM
$\boldsymbol{z}^k = \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$		$\boldsymbol{z}^k = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$
$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k)$	$\sigma = ?$	$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$
$\boldsymbol{s}^k = \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$		$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$

However, …

- Many CNNs denoisers do not have a tunable parameter for the noise standard deviation!

PnP: Incorporating a denoiser in the optimization

Washington University in St. Louis

- Plug-and-Play (PnP) embraces off-the-shelf image denoisers

PnP-FISTA

$$\boldsymbol{z}^k = \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$$
$$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k)$$
$$\boldsymbol{s}^k = \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

$\sigma = ?$

PnP-ADMM

$$\boldsymbol{z}^k = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$$
$$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$$
$$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$$

- Previous solution : denoiser selection
  - ✪ Idea: Training multiple CNN instances and select the one that works best.
  - ✪ Issues: Requires training multiple CNN instances and leads to suboptimal performance.

Proposed denoiser scaling technique

Washington University in St. Louis

- Plug-and-Play (PnP) embraces off-the-shelf image denoisers

PnP-FISTA		PnP-ADMM
$\boldsymbol{z}^k = \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$		$\boldsymbol{z}^k = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$
$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k)$	$\sigma = ?$	$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k + \boldsymbol{s}^{k-1})$
$\boldsymbol{s}^k = \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$		$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$

- **Our proposal** [Xu'20(1)] **: denoiser scaling**
  - ✪ Introduce a tunable parameter μ to adjust the denoising strength of a pre-trained CNN.

Without scaling: $\widehat{\boldsymbol{z}} = \mathsf{D}_\sigma(\boldsymbol{z})$

Denoiser scaling: $\widehat{\boldsymbol{z}} = \mu^{-1}\mathsf{D}_\sigma(\mu \boldsymbol{z}), \quad \mu > 0$

Performance of denoiser scaling

- CNN trained on noise level $\sigma = 20$, applied on noise level $\sigma = 30$, difference $\Delta_\sigma = 10$.

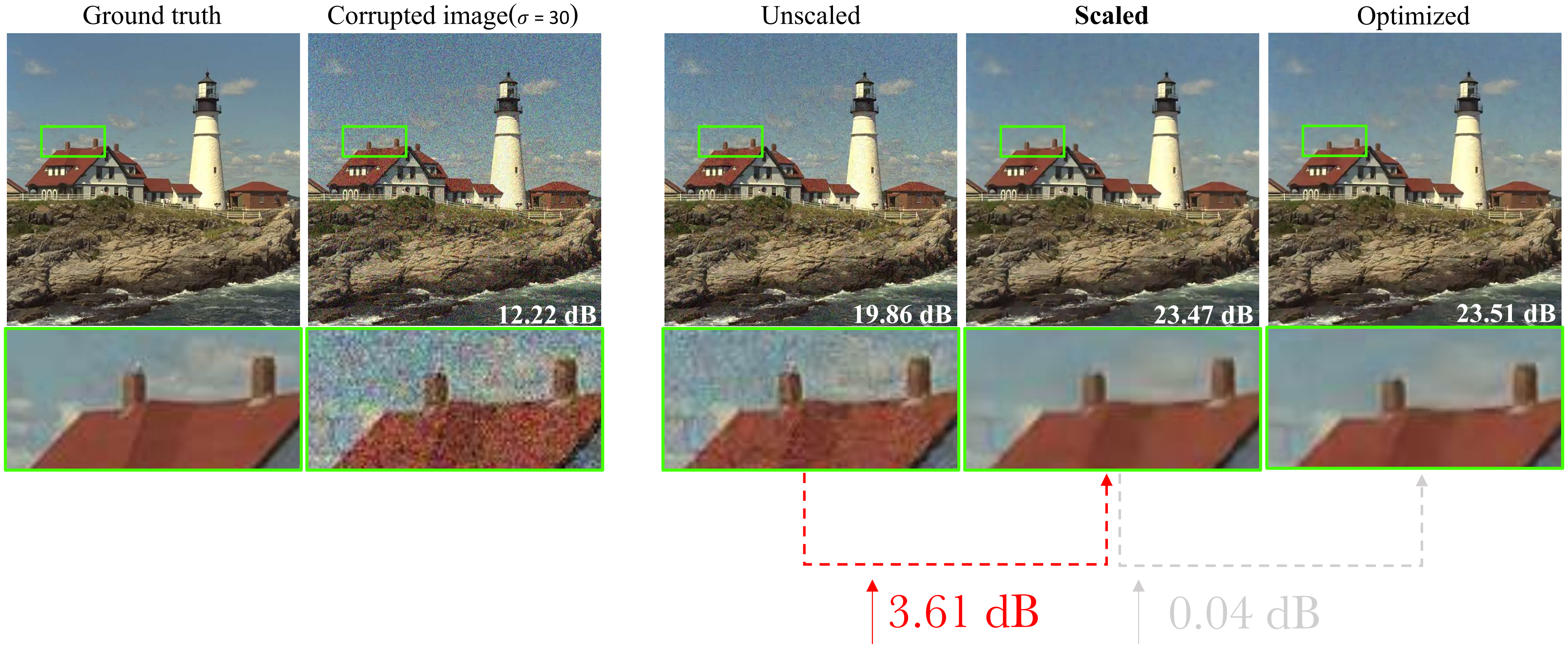
Noisy image:
$\boldsymbol{z}$

Without scaling:
$\widehat{\boldsymbol{z}} = \mathsf{D}_\sigma(\boldsymbol{z})$

With scaling:
$\widehat{\boldsymbol{z}} = \mu^{-1}\mathsf{D}_\sigma(\mu\boldsymbol{z})$

Performance of denoiser scaling

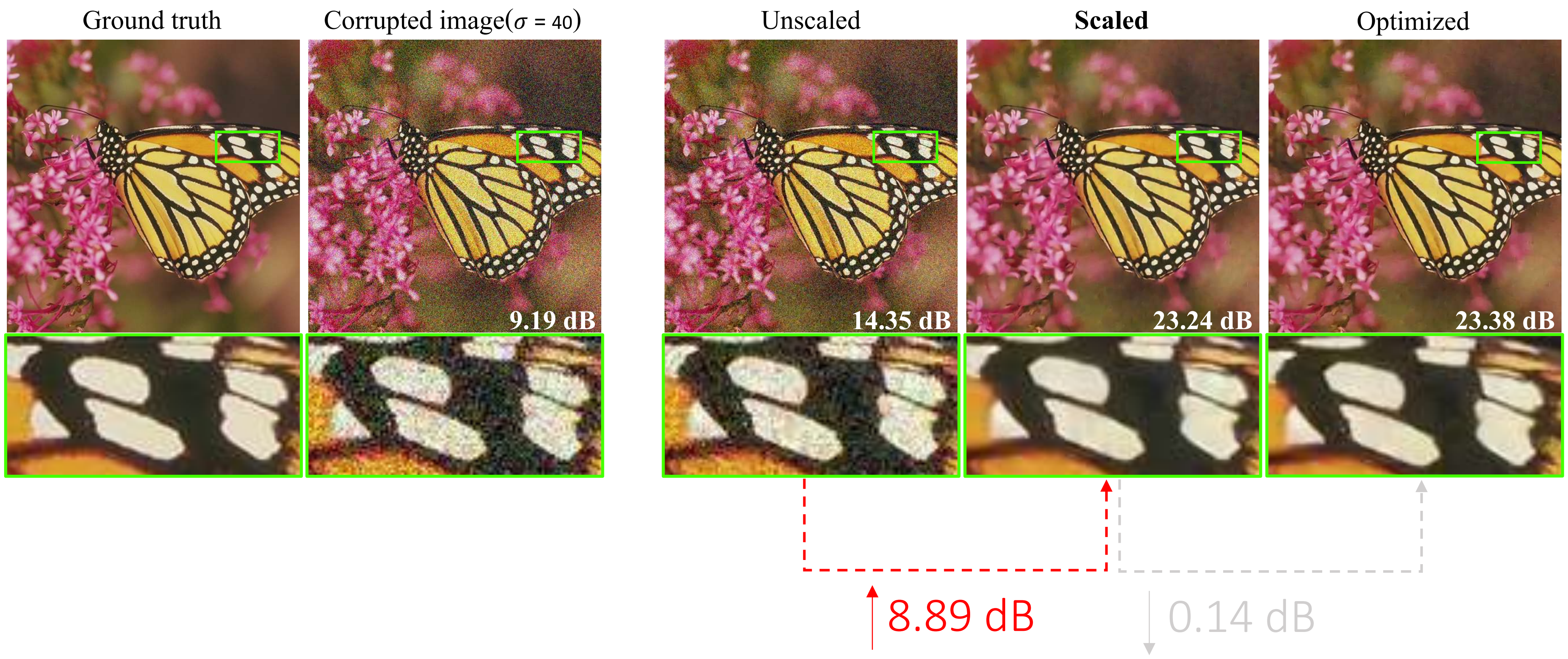
- CNN trained on noise level $\sigma = 20$, applied on noise level $\sigma = 30$, difference $\Delta_\sigma = 10$.



*Number written to image is signal-to-noise ratio (SNR)

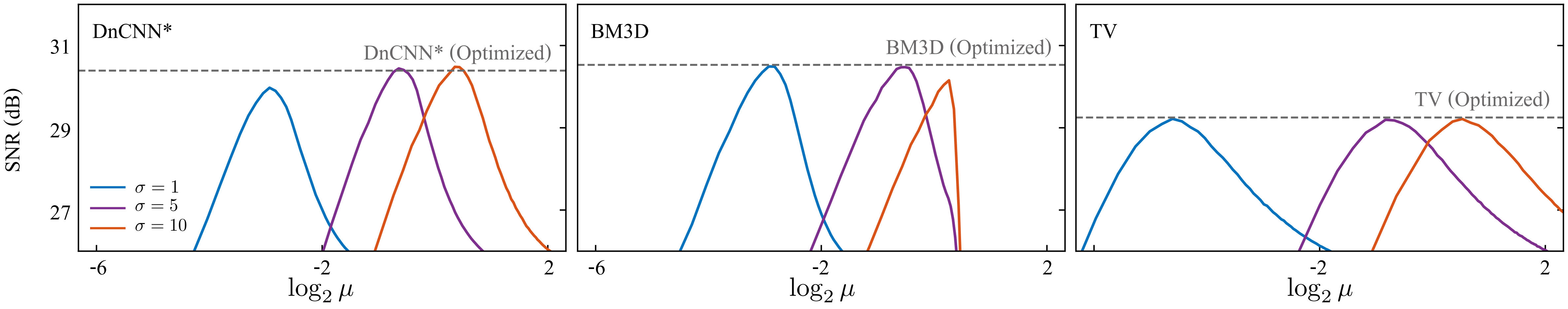
Performance of denoiser scaling

- CNN trained on noise level $\sigma = 20$, applied on noise level $\sigma = 40$, difference $\Delta_{\sigma} = 20$.



Theoretical analysis of denoiser scaling

- Denoiser scaling is proved to have the following properties:
  - ✪ When the denoiser is a minimum mean-squared error (MMSE) denoiser, adjusting μ is equivalent to scale the variance of AWGN by μ^{-2} in the MMSE estimation.
  - ✪ When denoiser is a proximal map $\mathsf{prox}_{\lambda h}(\boldsymbol{z}) := \arg\min_{\boldsymbol{x}}\{\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2 + \lambda h(\boldsymbol{x})\}$, where regularizer $h(\cdot)$ is 1-homogeneous with $h(\mu\,\cdot) = \mu\, h(\cdot)$, adjusting μ is equivalent to adjusting the weighting parameter λ of h.



*Plots are generated with noisy image at noise level σ = 7.23

PnP algorithms with denoiser scaling

Washington University in St. Louis

- PnP algorithms with denoiser scaling

PnP-FISTA

$$\boldsymbol{z}^k = \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$$
$$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k)$$
$$\boldsymbol{s}^k = \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

$$\Downarrow$$

Scaled PnP-FISTA

$$\boldsymbol{z}^k = \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$$
$$\boldsymbol{x}^k = \mu^{-1}\mathsf{D}_\sigma(\mu \boldsymbol{z}^k)$$
$$\boldsymbol{s}^k = \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

PnP-ADMM

$$\boldsymbol{z}^k = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$$
$$\boldsymbol{x}^k = \mathsf{D}_\sigma((\boldsymbol{z}^k + \boldsymbol{s}^{k-1}))$$
$$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$$

$$\Downarrow$$

Scaled PnP-ADMM

$$\boldsymbol{z}^k = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1})$$
$$\boldsymbol{x}^k = \mu^{-1}\mathsf{D}_\sigma(\mu(\boldsymbol{z}^k + \boldsymbol{s}^{k-1}))$$
$$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)$$

Inverse problem examples

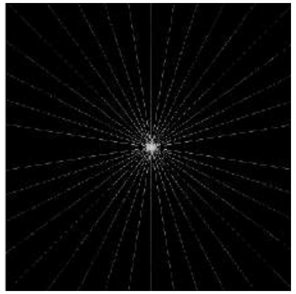
- Image Super-resolution (SR) and Magnetic resonance imaging (MRI) problem

Low-resolution image

SR inverse problem

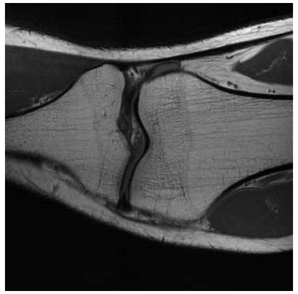
High-resolution image

Under-sampled frequencies



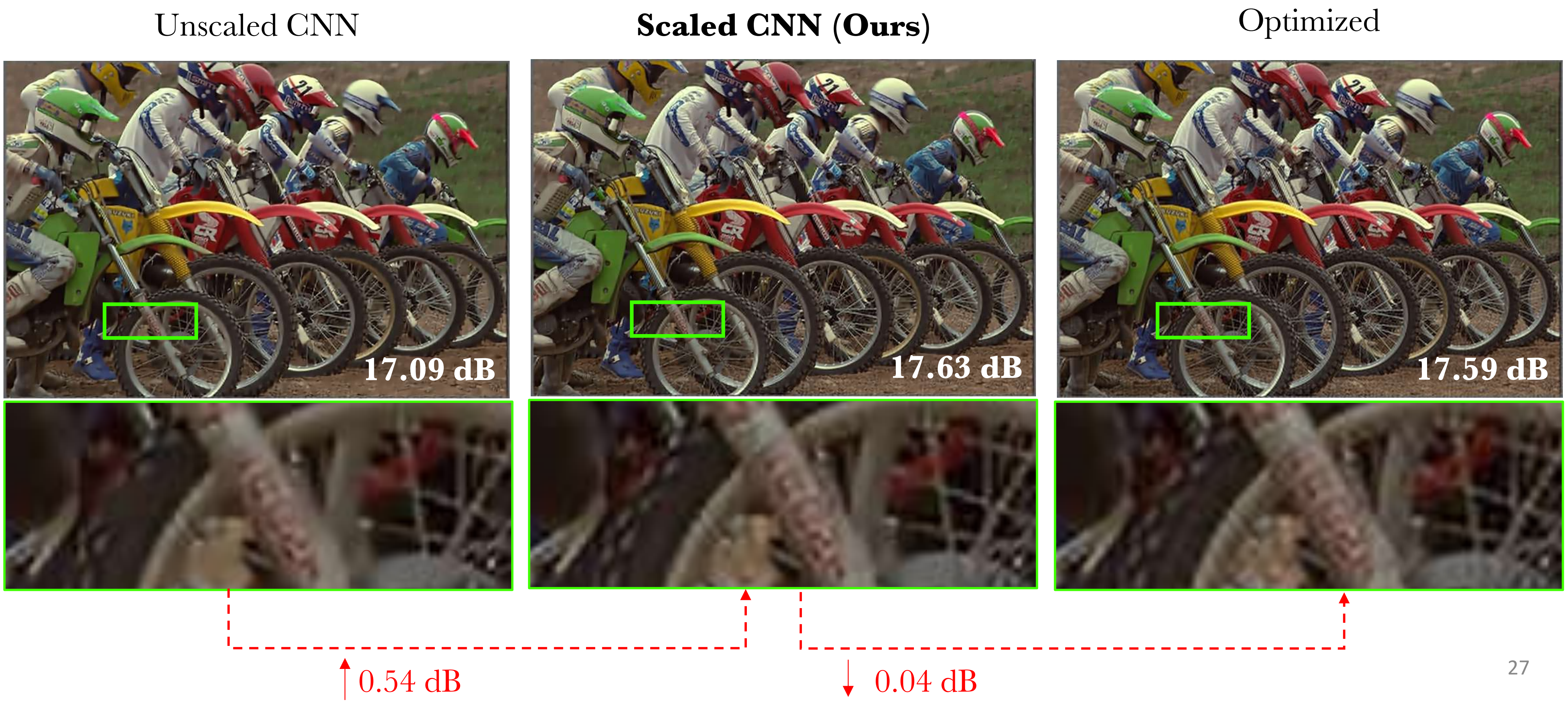
MRI inverse problem

Clean image



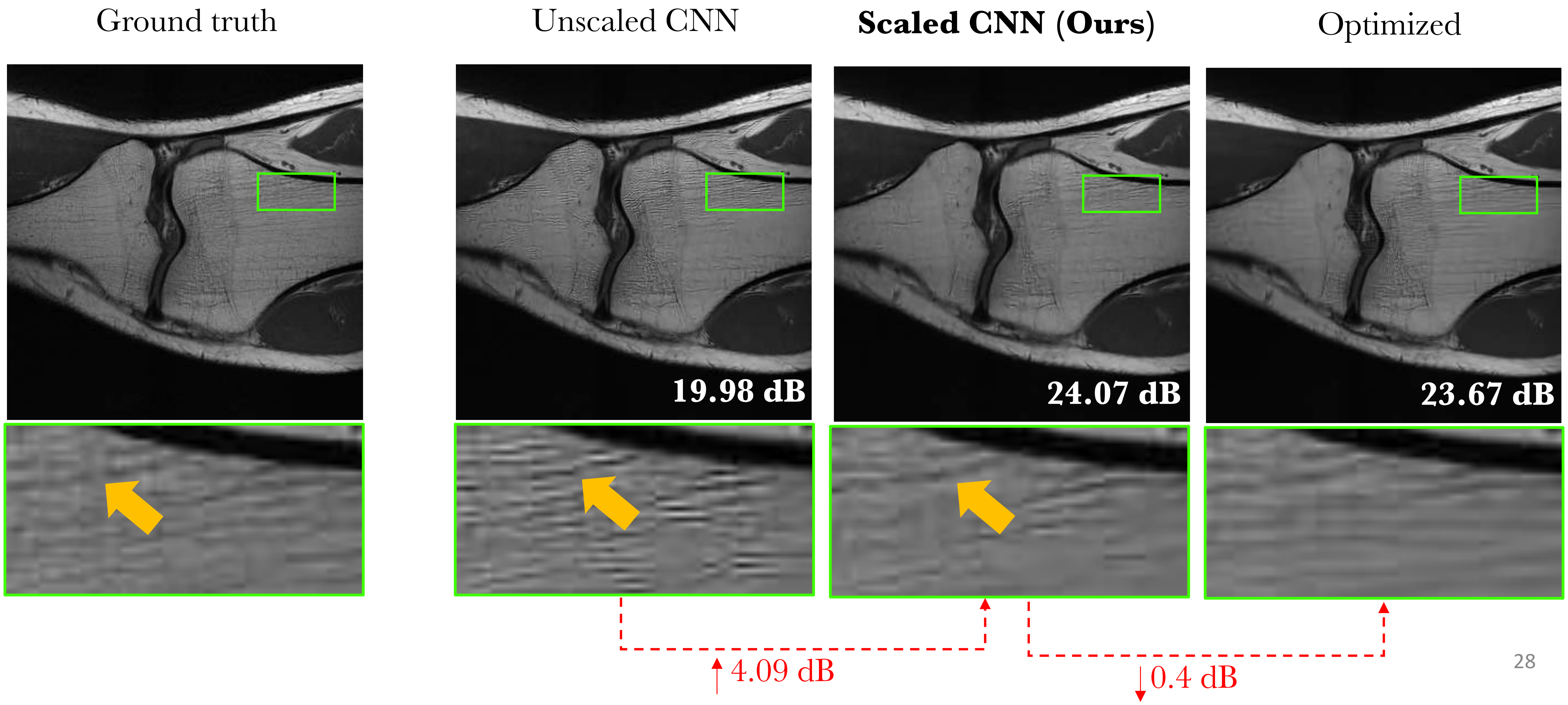
Scaling performance in image SR problem

- Scaling technique can sharpen the blurry edges caused by the suboptimal denoiser.



Scaling performance in MRI problem

- Scaling technique can alleviate the artifacts caused by the suboptimal denoiser.



Theoretical challenge of PnP scheme

Washington University in St. Louis

Good!

- Denoiser scaling can effectively boost the performance of PnP algorithms and achieve the optimal results for different popular denoisers!

However, ….

- PnP algorithm lose the interpretation as an optimization problem for an arbitrary denoiser.

Theoretical challenge of PnP scheme

Washington University in St. Louis

- PnP algorithm lose the interpretation as an optimization problem for an arbitrary denoiser.

FISTA

$$\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$$

$$\boldsymbol{x}^k \leftarrow \mathsf{prox}_{\gamma h}(\boldsymbol{z}^k) \quad \text{proximal map}$$

$$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

↕ objective function

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{g(\boldsymbol{x}) + \underbrace{h(\boldsymbol{x})}_{\text{prior}}\}$$

$\Rightarrow$

PnP-FISTA

$$\boldsymbol{z}^k = \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$$

$$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k) \quad \text{any off-the-shelf denoiser}$$

$$\boldsymbol{s}^k = \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

↕ objective function

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \{g(\boldsymbol{x}) + \underbrace{\ ?\ }_{\text{prior}}\}$$

For most off-the-shelf denoisers, it is impossible to write an explicit regularizer!

Provable Convergence of PnP with MMSE denoisers

Washington University in St. Louis

- Can we build a relationship between some cost function f and some type of denoisers when running PnP?

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) \quad + \quad ?$$

- Yes! Let's consider a MMSE denoiser:

$$\mathsf{D}_\sigma(\boldsymbol{y}) = \mathbb{E}[\boldsymbol{x} \mid \boldsymbol{y}] \quad \text{where} \quad \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}, \quad \boldsymbol{e} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}), \quad \boldsymbol{x} \sim p_{\boldsymbol{x}}.$$

- Convergence of PnP-ISTA with MMSE denoisers [For details, see Xu' 20(2)]

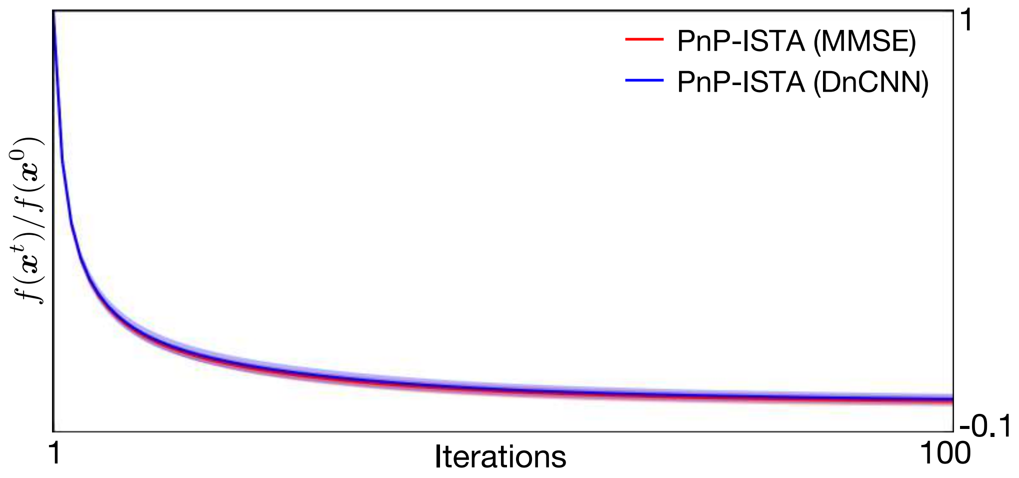
- The iterates produced by PnP-ISTA with an MMSE denoiser converge to a stationary point of some global cost function.

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + h(\boldsymbol{x}) \quad \text{with} \quad h(\boldsymbol{x}) := \begin{cases} -\frac{1}{2\gamma}\|\boldsymbol{x} - \mathsf{D}_\sigma^{-1}(\boldsymbol{x})\|^2 + \frac{\sigma^2}{\gamma} h_\sigma(\mathsf{D}_\sigma^{-1}(\boldsymbol{x})) & \text{for } \boldsymbol{x} \in \mathcal{X} \\ +\infty & \text{for } \boldsymbol{x} \notin \mathcal{X}, \end{cases}$$

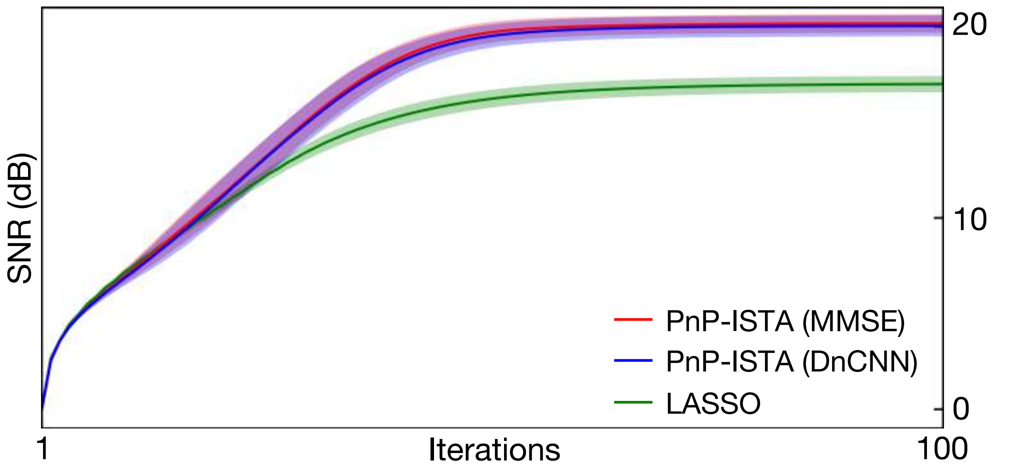
Provable Convergence of PnP with MMSE denoisers

Washington University in St. Louis

- Convergence and reconstruction performance of using MMSE denoiser and DnCNN for Bernoulli-Gaussian signals in compressive sensing



Convergence of PnP-ISTA for exact and approximate MMSE denoisers on loss function.



Convergence of PnP-ISTA for exact and approximate MMSE denoisers on SNR.

Summary of the talk

Washington University in St. Louis

1. Introduction to Plug-and-Play Priors (PnP)

- Denoiser strength selecting challenge
- Optimization interpretation and convergence analysis challenge

2. Denoiser scaling technique [Xu'20(1)]

- We proposed a denoiser scaling technique that can help with the denoising strength tuning especially for CNN type of denoisers.
- We showed that denoiser scaling can effectively boost the performance of PnP algorithms and achieve the optimal results.

3. Optimization interpretation and convergence analysis of PnP with MMSE denoisers [Xu'20(2)]

- We show that the iterates produced by PnP-ISTA with an MMSE denoiser converge to a stationary point of some global cost function.

Additional key citations

- [Chan'16]: Chan et.al. Plug-and-play ADMM for image restoration: Fixed-point convergence and applications, *IEEE TCI, 2016*
- [Venkat'13]: Venkat et.al. Plug-and-play priors for model based reconstruction, IEEE GlobalSIP, 2013.
- [Zakharov'19]: Zakharov et al. Few-Shot Adversarial Learning of Realistic Neural Talking Head Models, ICCV, 2019
- [Sun'19]: Sun et.al. Block coordinate regularization by denoising, NeurIPS, 2019.
- [Xia'19]: Xia et.al. Training Image Estimators without Image Ground-Truth, NeurIPS, 2019
- [Lee'18]: Lee et al. Deep residual learning for accelerated MRI using magnitude and phase networks, Trans. Bio. Eng. 2018
- [Ryu'19]: Ryu et al. Plug-and-play methods provably converge with properly trained denoisers, ICML 2019
- [Romano'17]: Romano et al. A little engine that could do: regularization by denoising, SIAM J. Imaging Sci, 2017
- [Sun'18]: Sun et.al. Efficient and accurate inversion of multiple scattering with deep learning, Opt. Express 26, 2018
- [Lehtinen'18]: Lehtinen et al. Noise2Noise, *ICML*. PMLR, 2018.
- [Ulyanov'18]: Ulyanov et al. Deep image prior, ICCV, 2018.
- [Aggarwal'18]: Aggarwal et al. MoDL: Model-based deep learning architecture for inverse problems, *IEEE TMI*, 2018
- [Zhang'18]: Zhang et al. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing, ICCV, 2018
- [Yang'16]: Yang et al. Deep ADMM-Net for compressive sensing MRI NeurIPS, 2016.
- [Liu'19]: Liu et al. Infusing Learned Priors into Model-Based Multispectral Imaging, *arXiv preprint arXiv:1909.09313* (2019).

Thanks!