Washington University in St. Louis

Computational Imaging: Reconciling **Models** and **Learning**

Ulugbek S. Kamilov
Computational Imaging Group (CIG)

point your camera
get the slides

ICCW LCI (Seoul, Republic or Korea) — 2 Nov 2019

@ukmlv • cigroup.wustl.edu • kamilov@wustl.edu

Computational imaging is going through a paradigm shift driven by machine learning

Computational imaging is going through a paradigm shift driven by machine learning

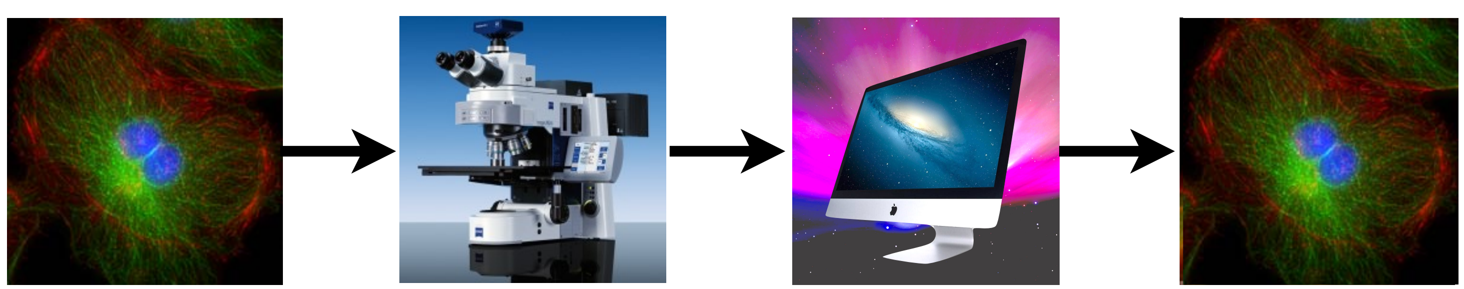
Past: Focus on hardware for image formation

input instrument output

Computational imaging is going through a paradigm shift driven by machine learning

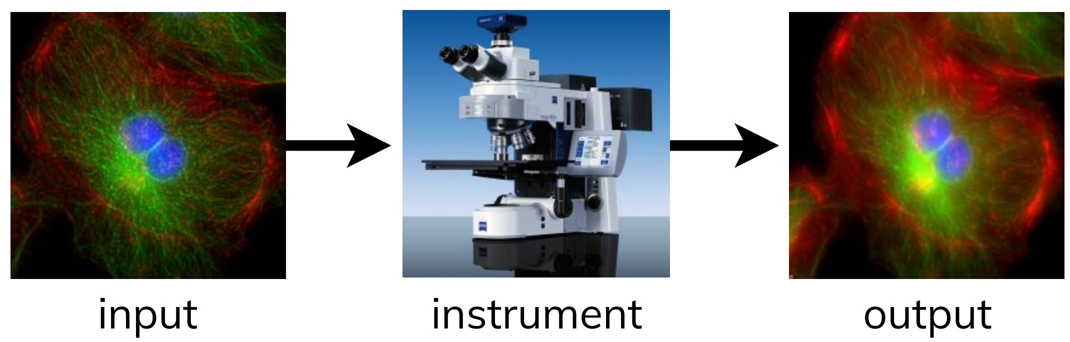
Past: Focus on hardware for image formation

Present: Use digital signal processing for improved performance

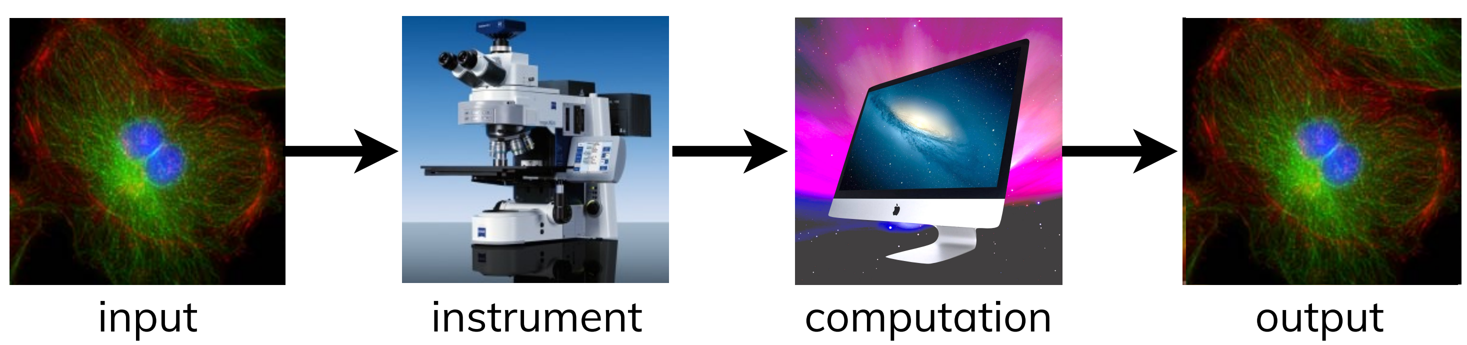


Computational imaging is going through a paradigm shift driven by machine learning

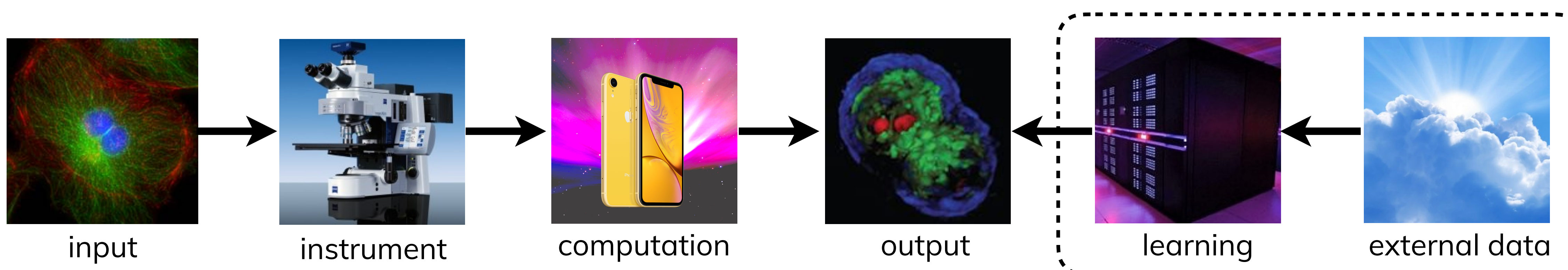
Past: Focus on hardware for image formation



Present: Use digital signal processing for improved performance



Near future: Machine learning for retrieving ***hidden*** information



Today we will talk about

- **Imaging as an inverse problem**
  From forward models to regularized inversion
- **Combining models and learning for imaging**
  From PnP to RED using denoising neural nets
- **Scaling image formation to large problems**
  Variants: PnP-SGD, On-RED, and BC-RED

Today we will talk about

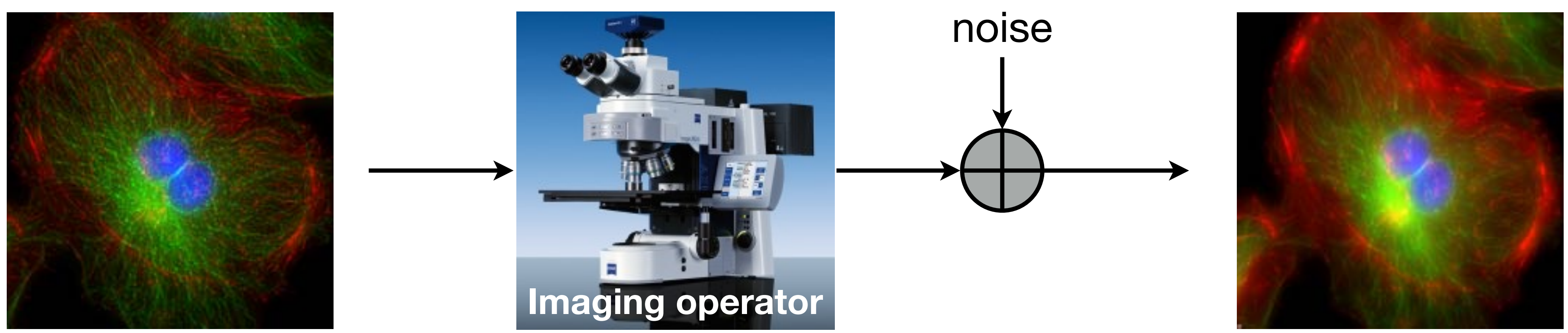
- Imaging as an inverse problem
  From forward models to regularized inversion

- Combining models and learning for imaging
  From PnP to RED using denoising neural nets

- Scaling image formation to large problems
  Variants: PnP-SGD, On-RED, and BC-RED

The vast majority of imaging problems can be formulated as inverse problems

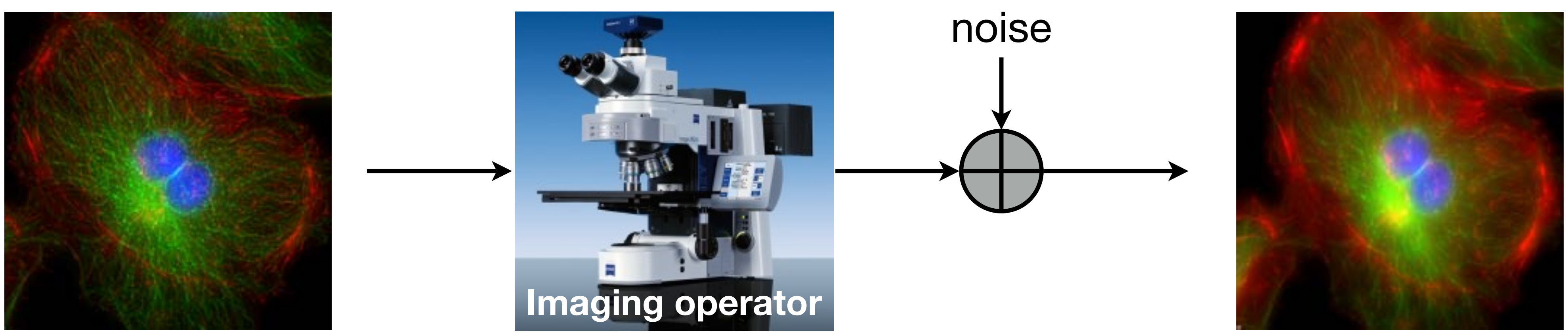
The vast majority of imaging problems can be formulated as inverse problems



$\boldsymbol{x}$ $\qquad$ $\mathbf{H}$ $\qquad$ $\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$

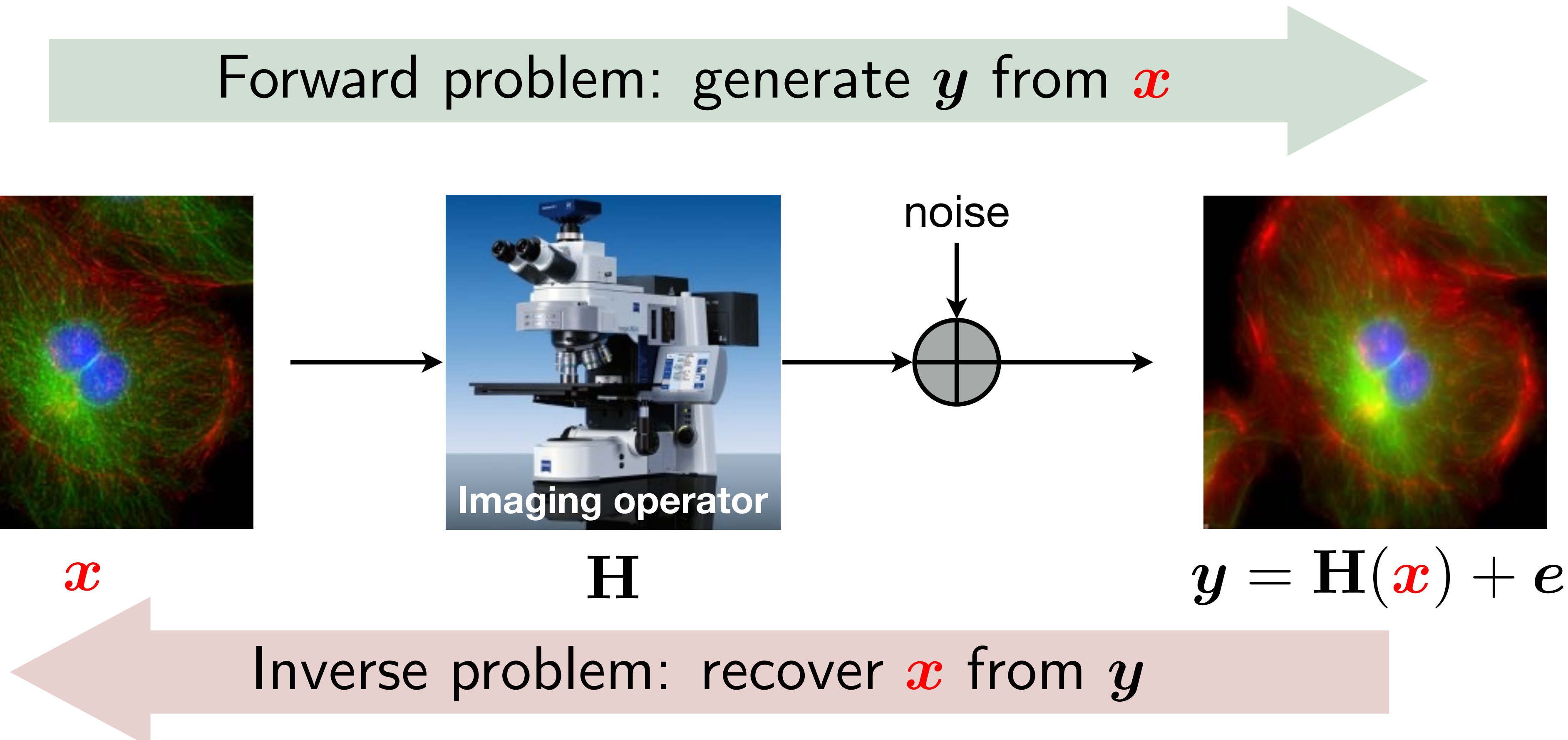
The vast majority of imaging problems can be formulated as inverse problems

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



$\boldsymbol{x}$ $\qquad$ $\mathbf{H}$ $\qquad$ $\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$

The vast majority of imaging problems can be formulated as inverse problems

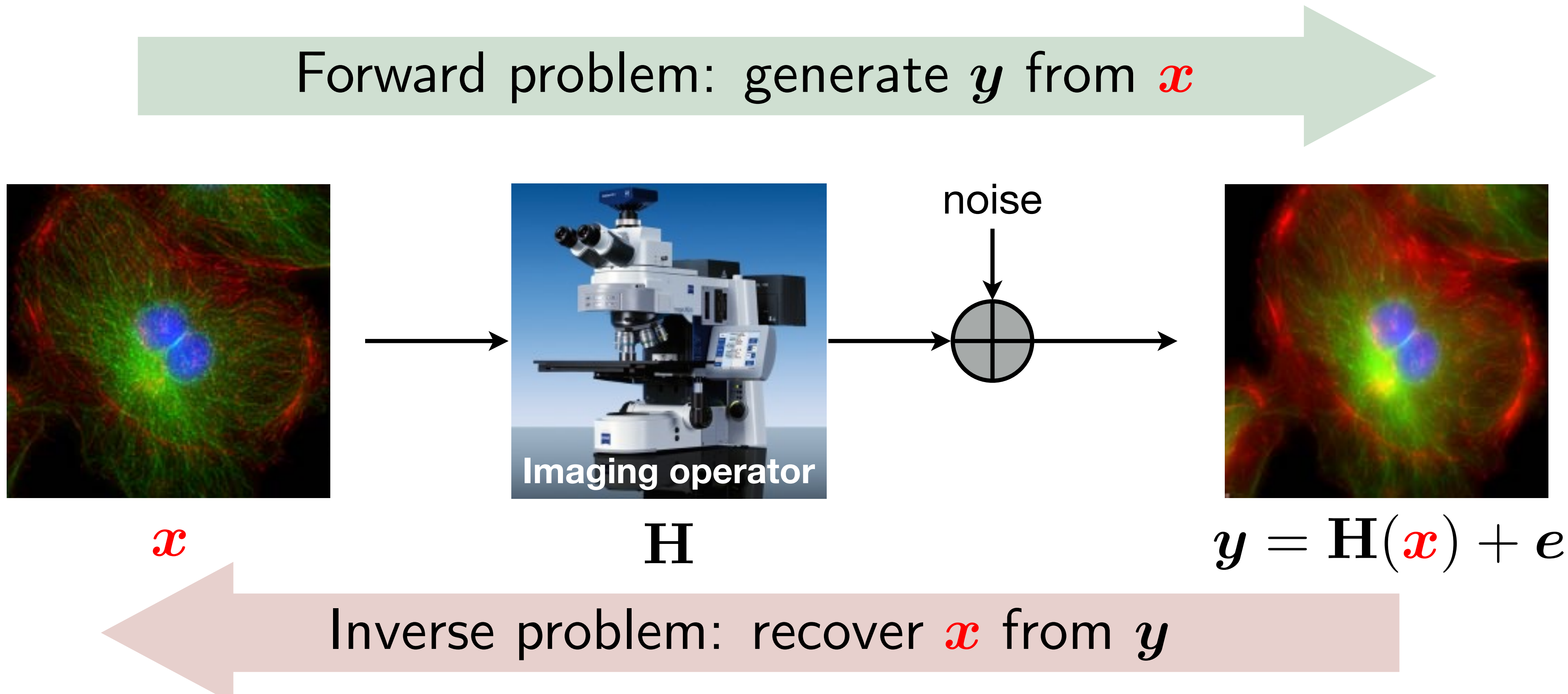


The vast majority of imaging problems can be formulated as inverse problems

Imaging Problem	Rediation	Forward Model	Variations
2D or 3D tomography	coherent x-ray	$\boldsymbol{y}_i = \mathbf{R}_{\boldsymbol{\theta}_i} \boldsymbol{x}$	parallel, cone beam
3D deconvolution microscopy	fluorescence	$\boldsymbol{y} = \mathbf{H} \boldsymbol{x}$	brightfield, confocal, light sheet
structured illumination microscopy (SIM)	fluorescence	$\boldsymbol{y}_i = \mathbf{H}\mathbf{W}_i \boldsymbol{x}$	full 3D reconstruction, non-sinusoidal patters
positron emission tomography (PET)	gamma rays	$\boldsymbol{y}_i = \mathbf{H}_{\boldsymbol{\theta}_i} \boldsymbol{x}$	list mode with time-of-flight
magnetic resonance imaging (MRI)	radio frequency	$\boldsymbol{y} = \mathbf{S}\mathbf{F} \boldsymbol{x}$	uniform or nonuniform sampling in k-space
dynamic MRI	radio frequency	$\boldsymbol{y}_{t,i} = \mathbf{S}_t\mathbf{F}\mathbf{W}_i \boldsymbol{x}$	gated or nongated, retrospective registration
optical diffraction tomography (ODT)	coherent light	$\boldsymbol{y}_i = \mathbf{W}_i\mathbf{F} \boldsymbol{x}$	with holography or gating interferometry

Source: Michael Unser

The vast majority of imaging problems can be formulated as inverse problems

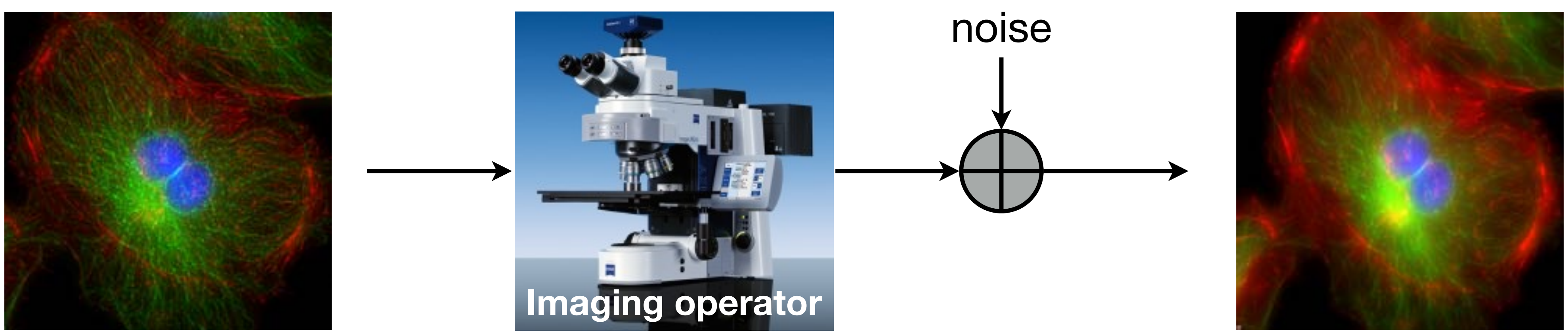


What makes imaging inverse problems challenging?

Washington University in St. Louis

The vast majority of imaging problems can be formulated as inverse problems

Forward problem: generate y from x



x $\quad$ $\mathbf{H}$ $\quad$ $y = \mathbf{H}(x) + e$

Inverse problem: recover x from y

What makes imaging inverse problems challenging?

- solution is not unique
- data is noisy (measurement & model noise)
- problem is intractable
- data is very high-dimensional

Image denoising is a fundamental inverse problem that highlights the importance of regularization

Image denoising is a fundamental inverse problem that highlights the importance of regularization

Additive white Gaussian noise (AWGN) model

$$\boxed{\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{e}}$$

noisy observation =
unknown desired + unknown undesired

Image denoising is a fundamental inverse problem that highlights the importance of regularization

Additive white Gaussian noise (AWGN) model

$$\boxed{\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{e}}$$

Problem: There are ∞ many possible solutions!

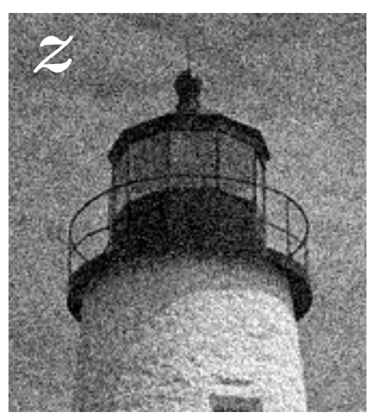
Image denoising is a fundamental inverse problem that highlights the importance of regularization

Additive white Gaussian noise (AWGN) model

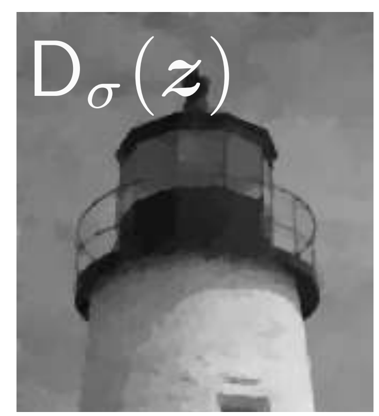
$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{e}$$

Image denoiser is a function for separating signal from noise

$$\mathsf{D}_\sigma : \text{more noisy image} \mapsto \text{less noisy image}$$



$\mapsto$



Source: Pascal Getreuer

Image denoising is a fundamental inverse problem that highlights the importance of regularization

Additive white Gaussian noise (AWGN) model

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{e}$$

Image denoiser is a function for separating signal from noise

$$\mathsf{D}_\sigma : \text{more noisy image} \mapsto \text{less noisy image}$$

Proximal operator formulates denoising as optimization

$$\mathsf{prox}_{\tau h}(\boldsymbol{z}) := \operatorname*{arg\,min}_{\boldsymbol{x}} \left\{ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \tau h(\boldsymbol{x}) \right\}$$

Source: Pascal Getreuer

Image denoising is a fundamental inverse problem that highlights the importance of regularization

Additive white Gaussian noise (AWGN) model

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{e}$$

Image denoiser is a function for separating signal from noise

$$\mathsf{D}_\sigma : \text{more noisy image} \mapsto \text{less noisy image}$$

Proximal operator formulates denoising as optimization

$$\mathsf{prox}_{\tau h}(\boldsymbol{z}) \; := \; \operatorname*{arg\,min}_{\boldsymbol{x}} \left\{ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \tau h(\boldsymbol{x}) \right\}$$

It turns out proximal operators can be used to compute a solution to many imaging inverse problems!

Regularized inversion provides a unified approach for integrating physics and prior knowledge

Regularized inversion provides a unified approach for integrating physics and prior knowledge

Image recovery is often an **ill-posed inverse problem**

$$\boxed{\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}}$$

noise, but also **information loss**, **model uncertainties**, etc.

Regularized inversion provides a unified approach for integrating physics and prior knowledge

Image recovery is often an ill-posed inverse problem

$$\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$$

Formulation as a **regularized optimization** problem

$$\boxed{\widehat{\boldsymbol{x}} = \operatorname*{arg\,min}_{\boldsymbol{x}} \{f(\boldsymbol{x})\}}$$

$$\boxed{f(\boldsymbol{x}) \; := \; g(\boldsymbol{x}) + h(\boldsymbol{x})}$$

data-fidelity term + regularizer

Regularized inversion provides a unified approach for integrating physics and prior knowledge

Image recovery is often an ill-posed inverse problem

$$\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$$

Formulation as a regularized optimization problem

$$\widehat{\boldsymbol{x}} = \mathop{\mathsf{arg\,min}}_{\boldsymbol{x}} \{f(\boldsymbol{x})\} \qquad f(\boldsymbol{x}) \;:=\; g(\boldsymbol{x}) + h(\boldsymbol{x})$$

Example: **Linear inverse problems** (20th century theory)

$$\begin{aligned}\widehat{\boldsymbol{x}} &= \mathop{\mathsf{arg\,min}}_{\boldsymbol{x}} \left\{\frac{1}{2}\|\boldsymbol{y} - \mathbf{H}\boldsymbol{x}\|_2^2 + \frac{\lambda}{2}\|\mathbf{D}\boldsymbol{x}\|_2^2\right\} \\ &= (\mathbf{H}^{\mathsf{H}}\mathbf{H} + \lambda\mathbf{D}^{\mathsf{H}}\mathbf{D})^{-1}\mathbf{H}^{\mathsf{H}}\boldsymbol{y} = \mathbf{R}_\lambda \cdot \boldsymbol{y}\end{aligned}$$

Regularized inversion provides a unified approach for integrating physics and prior knowledge

Image recovery is often an ill-posed inverse problem

$$\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$$

Formulation as a regularized optimization problem

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\arg\min}\,\{f(\boldsymbol{x})\} \qquad f(\boldsymbol{x}) := g(\boldsymbol{x}) + h(\boldsymbol{x})$$

Example: **Linear inverse problems** (20th century theory)

$$\begin{aligned}\widehat{\boldsymbol{x}} &= \underset{\boldsymbol{x}}{\arg\min}\left\{\frac{1}{2}\|\boldsymbol{y} - \mathbf{H}\boldsymbol{x}\|_2^2 + \frac{\lambda}{2}\|\mathbf{D}\boldsymbol{x}\|_2^2\right\}\\ &= (\mathbf{H}^{\mathsf{H}}\mathbf{H} + \lambda\mathbf{D}^{\mathsf{H}}\mathbf{D})^{-1}\mathbf{H}^{\mathsf{H}}\boldsymbol{y} = \boxed{\mathbf{R}_\lambda \cdot \boldsymbol{y}}\end{aligned}$$

Interpretation:
“Filtered” backprojection

Regularized inversion provides a unified approach for integrating physics and prior knowledge

Image recovery is often an ill-posed inverse problem

$$\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$$

Formulation as a regularized optimization problem

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\mathsf{arg\,min}}\, \{f(\boldsymbol{x})\} \qquad f(\boldsymbol{x}) := g(\boldsymbol{x}) + h(\boldsymbol{x})$$

Example: **Linear inverse problems** (20th century theory)

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\mathsf{arg\,min}} \left\{ \boxed{\frac{1}{2}\|\boldsymbol{y} - \mathbf{H}\boldsymbol{x}\|_2^2} + \frac{\lambda}{2}\boxed{\|\mathbf{D}\boldsymbol{x}\|_2^2} \right\}$$
$$= (\mathbf{H}^{\mathsf{H}}\mathbf{H} + \lambda \mathbf{D}^{\mathsf{H}}\mathbf{D})^{-1}\mathbf{H}^{\mathsf{H}}\boldsymbol{y} = \mathbf{R}_\lambda \cdot \boldsymbol{y}$$

Interpretation:
“Filtered” backprojection

Assumption:
Gaussians everwhere

Regularized inversion provides a unified approach for integrating physics and prior knowledge

Image recovery is often an ill-posed inverse problem

$$\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$$

Formulation as a regularized optimization problem

$$\widehat{\boldsymbol{x}} = \mathop{\mathsf{arg\,min}}_{\boldsymbol{x}} \{f(\boldsymbol{x})\} \qquad f(\boldsymbol{x}) := g(\boldsymbol{x}) + h(\boldsymbol{x})$$

Example: **Maximum a posteriori probability (MAP)** estimator

$$\boxed{g(\boldsymbol{x}) = -\log(p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x}))} \qquad \boxed{h(\boldsymbol{x}) = -\log(p_{\boldsymbol{x}}(\boldsymbol{x}))}$$

likelihood term   prior term

Washington University in St. Louis

Regularized inversion provides a unified approach for integrating physics and prior knowledge

Image recovery is often an ill-posed inverse problem

$$\boldsymbol{y} = \mathbf{H}(\boldsymbol{x}) + \boldsymbol{e}$$

Formulation as a regularized optimization problem

$$\widehat{\boldsymbol{x}} = \mathop{\mathsf{arg\,min}}_{\boldsymbol{x}} \{f(\boldsymbol{x})\} \qquad f(\boldsymbol{x}) := g(\boldsymbol{x}) + h(\boldsymbol{x})$$

Example: **AWGN** and **sparsity-promiting** prior

$$\boxed{g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \mathbf{H}(\boldsymbol{x})\|_2^2} \qquad \boxed{h(\boldsymbol{x}) = \lambda\|\mathbf{D}\boldsymbol{x}\|_1}$$

smooth data-fidelity
(least-squares)

nonsmooth regularizer
(gradient sparsity)

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

Optimization problem

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

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

Fast iterative shrinkage/thresholding algorithm (FISTA) vs. alternating direction method of multipliers (ADMM)

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

FISTA: grad data + prox prior

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

ADMM: prox data + prox prior

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

FISTA and ADMM are two popular algorithms for large-scale and nonsmooth optimization

Fast iterative shrinkage/thresholding algorithm (FISTA) vs. alternating direction method of multipliers (ADMM)

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

Both FISTA and ADMM alternate between increasing data consistency and reducing noise

- $\mathsf{I} - \gamma \nabla g$: less data consistent $\mapsto$ more data consistent
- $\mathsf{prox}_{\gamma g}$: less data consistent $\mapsto$ more data consistent
- $\mathsf{prox}_{\gamma h}$: more noisy image $\mapsto$ less noisy image

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

Today we will talk about

- Imaging as an inverse problem
  From forward models to regularized inversion
- Combining models and learning for imaging
  From PnP to RED using denoising neural nets
- Scaling image formation to large problems
  Variants: PnP-SGD, On-RED, and BC-RED

Is there a better way to represent images than using fixed sparsity-promoting transforms?

Is there a better way to represent images than using fixed sparsity-promoting transforms?

The interest in sparsity-driven imaging highlighted the importance of **structural priors for image formation**

Is there a better way to represent images than using fixed sparsity-promoting transforms?

The interest in sparsity-driven imaging highlighted the importance of structural priors for image formation

Do we know a more flexible, sophisticated, and data-adaptive tool for **characterizing imaging priors**?

Is there a better way to represent images than using fixed sparsity-promoting transforms?

The interest in sparsity-driven imaging highlighted the importance of structural priors for image formation

Do we know a more flexible, sophisticated, and data-adaptive tool for characterizing imaging priors?

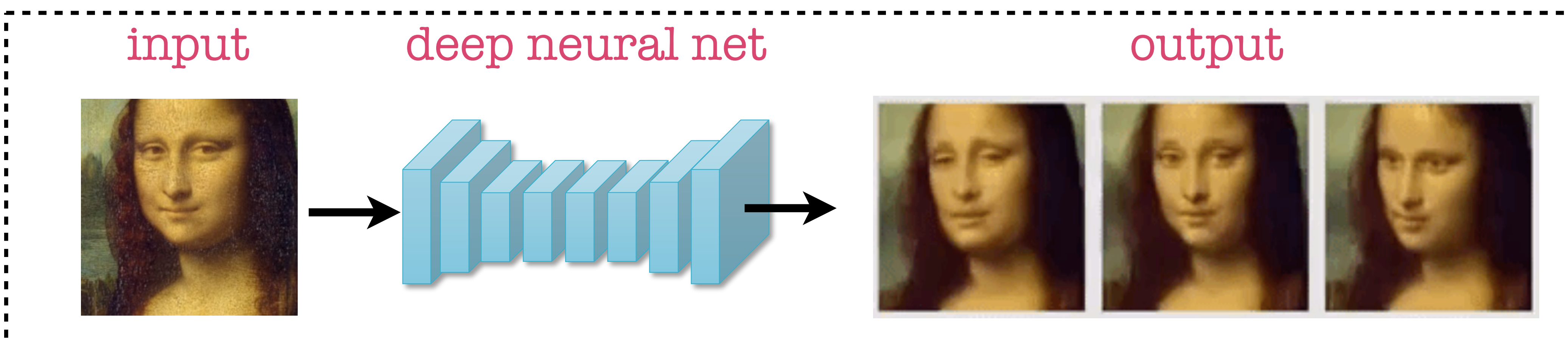
Yes, **deep neural nets** provide a state-of-the-art tool for representing and enforcing sophisticated structural information

Is there a better way to represent images than using fixed sparsity-promoting transforms?

The interest in sparsity-driven imaging highlighted the importance of structural priors for image formation

Do we know a more flexible, sophisticated, and data-adaptive tool for characterizing imaging priors?

Yes, **deep neural nets** provide a state-of-the-art tool for representing and enforcing sophisticated structural information



Zakharov *et al.*, “Few-Shot Adversarial Learning of Realistic Neural Talking Head Models,” *Proc. ICCV*, October 2019

Is there a better way to represent images than using fixed sparsity-promoting transforms?

The interest in sparsity-driven imaging highlighted the importance of structural priors for image formation

Do we know a more flexible, sophisticated, and data-adaptive tool for characterizing imaging priors?

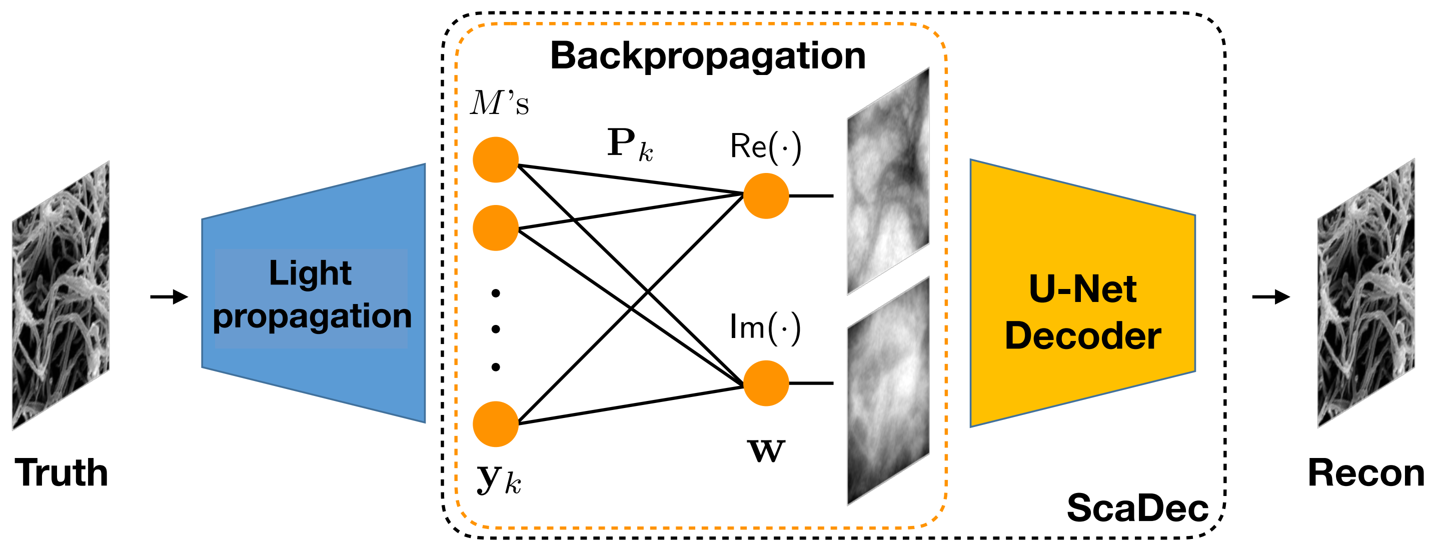
Yes, **deep neural nets** provide a state-of-the-art tool for representing and enforcing sophisticated structural information

How to use deep neural nets as priors for imaging?

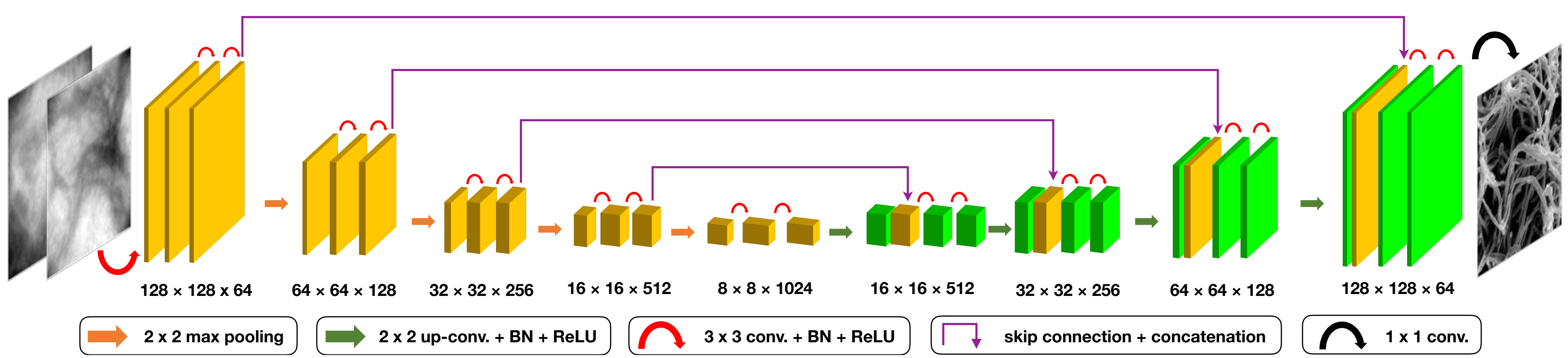
Basic idea: Train a mapping from the measured data to the desired image

Basic idea: Train a mapping from the measured data to the desired image

Example: Simulate light propagation and train a neural net to invert



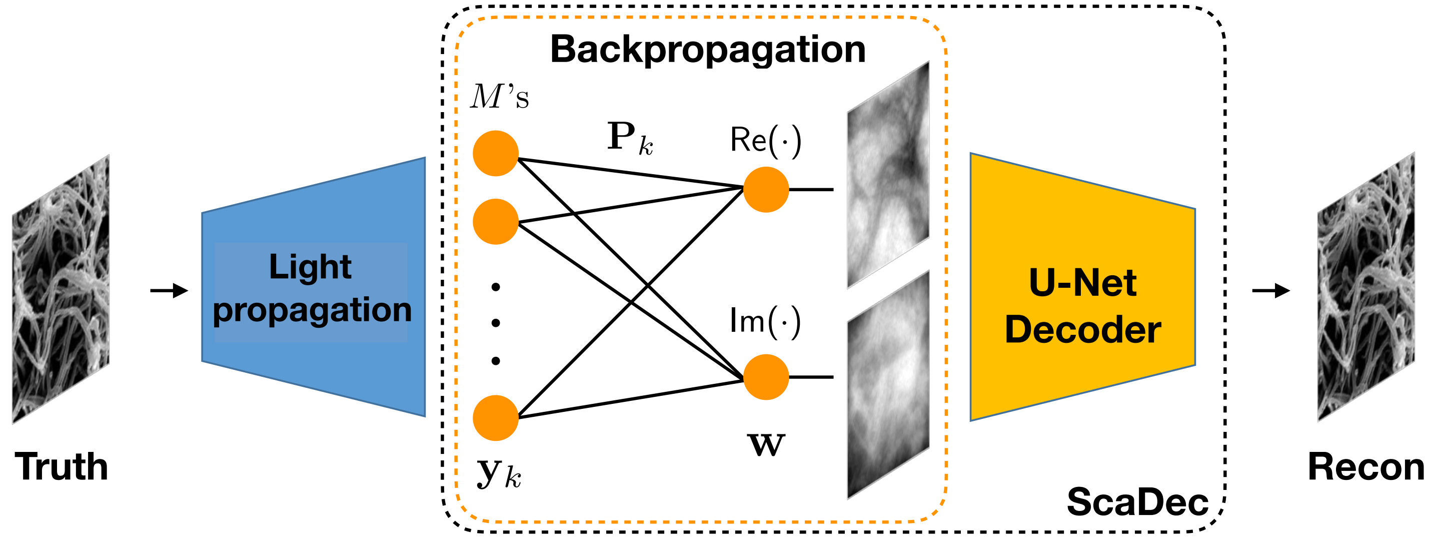
Deep neural net architecture (= U-Net)



Sun *et al.*, "Efficient and accurate inversion of multiple scattering with deep learning," *Optics Express*., vol. 26, no. 11, 2018

Basic idea: Train a mapping from the measured data to the desired image

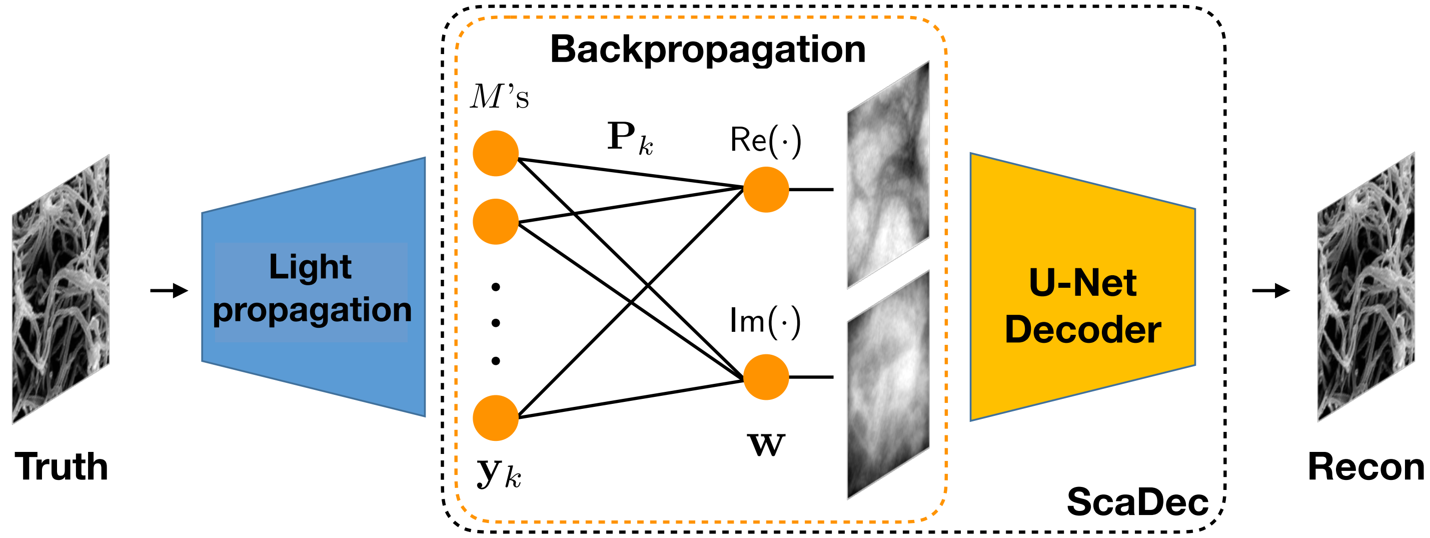
Example: Simulate light propagation and train a neural net to invert



Question: What are some of the key limitations of this approach?

Basic idea: Train a mapping from the measured data to the desired image

Example: Simulate light propagation and train a neural net to invert



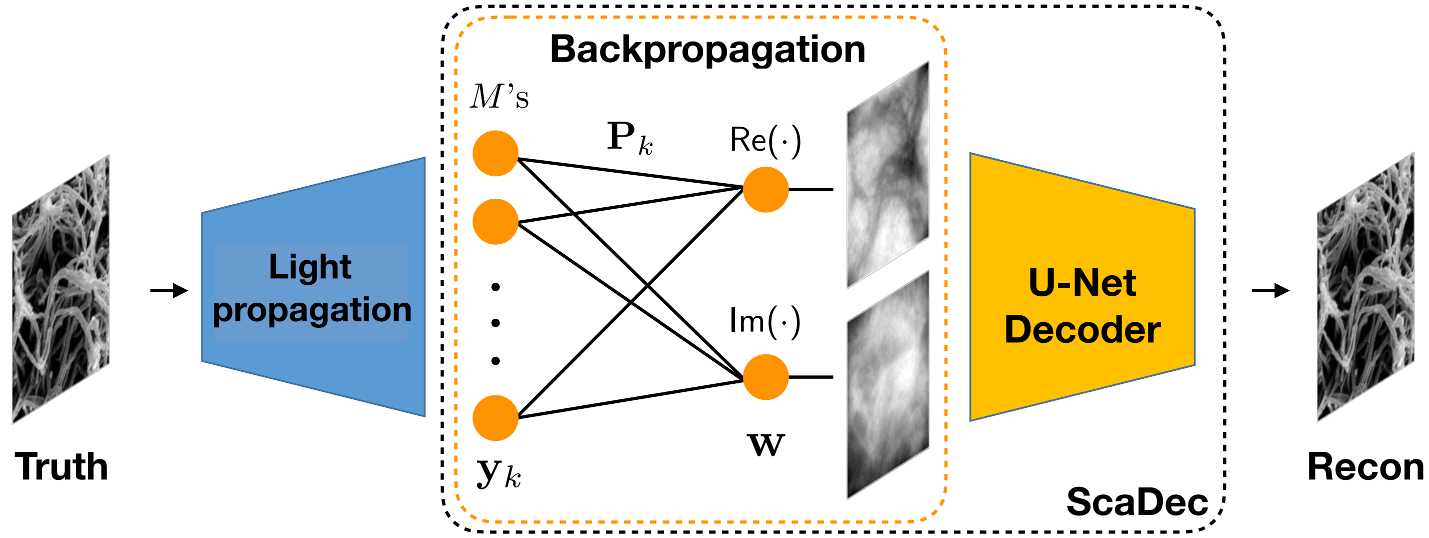
Question: What are some of the key limitations of this approach?

1) The neural net learns both the forward model and the prior

Hard to decouple the individual contributions of D and R

Basic idea: Train a mapping from the measured data to the desired image

Example: Simulate light propagation and train a neural net to invert



Question: What are some of the key limitations of this approach?

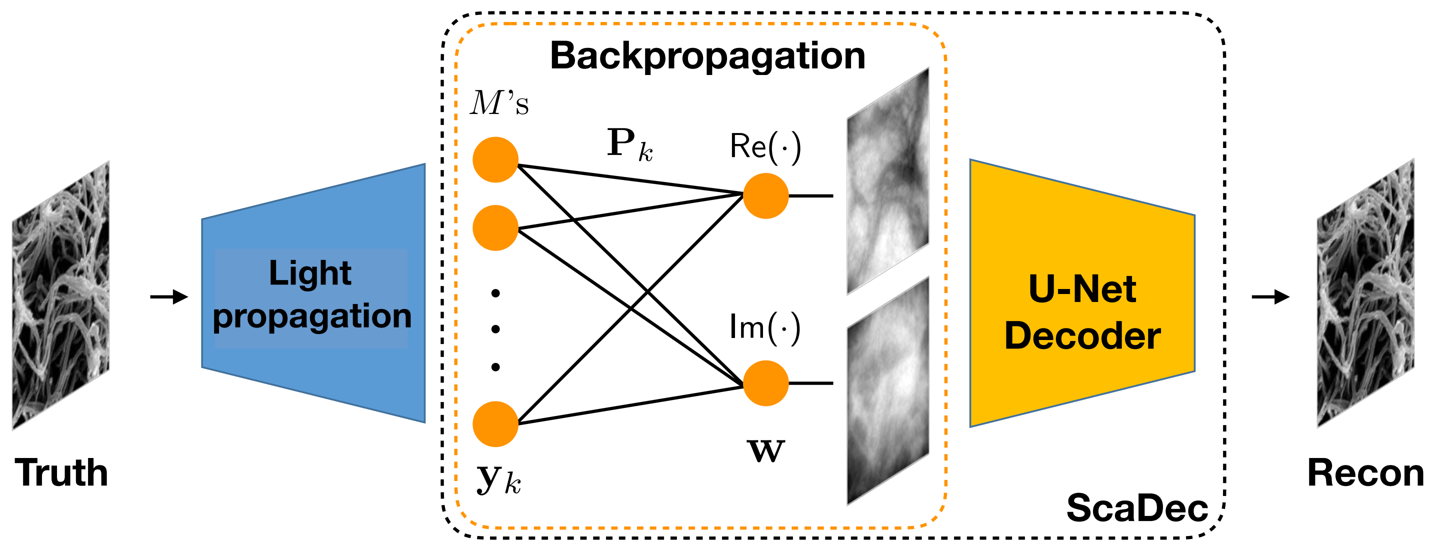
1) The neural net learns both the forward model and the prior

2) Consistency to the measured data is not ensured

No explicit measure of deviation from the data

Basic idea: Train a mapping from the measured data to the desired image

Example: Simulate light propagation and train a neural net to invert



Question: What are some of the key limitations of this approach?

1) The neural net learns both the forward model and the prior

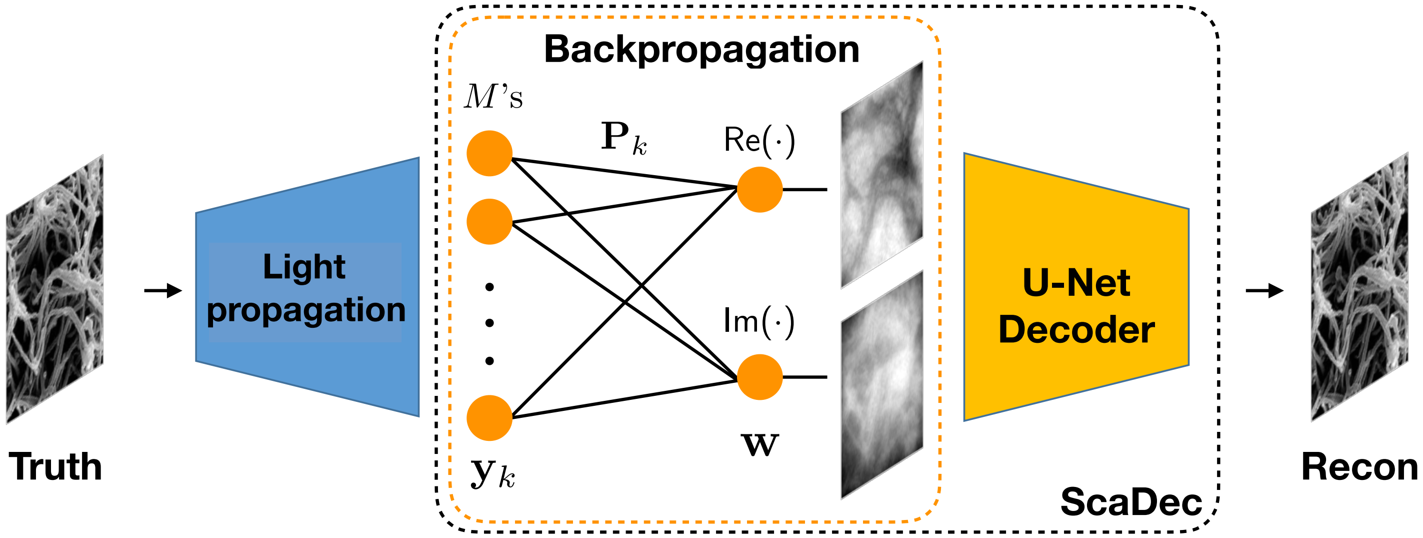
2) Consistency to the measured data is not ensured

3) Finding an efficient end-to-end mapping might be difficult

Fully connected layers are not suitable for large-scale problems

Basic idea: Train a mapping from the measured data to the desired image

Example: Simulate light propagation and train a neural net to invert



Question: What are some of the key limitations of this approach?

1) The neural net learns both the forward model and the prior

2) Consistency to the measured data is not ensured

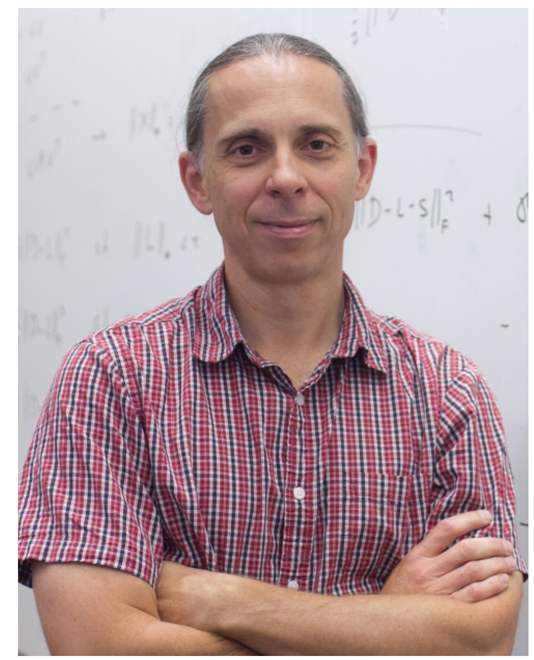
3) Finding an efficient end-to-end mapping might be difficult

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

Venkatakrishnan
(ORNL)

Bouman
(Purdue)

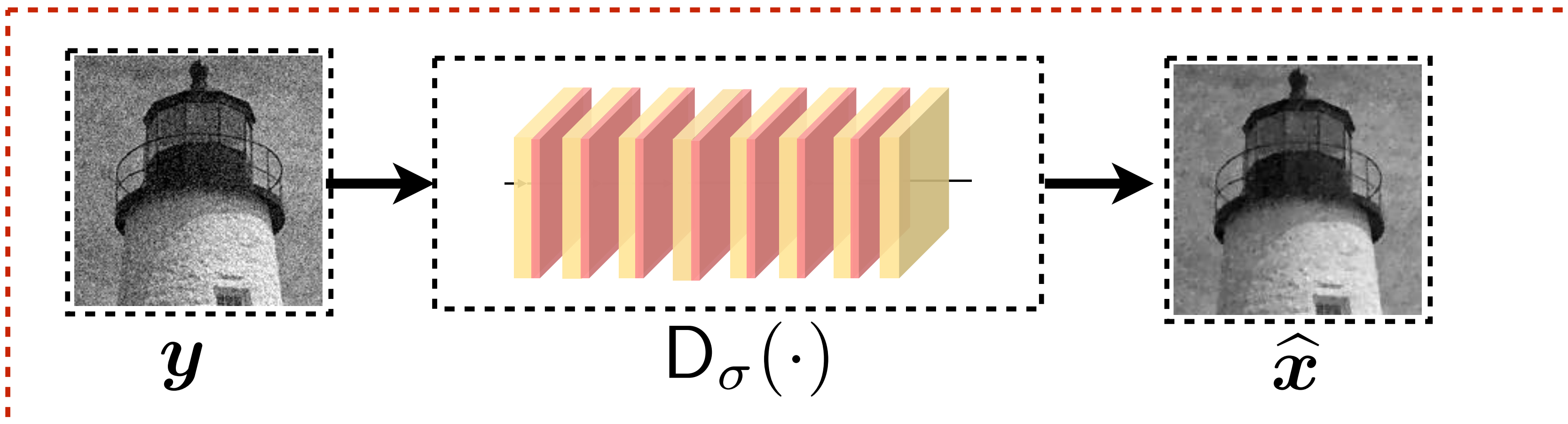


Wohlberg
(LANL)

Venkatakrishnan *et al.*, “Plug-and-Play Priors for Model Based Reconstruction,” *Proc. IEEE GlobalSIP*, pp. 945-948, December 2013.

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

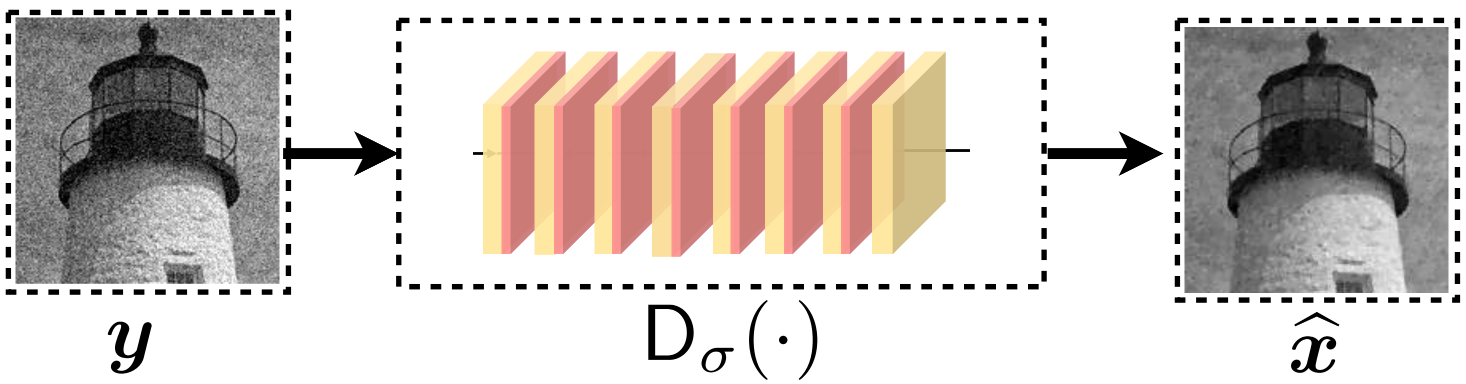
Build an **image denoiser** for AWGN (no info on forward model)



$\approx \mathsf{prox}_{\gamma h}(\boldsymbol{y})$

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

Build an image denoiser for AWGN (no info on forward model)

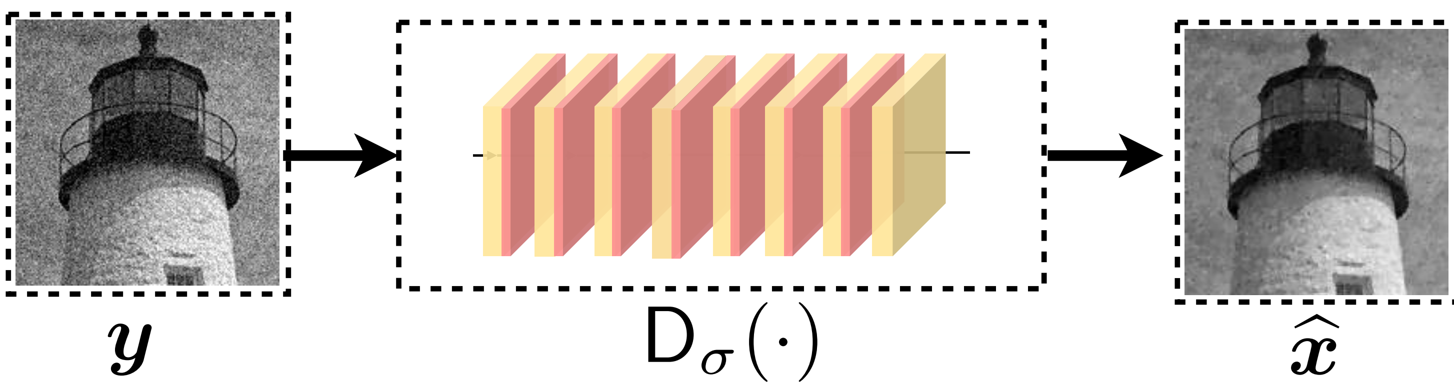


We define the following **"plug-and-play"** operators

- D_σ : more noisy image $\mapsto$ less noisy image
- $\mathsf{I} - \gamma \nabla g$: less data consistent $\mapsto$ more data consistent
- $\mathsf{prox}_{\gamma g}$: less data consistent $\mapsto$ more data consistent

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

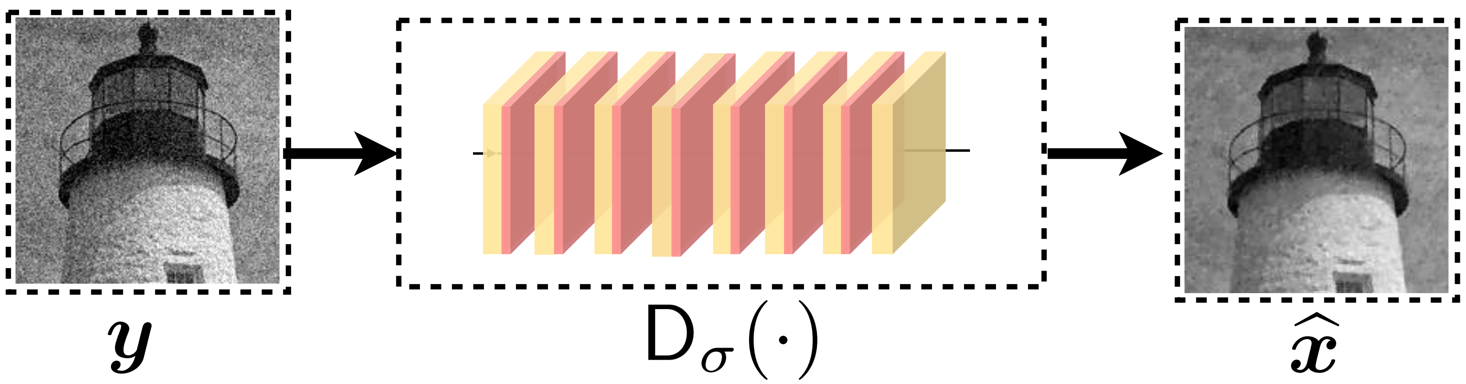
Build an image denoiser for AWGN (no info on forward model)



Combine operators into **Plug-and-Play Priors (PnP)** algorithms

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

Build an image denoiser for AWGN (no info on forward model)



Combine operators into Plug-and-Play Priors (PnP) algorithms

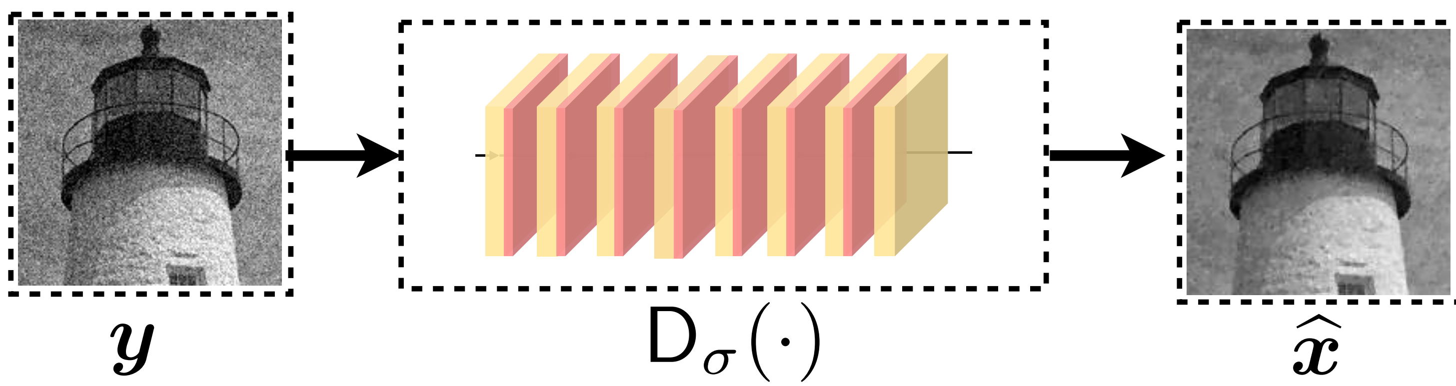
$$
\begin{aligned}
\boldsymbol{z}^k &\leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1}) \\
\boldsymbol{x}^k &\leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k + \boldsymbol{s}^{k-1}) \\
\boldsymbol{s}^k &\leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)
\end{aligned}
$$

PnP-ADMM

Venkatakrishnan *et al.*, "Plug-and-Play Priors for Model Based Reconstruction," *Proc. IEEE GlobalSIP*, pp. 945-948, December 2013.

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

Build an image denoiser for AWGN (no info on forward model)



Combine operators into Plug-and-Play Priors (PnP) algorithms

$$\begin{aligned}
\boldsymbol{z}^k &\leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1}) \\
\boldsymbol{x}^k &\leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k + \boldsymbol{s}^{k-1}) \\
\boldsymbol{s}^k &\leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)
\end{aligned}$$

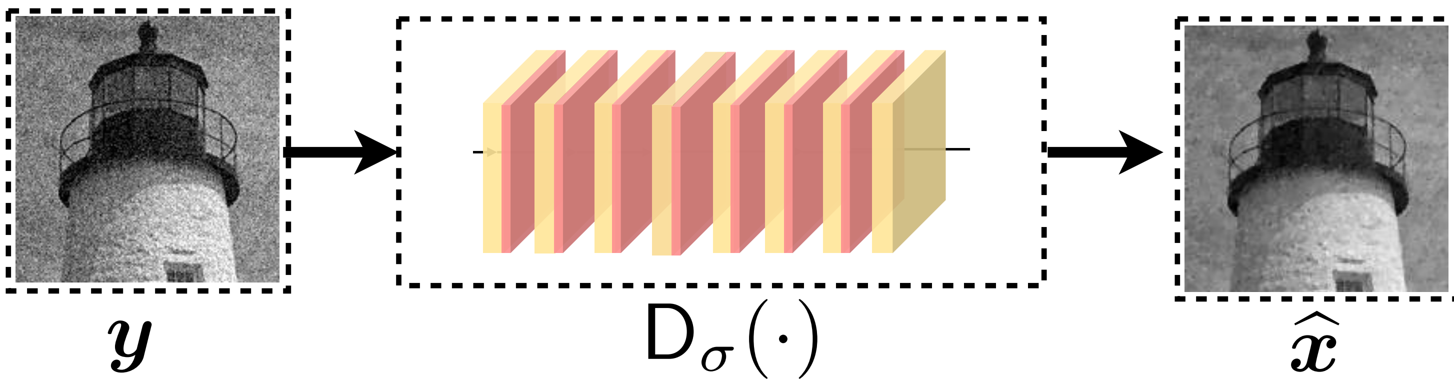
$$\begin{aligned}
\boldsymbol{z}^k &\leftarrow \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1}) \\
\boldsymbol{x}^k &\leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k) \\
\boldsymbol{s}^k &\leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})
\end{aligned}$$

PnP-FISTA

Kamilov *et al.*, “A Plug-and-Play Priors Approach for Solving Nonlinear Imaging Inverse Problems,” *IEEE Signal Process. Lett.*, 2017

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

Build an image denoiser for AWGN (no info on forward model)



Combine operators into Plug-and-Play Priors (PnP) algorithms

$$
\begin{aligned}
\boldsymbol{z}^k &\leftarrow \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} - \boldsymbol{s}^{k-1}) \\
\boldsymbol{x}^k &\leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k + \boldsymbol{s}^{k-1}) \\
\boldsymbol{s}^k &\leftarrow \boldsymbol{s}^{k-1} + (\boldsymbol{z}^k - \boldsymbol{x}^k)
\end{aligned}
$$

PnP-ADMM

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

PnP-FISTA

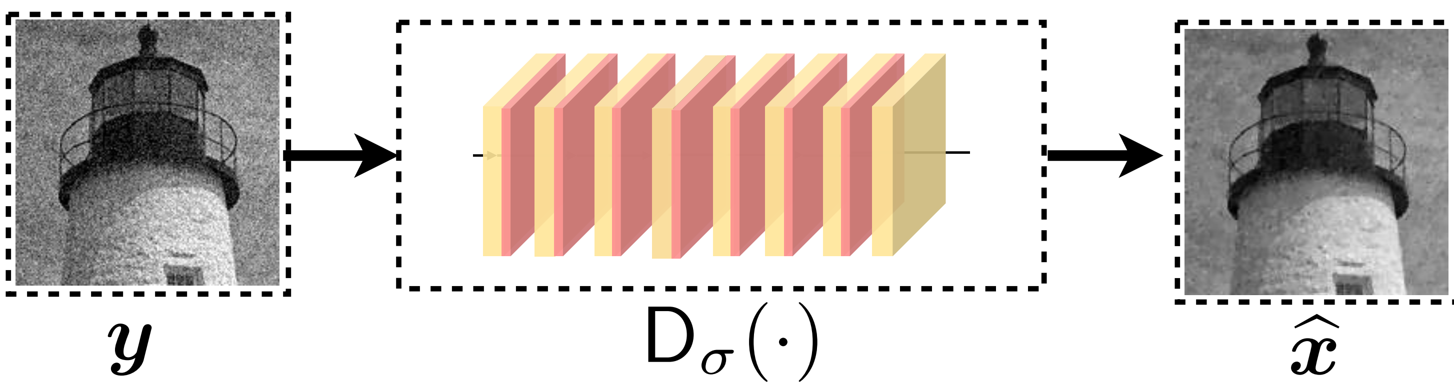
Venkatakrishnan *et al.*, "Plug-and-Play Priors for Model Based Reconstruction," *Proc. IEEE GlobalSIP*, pp. 945-948, December 2013.

Kamilov *et al.*, "A Plug-and-Play Priors Approach for Solving Nonlinear Imaging Inverse Problems," *IEEE Signal Process. Lett.*, 2017

Washington University in St. Louis

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

Build an image denoiser for AWGN (no info on forward model)



Combine operators into Plug-and-Play Priors (PnP) algorithms

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

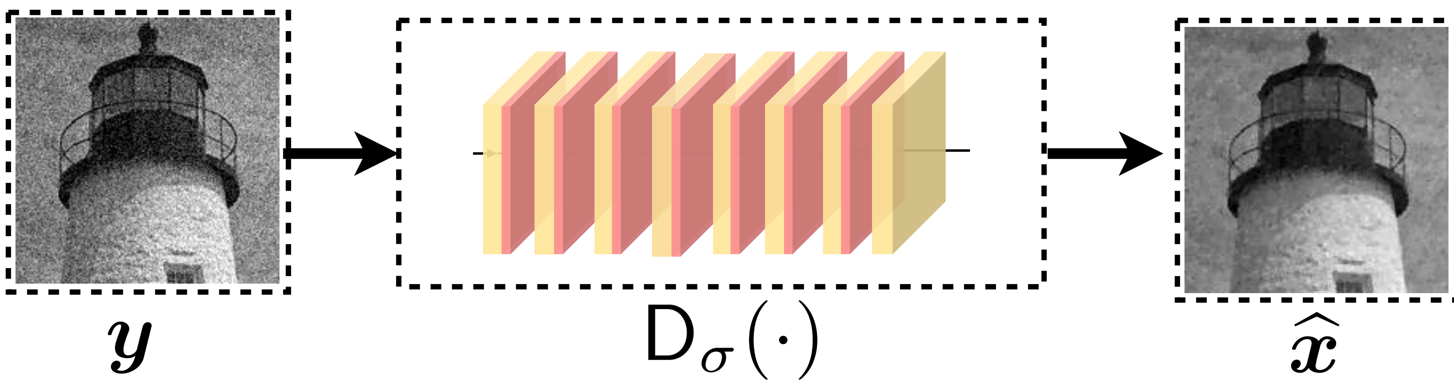
$$\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$$ forward model
$$\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k)$$ denoiser as a prior
$$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

Venkatakrishnan *et al.*, "Plug-and-Play Priors for Model Based Reconstruction," *Proc. IEEE GlobalSIP*, pp. 945-948, December 2013.

Kamilov *et al.*, "A Plug-and-Play Priors Approach for Solving Nonlinear Imaging Inverse Problems," *IEEE Signal Process. Lett.*, 2017

Plug-and-play priors (PnP) approach separates the forward model from the learned prior

Build an image denoiser for AWGN (no info on forward model)



Combine operators into Plug-and-Play Priors (PnP) algorithms

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

$$\boldsymbol{z}^k \leftarrow \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$$ forward model
$$\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k)$$ denoiser as a prior
$$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

PnP algorithms allow for an explicit separation between the forward model and the image prior

PnP might look heuristic, but it has a rigorous foundation in monotone operator theory

PnP might look heuristic, but it has a rigorous foundation in monotone operator theory

Consider the following set of **fixed points**

$$\mathsf{fix}(\mathsf{P}) \; := \; \{\boldsymbol{x} : \boldsymbol{x} = \mathsf{P}(\boldsymbol{x})\}$$

$$\mathsf{P}(\boldsymbol{x}) \; := \; \mathsf{D}_\sigma(\boldsymbol{x} - \gamma \nabla g(\boldsymbol{x}))$$

Sun *et al.*, “An Online Plug-and-Play Algorithm for Regularized Image Reconstruction,” *IEEE Trans. Comp. Imag*, 2019

PnP might look heuristic, but it has a rigorous foundation in monotone operator theory

Consider the following set of fixed points

$$\mathsf{fix}(\mathsf{P}) := \{\boldsymbol{x} : \boldsymbol{x} = \mathsf{P}(\boldsymbol{x})\} \qquad \mathsf{P}(\boldsymbol{x}) := \mathsf{D}_\sigma(\boldsymbol{x} - \gamma \nabla g(\boldsymbol{x}))$$

that satisfy the **consensus equilibrium (CE) equations**

$$\boxed{\begin{aligned} \boldsymbol{x}^* &= \mathsf{prox}_{\gamma g}(\boldsymbol{x}^* - \boldsymbol{u}) \\ \boldsymbol{x}^* &= \mathsf{D}_\sigma(\boldsymbol{x}^* + \boldsymbol{u}) \end{aligned}}$$

$$\boldsymbol{x}^* - \boldsymbol{u} \xrightarrow[\text{data}]{} \overset{\boldsymbol{x}^*}{\cdot} \xleftarrow[\text{prior}]{} \boldsymbol{x}^* + \boldsymbol{u}$$

Buzzard *et al.*, “Plug-and-Play Unplugged: Optimization free reconstruction using consensus equilibrium,” 2019

Sun *et al.*, “An Online Plug-and-Play Algorithm for Regularized Image Reconstruction,” *IEEE Trans. Comp. Imag*, 2019

PnP might look heuristic, but it has a rigorous foundation in monotone operator theory

Consider the following set of fixed points

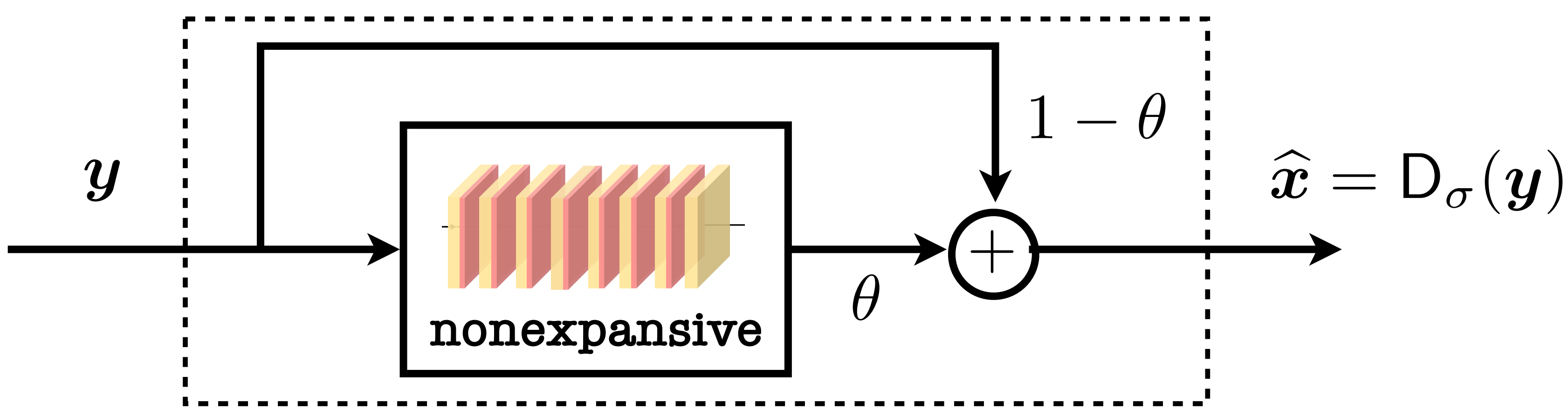
$$\mathsf{fix}(\mathsf{P}) := \{\boldsymbol{x} : \boldsymbol{x} = \mathsf{P}(\boldsymbol{x})\} \qquad \mathsf{P}(\boldsymbol{x}) := \mathsf{D}_\sigma(\boldsymbol{x} - \gamma \nabla g(\boldsymbol{x}))$$

that satisfy the consensus equilibrium (CE) equations

$$\boldsymbol{x}^* = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^* - \boldsymbol{u})$$
$$\boldsymbol{x}^* = \mathsf{D}_\sigma(\boldsymbol{x}^* + \boldsymbol{u})$$

$$\boldsymbol{x}^* - \boldsymbol{u} \xrightarrow[\text{data}]{} \boldsymbol{x}^* \xleftarrow[\text{prior}]{} \boldsymbol{x}^* + \boldsymbol{u}$$

For the analysis, we go beyond proximals to the class of **averaged denoisers** that can be trained using spectral normalization



Sun *et al.*, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," *IEEE Trans. Comp. Imag*, 2019

PnP might look heuristic, but it has a rigorous foundation in monotone operator theory

Proposition 1: Let $\mathsf{D}_\sigma(\cdot) = \mathsf{prox}_{\gamma h}(\cdot)$ for $\gamma, \sigma > 0$. Then, $\boldsymbol{x}^* \in \mathsf{fix}(\mathsf{P})$ if and only if it minimizes $f = g + h$.

1 – PnP is fully backward compatible with proximal algorithms

Sun *et al.*, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," *IEEE Trans. Comp. Imag*, 2019

PnP might look heuristic, but it has a rigorous foundation in monotone operator theory

Proposition 1: Let $\mathsf{D}_\sigma(\cdot) = \mathsf{prox}_{\gamma h}(\cdot)$ for $\gamma, \sigma > 0$. Then, $\boldsymbol{x}^* \in \mathsf{fix}(\mathsf{P})$ if and only if it minimizes $f = g + h$.

Proposition 2: Run PnP-ISTA for $t \geq 1$ iterations under a convex data-fidelity term and a θ-averaged denoiser with the step $\gamma \in (0, 1/L]$. Then, for $\boldsymbol{x}^* \in \mathsf{fix}(\mathsf{P})$

$$\frac{1}{t}\sum_{k=1}^{t} \|\boldsymbol{x}^{k-1} - \mathsf{P}(\boldsymbol{x}^{k-1})\|_2^2 \leq \frac{1}{t}\left(\frac{1+\theta}{1-\theta}\right)\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_2^2.$$

2 – It converges beyond proximal operator denoisers as O(1/t)

Sun *et al.*, “An Online Plug-and-Play Algorithm for Regularized Image Reconstruction,” *IEEE Trans. Comp. Imag*, 2019

Washington University in St. Louis

PnP might look heuristic, but it has a rigorous foundation in monotone operator theory

Proposition 1: Let $\mathsf{D}_\sigma(\cdot) = \mathsf{prox}_{\gamma h}(\cdot)$ for $\gamma, \sigma > 0$. Then, $\boldsymbol{x}^* \in \mathsf{fix}(\mathsf{P})$ if and only if it minimizes $f = g + h$.

Proposition 2: Run PnP-ISTA for $t \geq 1$ iterations under a convex data-fidelity term and a θ-averaged denoiser with the step $\gamma \in (0, 1/L]$. Then, for $\boldsymbol{x}^* \in \mathsf{fix}(\mathsf{P})$

$$\frac{1}{t}\sum_{k=1}^{t} \|\boldsymbol{x}^{k-1} - \mathsf{P}(\boldsymbol{x}^{k-1})\|_2^2 \leq \frac{1}{t}\left(\frac{1+\theta}{1-\theta}\right)\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_2^2.$$

Proposition 3: Under assumptions above, the set of fixed-points of PnP-ADMM coincides with $\mathsf{fix}(\mathsf{P})$.

3 – PnP-FISTA and PnP-ADMM have the same fixed points

Sun *et al.*, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," *IEEE Trans. Comp. Imag*, 2019

PnP might look heuristic, but it has a rigorous foundation in monotone operator theory

Proposition 1: Let $\mathsf{D}_\sigma(\cdot) = \mathsf{prox}_{\gamma h}(\cdot)$ for $\gamma, \sigma > 0$. Then, $\boldsymbol{x}^* \in \mathsf{fix}(\mathsf{P})$ if and only if it minimizes $f = g + h$.

Proposition 2: Run PnP-ISTA for $t \geq 1$ iterations under a convex data-fidelity term and a θ-averaged denoiser with the step $\gamma \in (0, 1/L]$. Then, for $\boldsymbol{x}^* \in \mathsf{fix}(\mathsf{P})$

$$\frac{1}{t}\sum_{k=1}^{t} \|\boldsymbol{x}^{k-1} - \mathsf{P}(\boldsymbol{x}^{k-1})\|_2^2 \leq \frac{1}{t}\left(\frac{1+\theta}{1-\theta}\right)\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_2^2.$$

Proposition 3: Under assumptions above, the set of fixed-points of PnP-ADMM coincides with $\mathsf{fix}(\mathsf{P})$.

Sun *et al.*, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," *IEEE Trans. Comp. Imag*, 2019

RED is an alternative to PnP that seeks to have an explicit regularizer for a given denoiser

Washington University in St. Louis

RED is an alternative to PnP that seeks to have an explicit regularizer for a given denoiser

Romano
(Stanford)

Elad
(Technion)

Milanfar
(Google)

Romano *et al.*, "The Little Engine That Could: Regularization by Denoising," *SIAM J. Imaging Sci.*, vol. 10, no. 4, 2017

RED is an alternative to PnP that seeks to have an explicit regularizer for a given denoiser

Consider the following GM-RED algorithm

$$\boxed{\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \gamma \mathsf{G}(\boldsymbol{x}^{t-1})}$$

"gradient" descent

$$\boxed{\mathsf{G}(\boldsymbol{x}) := \underbrace{\nabla g(\boldsymbol{x})}_{\text{data}} + \tau \underbrace{(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))}_{\text{prior}}}$$

Romano *et al.*, "The Little Engine That Could: Regularization by Denoising," *SIAM J. Imaging Sci.*, vol. 10, no. 4, 2017

RED is an alternative to PnP that seeks to have an explicit regularizer for a given denoiser

Consider the following GM-RED algorithm

$$\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \gamma \mathsf{G}(\boldsymbol{x}^{t-1}) \qquad \mathsf{G}(\boldsymbol{x}) := \nabla g(\boldsymbol{x}) + \tau(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))$$

For a locally homogeneous denoiser that has a symmetric Jacobian, GM-RED* solves the following problem

$$\boxed{\min_{\boldsymbol{x}} \{\underbrace{g(\boldsymbol{x})}_{\text{data}} + \underbrace{h_{\mathsf{red}}(\boldsymbol{x})}_{\text{prior}}\}} \qquad \boxed{h_{\mathsf{red}}(\boldsymbol{x}) := \frac{\tau}{2}\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))}$$

Romano *et al.*, "The Little Engine That Could: Regularization by Denoising," *SIAM J. Imaging Sci.*, vol. 10, no. 4, 2017

Reehorst *et al.*, "Regularization by Denoising: Clarifications and New Interpretations," *IEEE TCI*, vol. 5, no. 1, 2019

RED is an alternative to PnP that seeks to have an explicit regularizer for a given denoiser

Consider the following GM-RED algorithm

$$\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \gamma \mathsf{G}(\boldsymbol{x}^{t-1}) \qquad \mathsf{G}(\boldsymbol{x}) := \nabla g(\boldsymbol{x}) + \tau(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))$$

For a locally homogeneous denoiser that has a symmetric Jacobian, GM-RED* solves the following problem

$$\min_{\boldsymbol{x}} \{g(\boldsymbol{x}) + h_{\mathsf{red}}(\boldsymbol{x})\} \qquad h_{\mathsf{red}}(\boldsymbol{x}) := \frac{\tau}{2}\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))$$

This convex-optimization interpretation is powerful, but it also restricts the class of acceptable denoisers

Recall that one of the key conditions we assumed for the denoiser $f(\mathbf{x})$ is that it is homogeneous of degree 1,

$$f(c\mathbf{x}) = cf(\mathbf{x}). \tag{51}$$

This immediately notifies us that the $\rho(\mathbf{x})$ we wish to mimic in (50) must be 2-homogeneous

"Appendix A: Can we mimic any denoiser?"

Romano *et al.*, "The Little Engine That Could: Regularization by Denoising," *SIAM J. Imaging Sci.*, vol. 10, no. 4, 2017

Monotone operator theory provides an alternative interpretation to RED beyond convex optimization

Monotone operator theory provides an alternative interpretation to RED beyond convex optimization

Consider again the GM-RED algorithms

$$\boxed{\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \gamma \mathsf{G}(\boldsymbol{x}^{t-1})}$$

"gradient" descent

$$\boxed{\mathsf{G}(\boldsymbol{x}) := \underbrace{\nabla g(\boldsymbol{x})}_{\text{data}} + \tau\underbrace{(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))}_{\text{prior}}}$$

Monotone operator theory provides an alternative interpretation to RED beyond convex optimization

Consider again the GM-RED algorithms

$$\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \gamma \mathsf{G}(\boldsymbol{x}^{t-1}) \qquad\qquad \mathsf{G}(\boldsymbol{x}) \;:=\; \nabla g(\boldsymbol{x}) + \tau(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))$$

and the set of fixed points of G

$$\boxed{\mathsf{zer}(\mathsf{G}) \;:=\; \{\boldsymbol{x} : \mathsf{G}(\boldsymbol{x}) = \mathbf{0}\}}$$

a set of all images
where G is zero

Monotone operator theory provides an alternative interpretation to RED beyond convex optimization

Consider again the GM-RED algorithms

$$\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \gamma \mathsf{G}(\boldsymbol{x}^{t-1}) \qquad \mathsf{G}(\boldsymbol{x}) := \nabla g(\boldsymbol{x}) + \tau(\boldsymbol{x} - \mathsf{D}_\sigma(\boldsymbol{x}))$$

and the set of fixed points of G

$$\mathsf{zer}(\mathsf{G}) := \{\boldsymbol{x} : \mathsf{G}(\boldsymbol{x}) = \mathbf{0}\}$$

Just like PnP, RED leverages deep neural net denoisers without differentiating them, but gives an explicit control over data fidelity

If $\boldsymbol{x}^* \in \mathsf{zer}(\nabla g) \cap \mathsf{fix}(\mathsf{D})$, then $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{G})$

If $\mathsf{zer}(\nabla g) \cap \mathsf{fix}(\mathsf{D}) = \varnothing$, then $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{G})$ is a trade-off controlled by $\tau > 0$

Sun *et al.*, "Block Coordinate Regularization by Denoising," *Proc. NeurIPS 2019* (Vancouver, Canada, December 8-14)

The theoretical convergence of GM-RED to its fixed points can be fully characterized

The theoretical convergence of GM-RED to its fixed points can be fully characterized

Theorem 1. Run GM-RED for $t \geq 1$ iterations under Assumptions 1-3 using a fixed step-size $0 < \gamma \leq 1/(L + 2\tau)$. Then, we have

$$\|\mathsf{G}(\boldsymbol{x}^{t-1})\|_2^2 \leq \frac{(L + 2\tau)}{\gamma t} R_0^2$$

1 – GM-RED converge to zer(G) as O(1/t)

Sun *et al.*, "Block Coordinate Regularization by Denoising," *Proc. NeurIPS 2019* (Vancouver, Canada, December 8-14)

Liu *et al.*, "Infusing Learned Priors into Model-Based Multispectral Imaging," *Proc. CAMSAP* 2019

Washington University in St. Louis

The theoretical convergence of GM-RED to its fixed points can be fully characterized

Theorem 1. Run GM-RED for $t \geq 1$ iterations under Assumptions 1-3 using a fixed step-size $0 < \gamma \leq 1/(L + 2\tau)$. Then, we have

$$\|\mathsf{G}(\boldsymbol{x}^{t-1})\|_2^2 \leq \frac{(L+2\tau)}{\gamma t} R_0^2$$

Theorem 2. Run GM-RED for $t \geq 1$ iterations with a proximal operator denoiser under Assumptions 1-4 using a fixed step-size $0 < \gamma \leq 1/(L + 2\tau)$. Then, we have

$$f(\boldsymbol{x}^t) - f^* \leq \frac{2}{\gamma t} R_0^2 + \frac{G_0^2}{2\tau},$$

where the function $f = g + h$ and f^* is its minimum.

2 – GM-RED is backward compatible with proximal optimization

Sun *et al.*, "Block Coordinate Regularization by Denoising," *Proc. NeurIPS 2019* (Vancouver, Canada, December 8-14)

Liu *et al.*, "Infusing Learned Priors into Model-Based Multispectral Imaging," *Proc. CAMSAP* 2019

The theoretical convergence of GM-RED to its fixed points can be fully characterized

Theorem 1. Run GM-RED for $t \geq 1$ iterations under Assumptions 1-3 using a fixed step-size $0 < \gamma \leq 1/(L + 2\tau)$. Then, we have

$$\|\mathsf{G}(\boldsymbol{x}^{t-1})\|_2^2 \leq \frac{(L+2\tau)}{\gamma t} R_0^2$$

Theorem 2. Run GM-RED for $t \geq 1$ iterations with a proximal operator denoiser under Assumptions 1-4 using a fixed step-size $0 < \gamma \leq 1/(L + 2\tau)$. Then, we have

$$f(\boldsymbol{x}^t) - f^* \leq \frac{2}{\gamma t} R_0^2 + \frac{G_0^2}{2\tau},$$

where the function $f = g + h$ and f^* is its minimum.

Sun *et al.*, "Block Coordinate Regularization by Denoising," *Proc. NeurIPS 2019* (Vancouver, Canada, December 8-14)

Liu *et al.*, "Infusing Learned Priors into Model-Based Multispectral Imaging," *Proc. CAMSAP* 2019

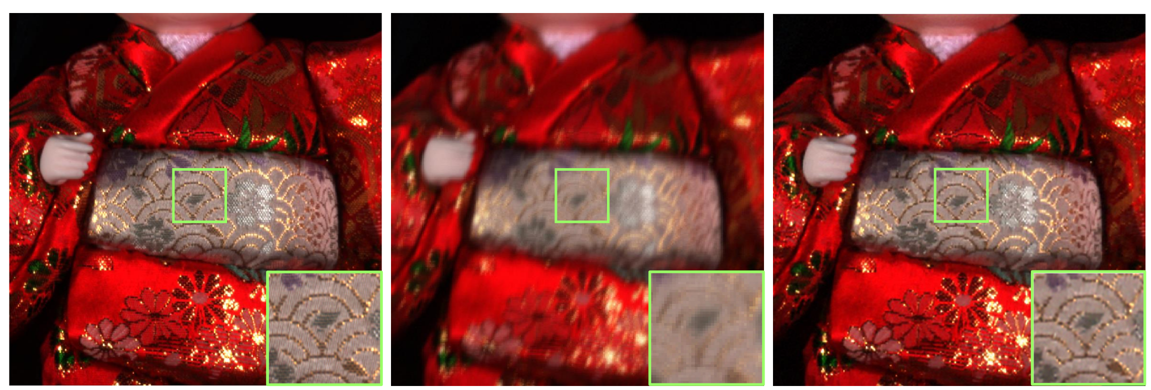
GM-RED with denoising neural net priors can significantly outperform traditional priors

Liu *et al.*, "Infusing Learned Priors into Model-Based Multispectral Imaging," *Proc. CAMSAP* 2019

GM-RED with denoising neural net priors can significantly outperform traditional priors

Table 1. Summary of average SNR performance of different methods for multispectral image super-resolution.

Kernel	Scale	3DTV	BM3D	DnCNN*
Gaussian 7×7	$\times 2$	24.01	24.67	25.20
	$\times 3$	22.74	23.33	23.43
	$\times 4$	20.41	20.94	21.03
Motion blur 19×19	$\times 2$	23.30	24.49	25.21
	$\times 3$	20.17	20.85	21.01
	$\times 4$	18.86	19.34	19.43



Ground-truth Zoomed LR image RED-DnCNN* (22.25dB)

Ground-truth Zoomed LR image RED-DnCNN* (31.59dB)

Liu *et al.*, "Infusing Learned Priors into Model-Based Multispectral Imaging," *Proc. CAMSAP* 2019

Today we will talk about

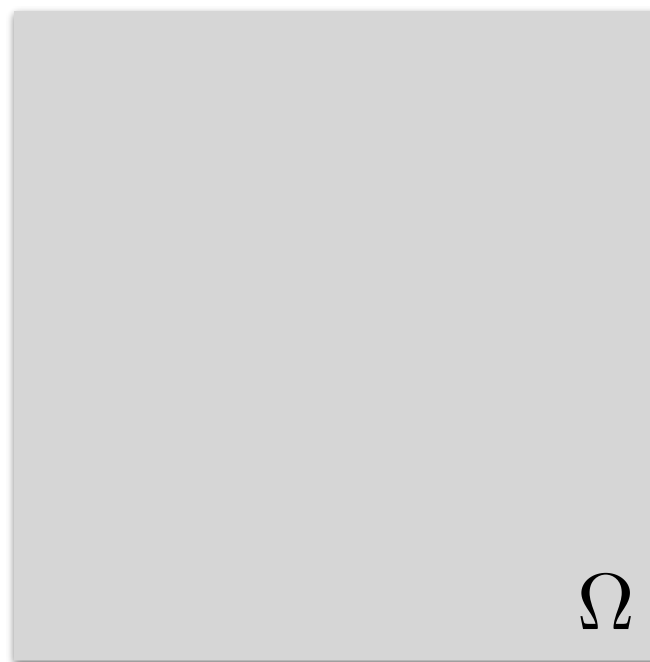
- Imaging as an inverse problem
  From forward models to regularized inversion

- Combining models and learning for imaging
  From PnP to RED using denoising neural nets

- **Scaling image formation to large problems**
  Variants: PnP-SGD, On-RED, and BC-RED

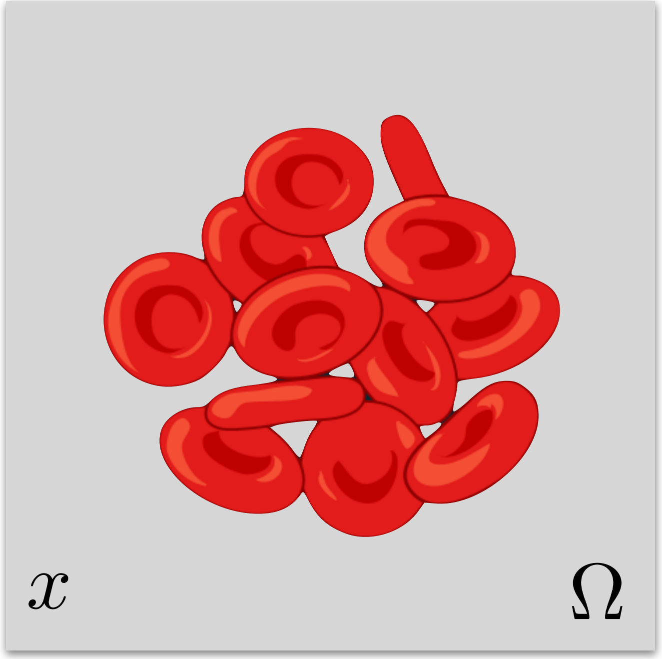
Motivation: optical tomographic microscopy replaces x-rays with the visible light

Motivation: optical tomographic microscopy replaces x-rays with the visible light



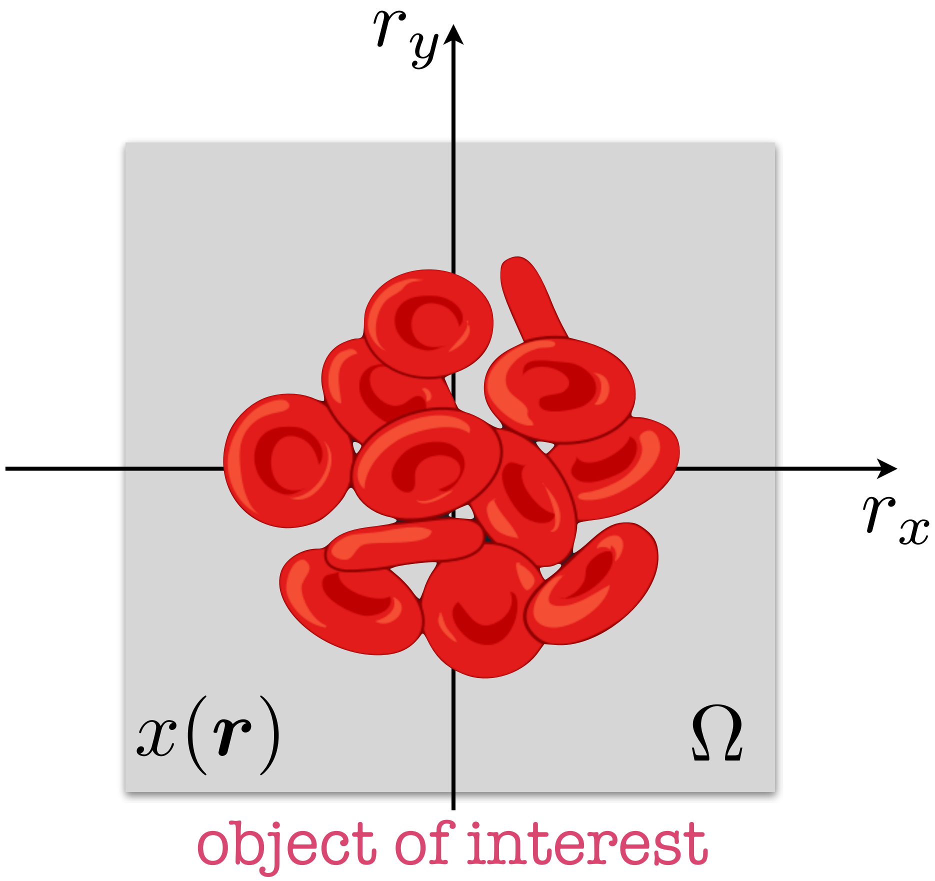
volume of interest

Motivation: optical tomographic microscopy replaces x-rays with the visible light

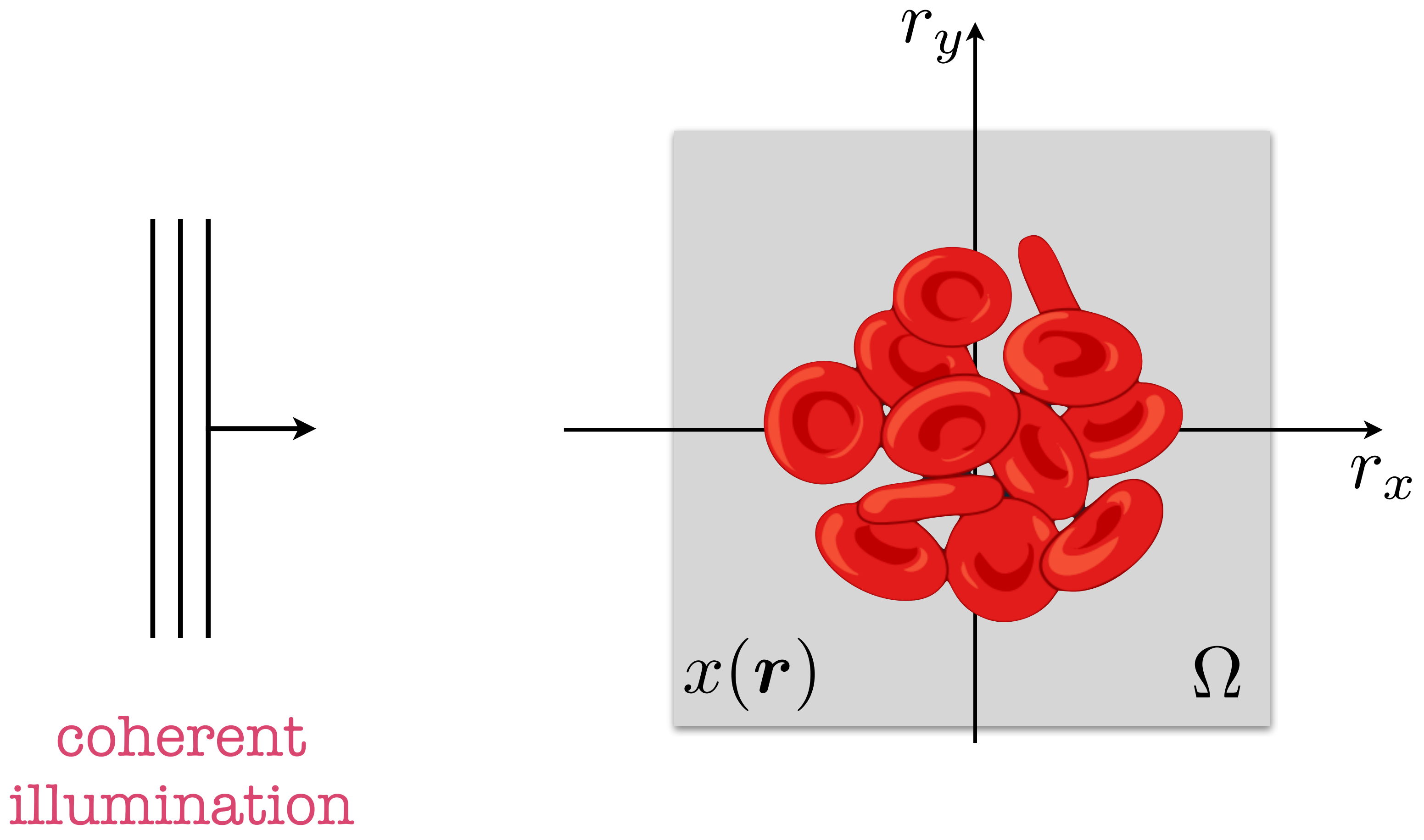


object of interest

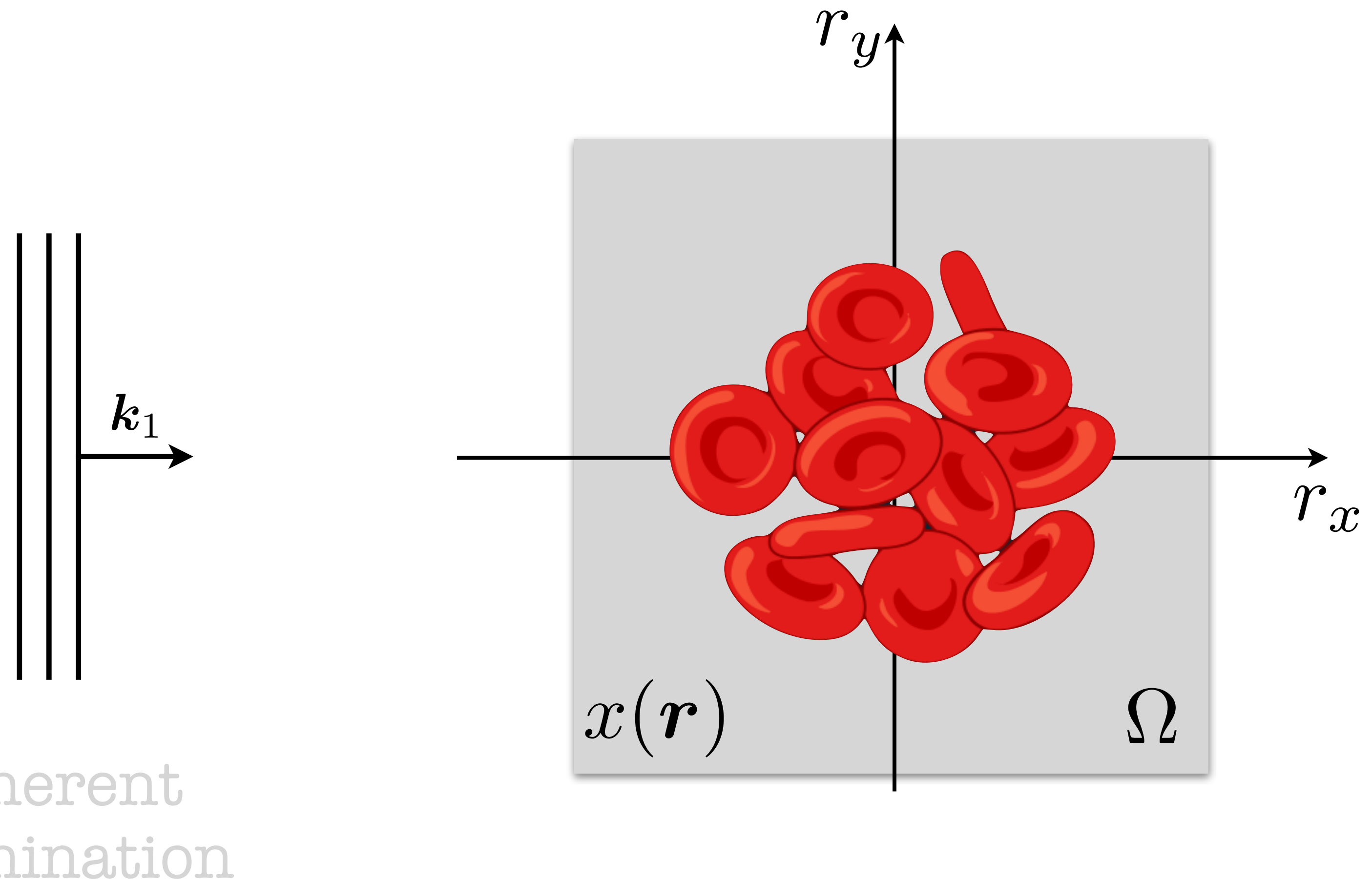
Motivation: optical tomographic microscopy replaces x-rays with the visible light



Motivation: optical tomographic microscopy replaces x-rays with the visible light

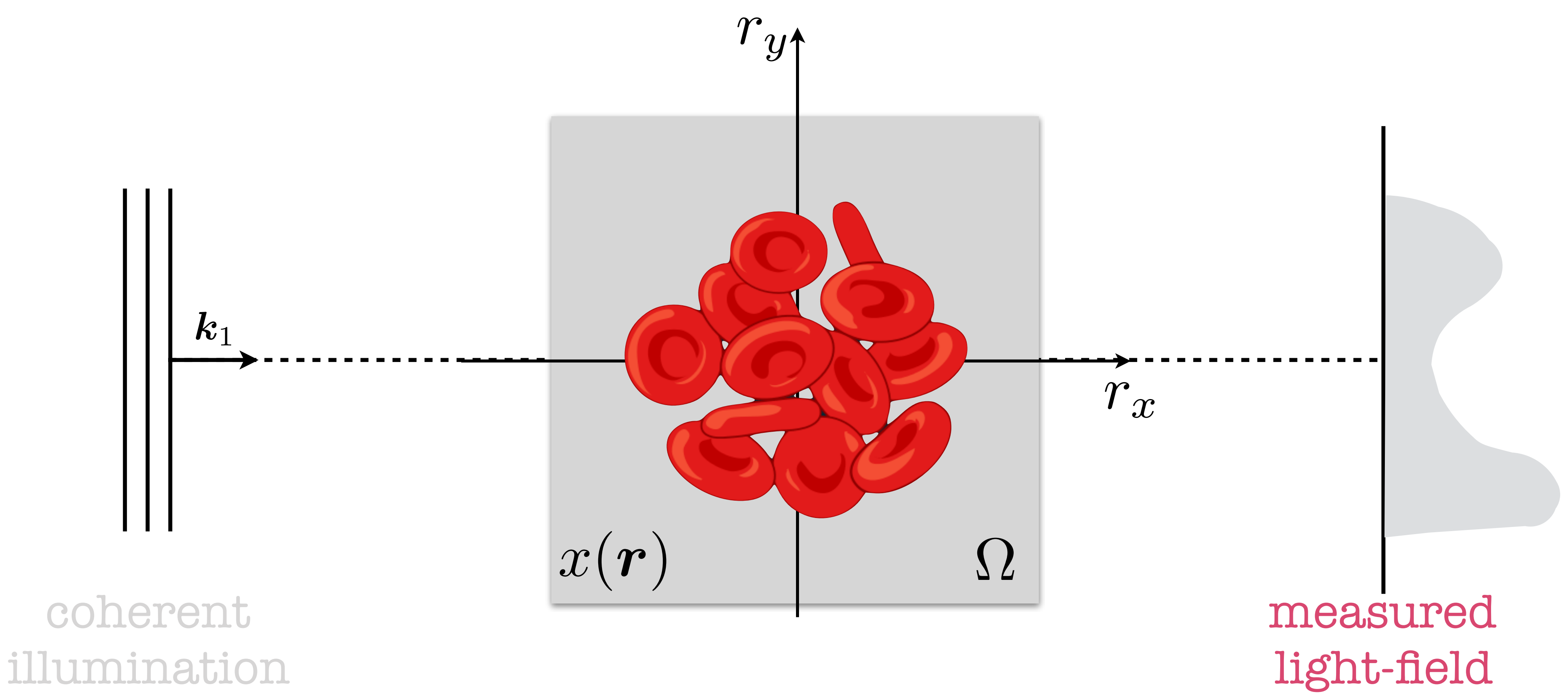


Motivation: optical tomographic microscopy replaces x-rays with the visible light



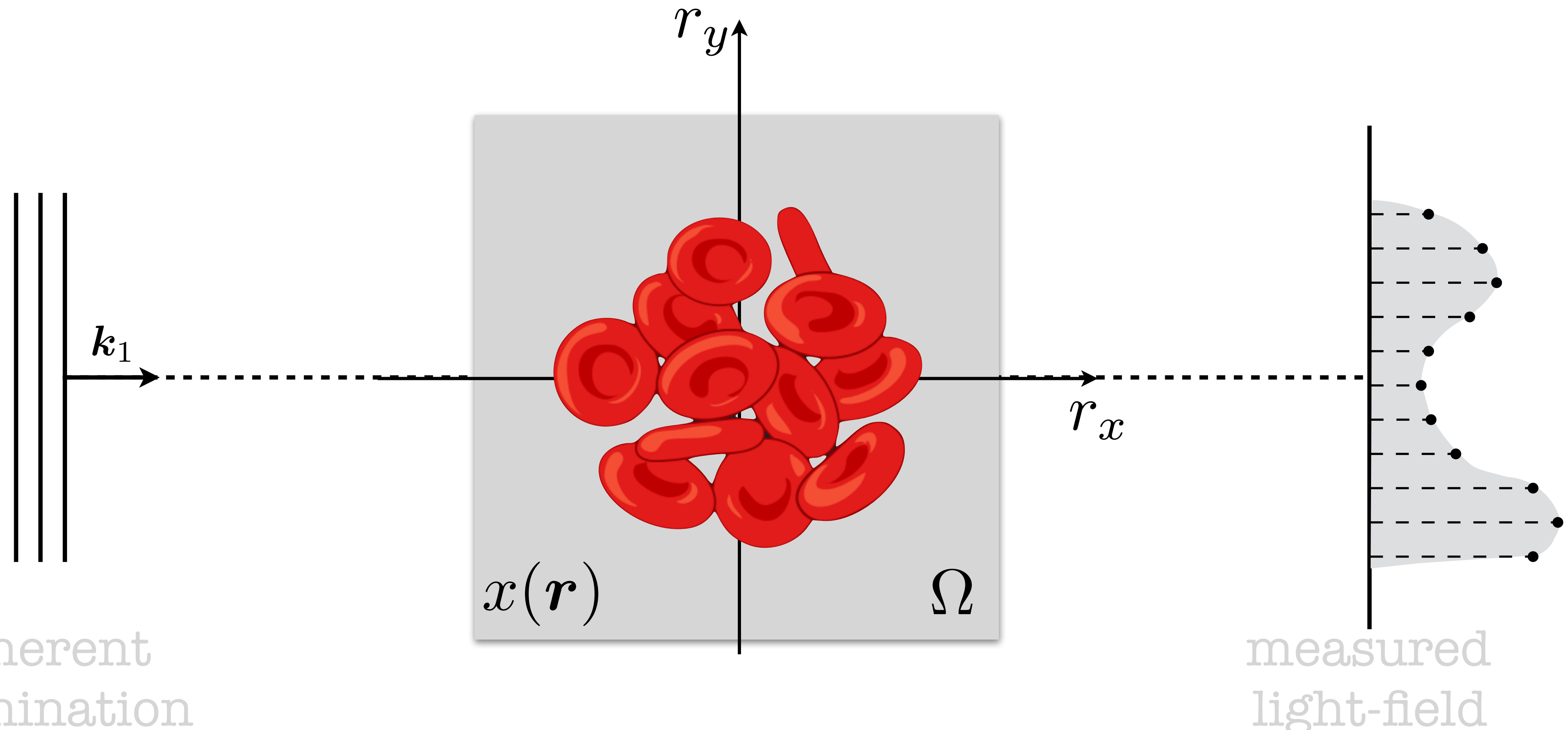
$$u_{\text{in}}^1(\boldsymbol{r}) = \mathrm{e}^{\mathrm{j}\langle \boldsymbol{k}_1, \boldsymbol{r} \rangle}$$

Motivation: optical tomographic microscopy replaces x-rays with the visible light



$$u_{\text{in}}^1(\boldsymbol{r}) = \mathrm{e}^{\mathrm{j}\langle \boldsymbol{k}_1, \boldsymbol{r} \rangle}$$

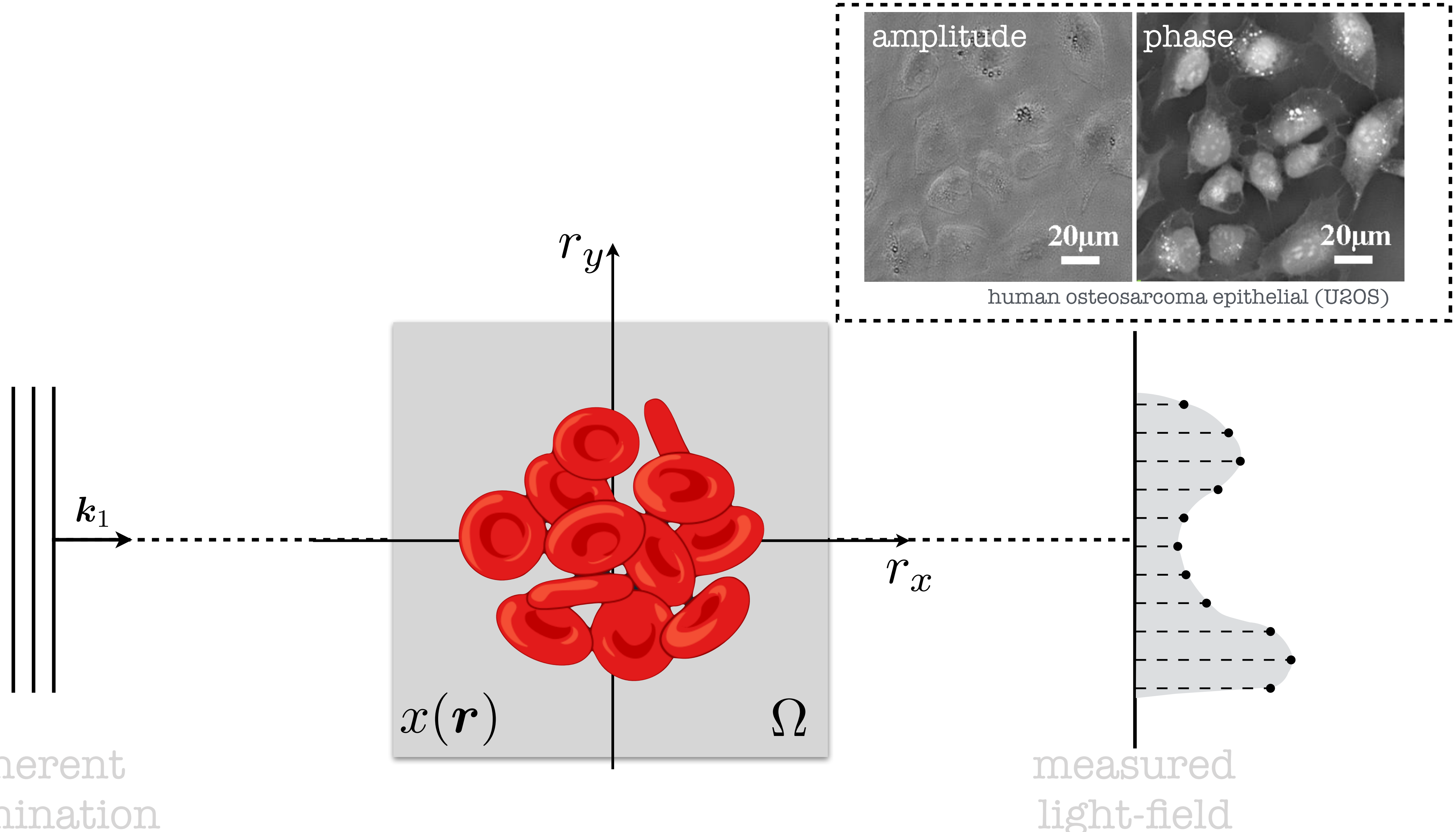
Motivation: optical tomographic microscopy replaces x-rays with the visible light



$$u_{\mathsf{in}}^{1}(\boldsymbol{r}) = \mathsf{e}^{\mathsf{j}\langle \boldsymbol{k}_1, \boldsymbol{r}\rangle}$$

$$\boldsymbol{y}_1$$

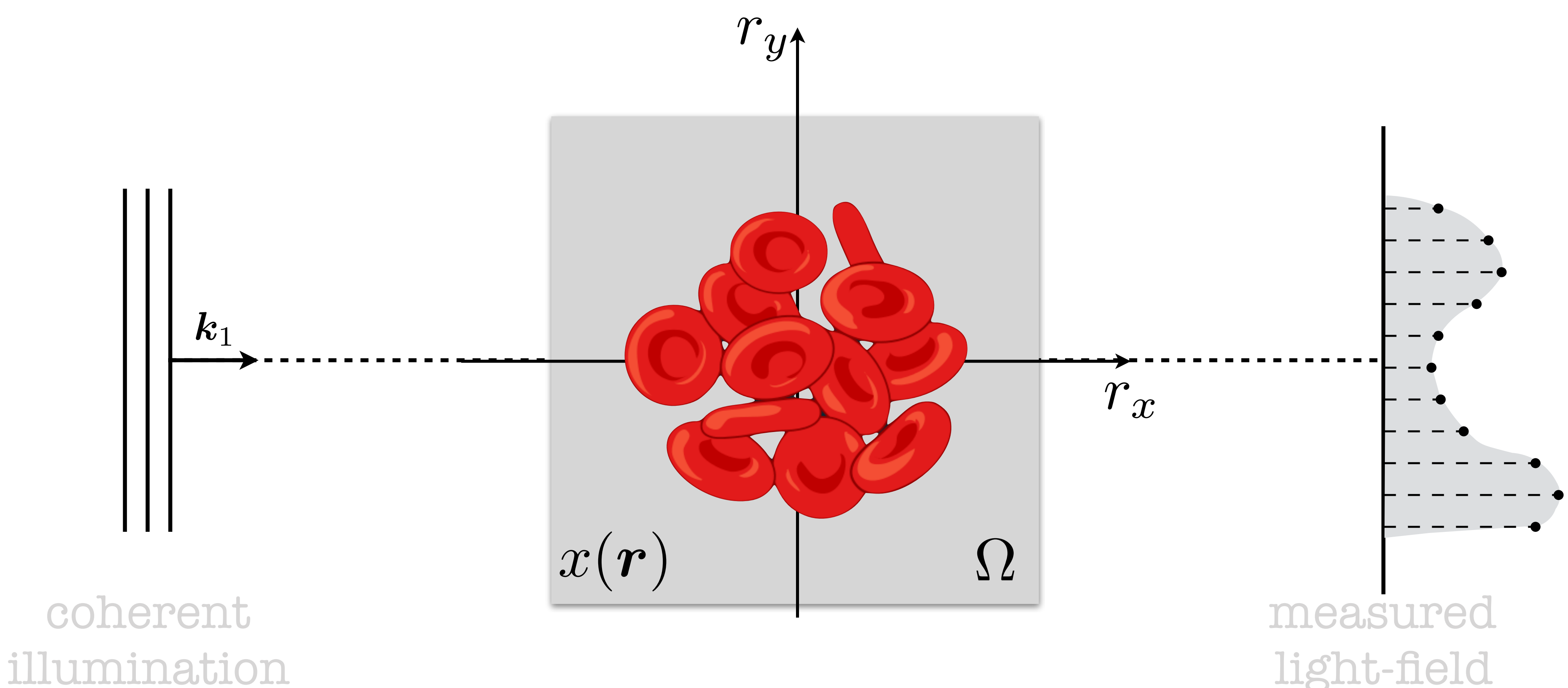
Motivation: optical tomographic microscopy replaces x-rays with the visible light



$$u_{\mathsf{in}}^1(\boldsymbol{r}) = \mathsf{e}^{\mathsf{j}\langle \boldsymbol{k}_1, \boldsymbol{r}\rangle}$$

$$\boldsymbol{y}_1$$

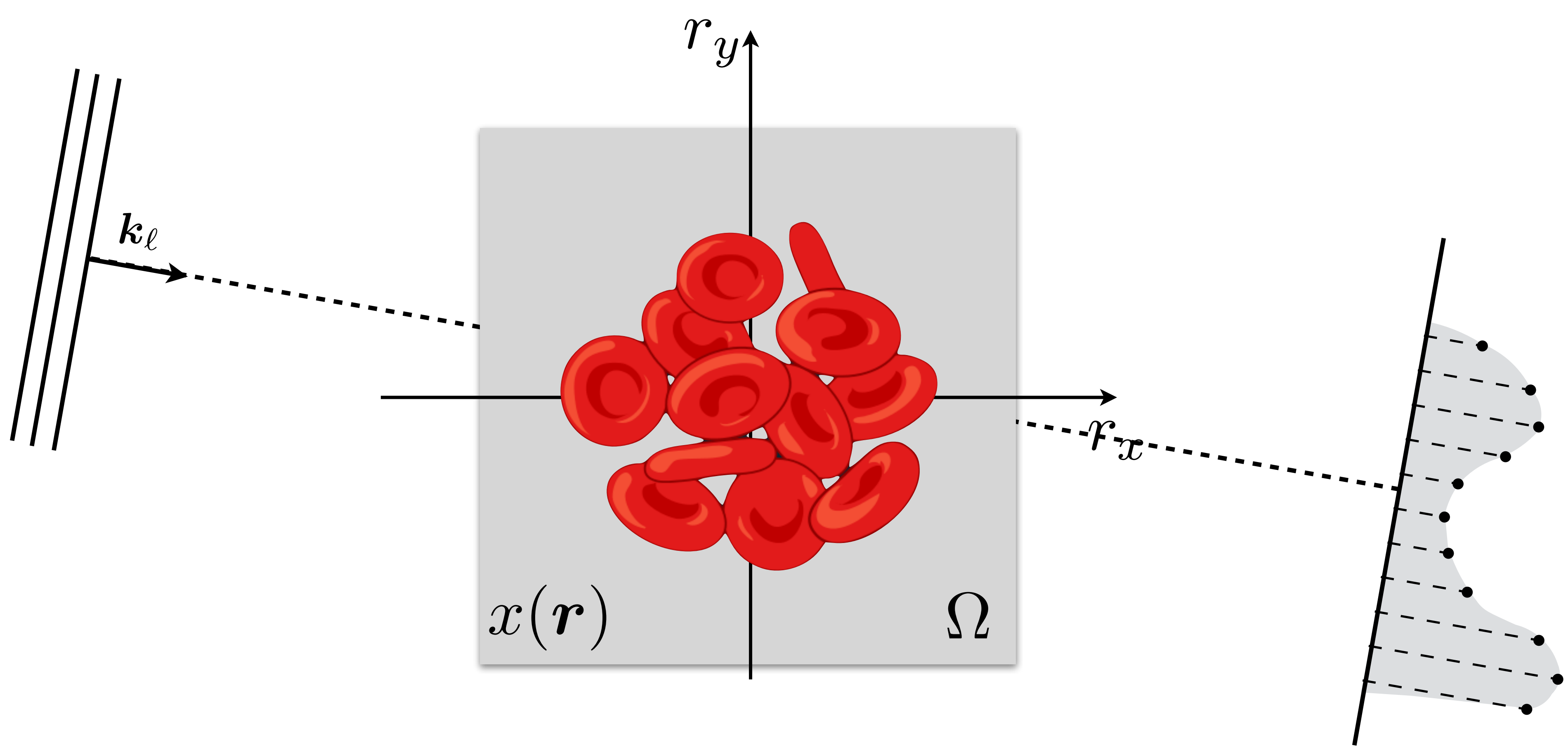
Motivation: optical tomographic microscopy replaces x-rays with the visible light



$$u_{\mathrm{in}}^{1}(\boldsymbol{r}) = \mathrm{e}^{\mathrm{j}\langle \boldsymbol{k}_1, \boldsymbol{r} \rangle}$$

$$\boldsymbol{y}_1$$

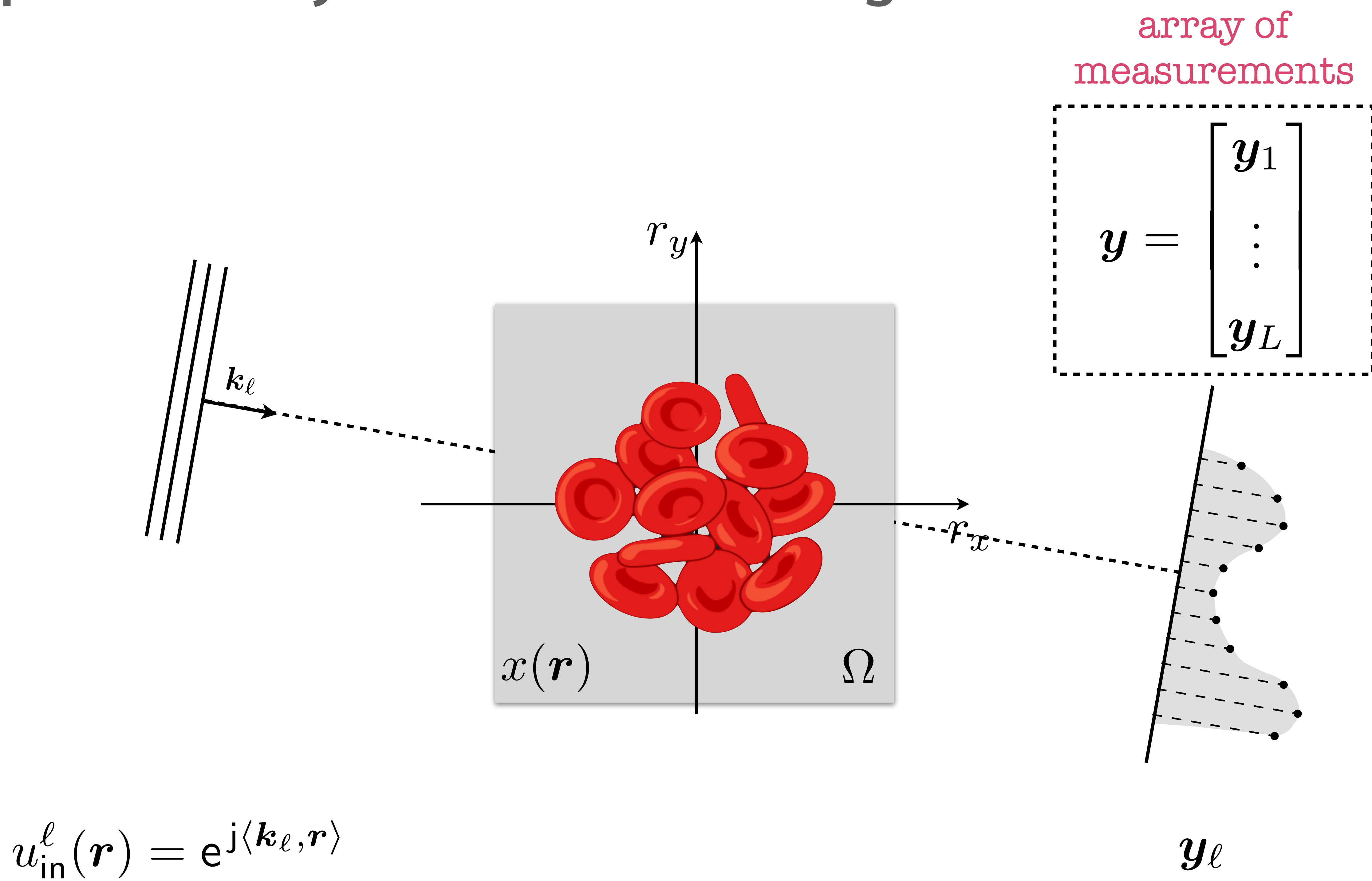
Motivation: optical tomographic microscopy replaces x-rays with the visible light



$$u^\ell_{\text{in}}(\boldsymbol{r}) = \mathrm{e}^{\mathrm{j}\langle \boldsymbol{k}_\ell, \boldsymbol{r}\rangle}$$

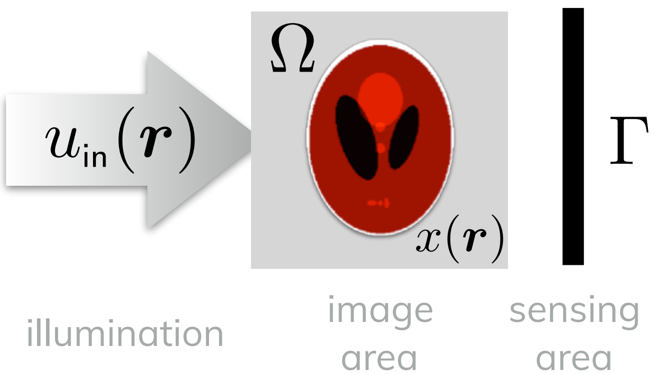
$$\boldsymbol{y}_\ell$$

Motivation: optical tomographic microscopy replaces x-rays with the visible light



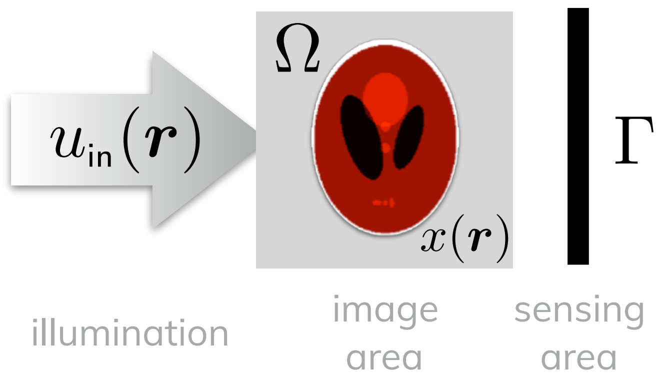
For each illumination the forward model is the (approximate) solution of the wave equation

For each illumination the forward model is the (approximate) solution of the wave equation



For each illumination the forward model is the (approximate) solution of the wave equation

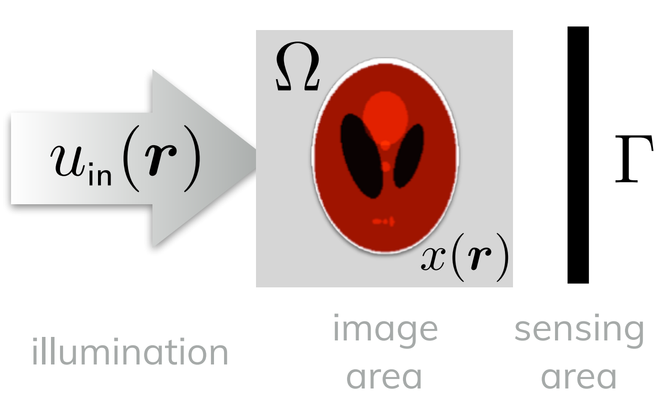
The **Helmholtz equation** for modeling object-light interactions



For each illumination the forward model is the (approximate) solution of the wave equation

The **Helmholtz equation** for modeling object-light interactions

$$\boxed{(\Delta + k_{\mathrm{b}}^2 \mathrm{I}) u_{\mathsf{sc}}(\boldsymbol{r}) = x(\boldsymbol{r}) u(\boldsymbol{r})} \qquad u(\boldsymbol{r}) = u_{\mathsf{in}}(\boldsymbol{r}) + u_{\mathsf{sc}}(\boldsymbol{r})$$

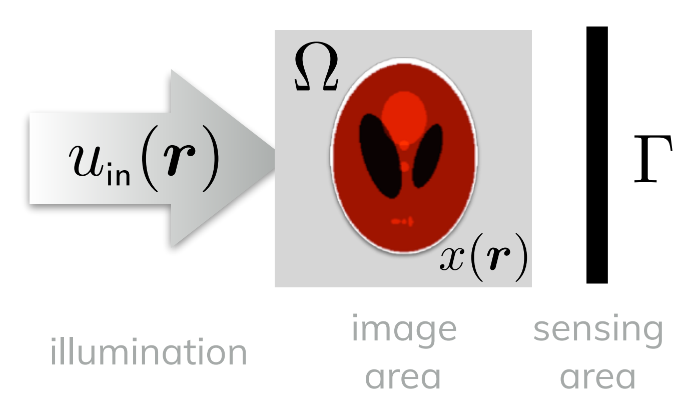


For each illumination the forward model is the (approximate) solution of the wave equation

The Helmholtz equation for modeling object-light interactions

$$(\Delta + k_{\mathrm{b}}^2 \mathrm{I}) u_{\mathsf{sc}}(\boldsymbol{r}) = x(\boldsymbol{r}) u(\boldsymbol{r}) \qquad u(\boldsymbol{r}) = u_{\mathsf{in}}(\boldsymbol{r}) + u_{\mathsf{sc}}(\boldsymbol{r})$$

The **Domain-integral formulation** with the Green's function



For each illumination the forward model is the (approximate) solution of the wave equation

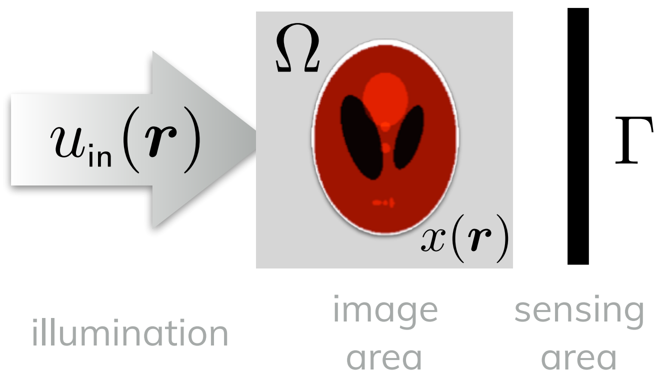
The Helmholtz equation for modeling object-light interactions

$$(\Delta + k_{\mathrm{b}}^2 \mathrm{I}) u_{\mathsf{sc}}(\boldsymbol{r}) = x(\boldsymbol{r}) u(\boldsymbol{r}) \qquad u(\boldsymbol{r}) = u_{\mathsf{in}}(\boldsymbol{r}) + u_{\mathsf{sc}}(\boldsymbol{r})$$

The Domain-integral formulation with the Green's function

$$\boxed{u_{\mathsf{sc}}(\boldsymbol{r}) = \int_{\Omega} g(\boldsymbol{r} - \boldsymbol{r}')\, x(\boldsymbol{r}')\, u(\boldsymbol{r}')\, \mathsf{d}\boldsymbol{r}} \qquad g(\boldsymbol{r}) \;:=\; \frac{\mathsf{e}^{\mathsf{j}k_{\mathsf{b}}\|\boldsymbol{r}\|}}{4\pi\|\boldsymbol{r}\|}$$

nonlinear object-light coupling



For each illumination the forward model is the (approximate) solution of the wave equation

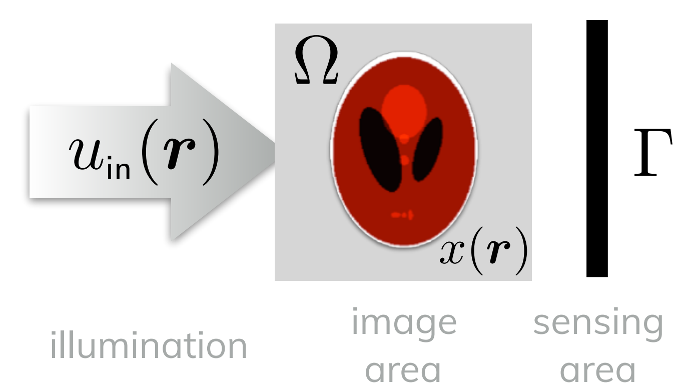
The Helmholtz equation for modeling object-light interactions

$$(\Delta + k_{\mathrm{b}}^2 \mathrm{I}) u_{\mathsf{sc}}(\boldsymbol{r}) = x(\boldsymbol{r}) u(\boldsymbol{r}) \qquad u(\boldsymbol{r}) = u_{\mathsf{in}}(\boldsymbol{r}) + u_{\mathsf{sc}}(\boldsymbol{r})$$

The Domain-integral formulation with the Green's function

$$u_{\mathsf{sc}}(\boldsymbol{r}) = \int_{\Omega} g(\boldsymbol{r} - \boldsymbol{r}')\, x(\boldsymbol{r}')\, \boxed{u(\boldsymbol{r}')}\, \mathsf{d}\boldsymbol{r} \qquad g(\boldsymbol{r}) := \frac{\mathsf{e}^{\mathsf{j} k_{\mathsf{b}} \|\boldsymbol{r}\|}}{4\pi \|\boldsymbol{r}\|}$$

accounts for
multi scattering



For each illumination the forward model is the (approximate) solution of the wave equation

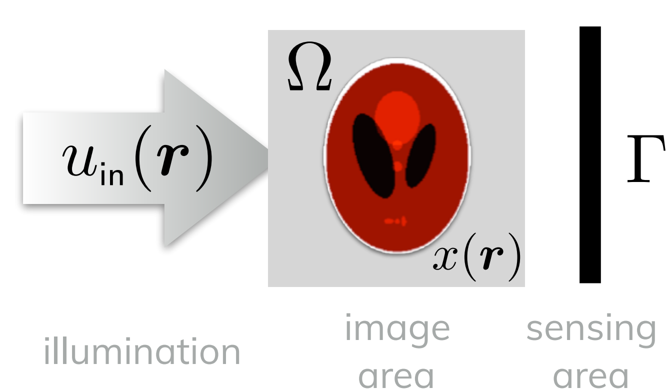
The Helmholtz equation for modeling object-light interactions

$$(\Delta + k_\mathrm{b}^2 \mathrm{I}) u_\mathsf{sc}(\boldsymbol{r}) = x(\boldsymbol{r}) u(\boldsymbol{r}) \qquad u(\boldsymbol{r}) = u_\mathsf{in}(\boldsymbol{r}) + u_\mathsf{sc}(\boldsymbol{r})$$

The Domain-integral formulation with the Green's function

$$u_\mathsf{sc}(\boldsymbol{r}) = \int_\Omega g(\boldsymbol{r} - \boldsymbol{r}')\, x(\boldsymbol{r}')\, u(\boldsymbol{r}')\, \mathrm{d}\boldsymbol{r} \qquad g(\boldsymbol{r}) := \frac{\mathrm{e}^{\mathrm{j}k_\mathrm{b}\|\boldsymbol{r}\|}}{4\pi\|\boldsymbol{r}\|}$$

There are many approximations to solve this equation



For each illumination the forward model is the (approximate) solution of the wave equation

The Helmholtz equation for modeling object-light interactions

$$(\Delta + k_{\mathrm{b}}^2 \mathrm{I}) u_{\mathrm{sc}}(\boldsymbol{r}) = x(\boldsymbol{r}) u(\boldsymbol{r}) \qquad u(\boldsymbol{r}) = u_{\mathrm{in}}(\boldsymbol{r}) + u_{\mathrm{sc}}(\boldsymbol{r})$$

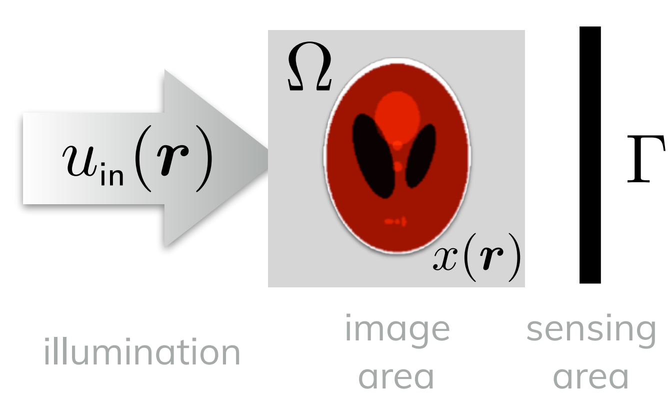
The Domain-integral formulation with the Green's function

$$u_{\mathrm{sc}}(\boldsymbol{r}) = \int_{\Omega} g(\boldsymbol{r} - \boldsymbol{r}')\, x(\boldsymbol{r}')\, u(\boldsymbol{r}')\, \mathrm{d}\boldsymbol{r} \qquad g(\boldsymbol{r}) := \frac{\mathrm{e}^{\mathrm{j}k_{\mathrm{b}}\|\boldsymbol{r}\|}}{4\pi\|\boldsymbol{r}\|}$$

There are many approximations to solve this equation

$$u_{\mathrm{sc}}(\boldsymbol{r}) \approx \mathrm{H}\{x, u_{\mathrm{in}}\}(\boldsymbol{r})$$

(for each illumination)



For many imaging inverse problems, the traditional batch data processing is suboptimal

For many imaging inverse problems, the traditional batch data processing is suboptimal

Consider the data-fidelity of the following form

$$\boxed{g(\boldsymbol{x}) = \frac{1}{L}\sum_{\ell=1}^{L} g_\ell(\boldsymbol{x}) = \frac{1}{2L}\sum_{\ell=1}^{L} \|\boldsymbol{y} - \mathbf{H}_\ell(\boldsymbol{x})\|_2^2}$$

Sum of L data-fidelity terms

Washington University in St. Louis

For many imaging inverse problems, the traditional batch data processing is suboptimal

Consider the data-fidelity of the following form

$$g(\boldsymbol{x}) = \frac{1}{L}\sum_{\ell=1}^{L} g_\ell(\boldsymbol{x}) = \frac{1}{2L}\sum_{\ell=1}^{L} \|\boldsymbol{y} - \mathbf{H}_\ell(\boldsymbol{x})\|_2^2$$

Consider the complexity of computing
the **full-batch gradient** and the **mini-batch gradient**

$$\boxed{\nabla g(\boldsymbol{x}) = \frac{1}{L}\sum_{\ell=1}^{L} \nabla g_\ell(\boldsymbol{x})} \quad \text{vs.} \quad \boxed{\hat{\nabla} g(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} \nabla g_{\ell_b}(\boldsymbol{x})}$$

Uses all the measurements

Uses a subset of measurements

PnP-SGD enables to scale PnP-FISTA to a large number of tomographic measurements

PnP-SGD enables to scale PnP-FISTA to a large number of tomographic measurements

PnP-FISTA

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

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

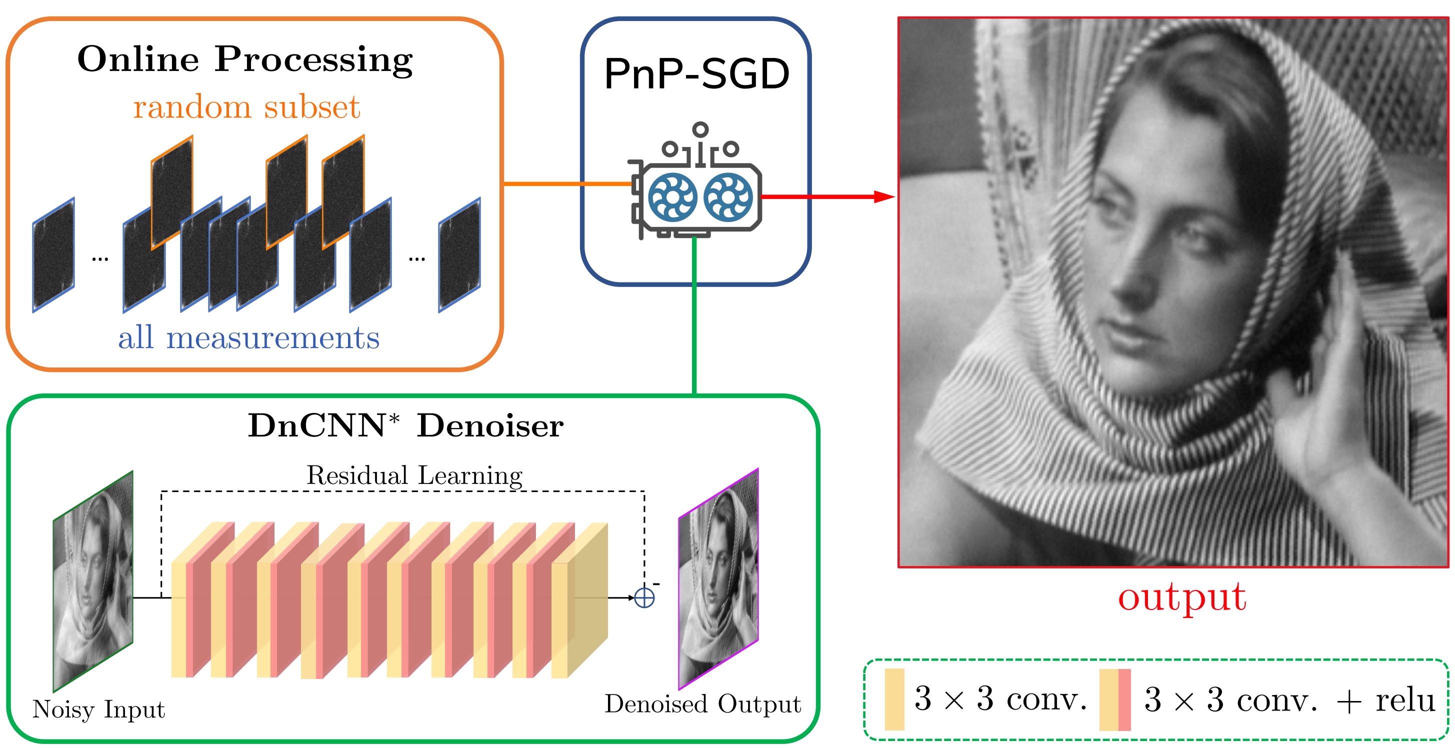
$$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

PnP-SGD

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

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

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



PnP-SGD uses only B << L measurements per iteration

Sun *et al.*, "An Online Plug-and-Play Algorithm for Regularized Image Reconstruction," *IEEE Trans. Comp. Imag*, 2019

PnP-SGD enables to scale PnP-FISTA to a large number of tomographic measurements

PnP-FISTA

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

PnP-SGD

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

Theorem 1. Run PnP-SGD for $t \geq 1$ iterations under Assumptions 1-3 using a fixed step-size $0 < \gamma \leq 1/L$ and a fixed minibatch size $B = t$. Then, we have

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=1}^{t} \|\boldsymbol{x}^{k-1} - \mathsf{P}(\boldsymbol{x}^{k-1})\|_2^2\right] \leq \frac{C}{\sqrt{t}},$$

where $C > 0$ is a constant.

Fixed point convergence similar to the traditional SGD

Sun *et al.*, “An Online Plug-and-Play Algorithm for Regularized Image Reconstruction,” *IEEE Trans. Comp. Imag*, 2019

PnP-SGD enables to scale PnP-FISTA to a large number of tomographic measurements

PnP-FISTA

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

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

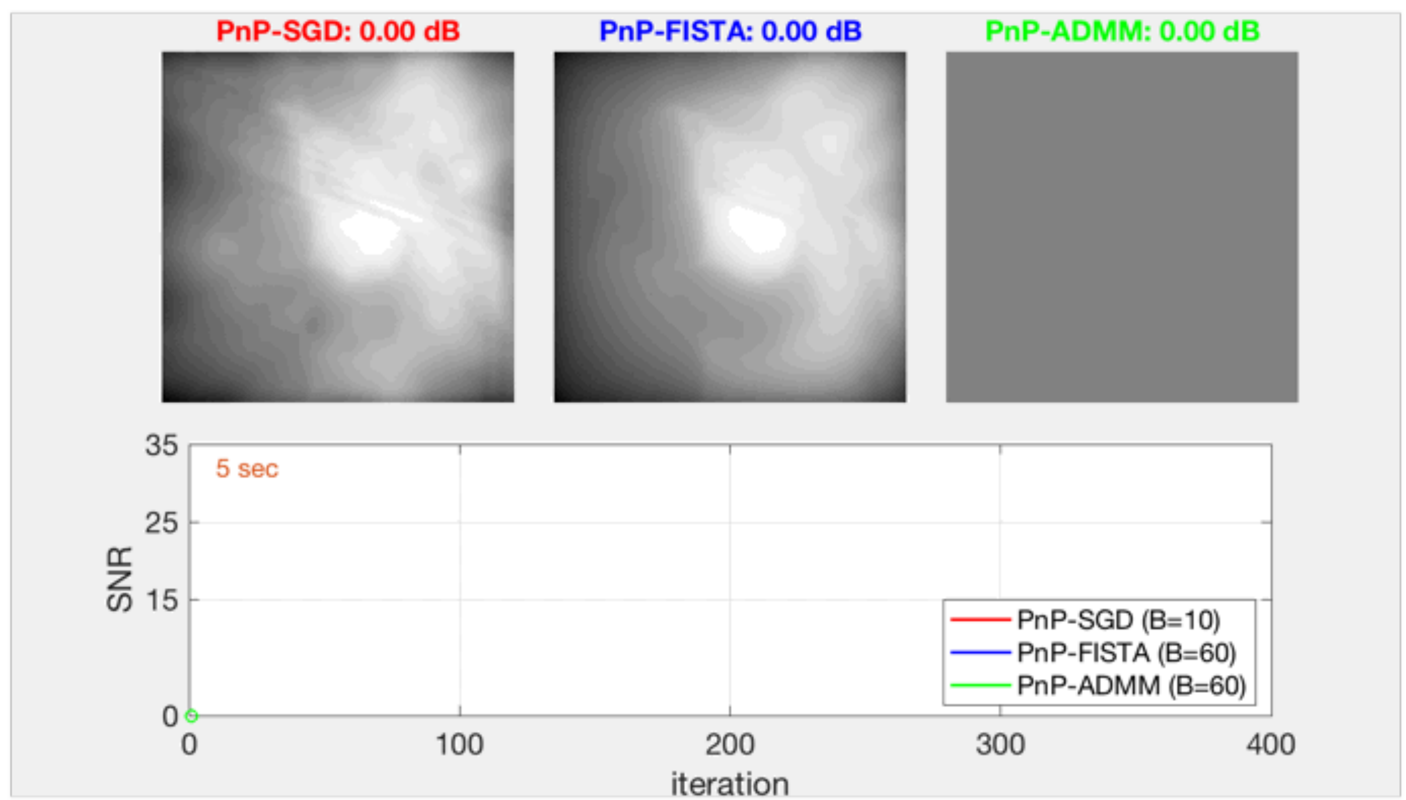
$$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$

PnP-SGD

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

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

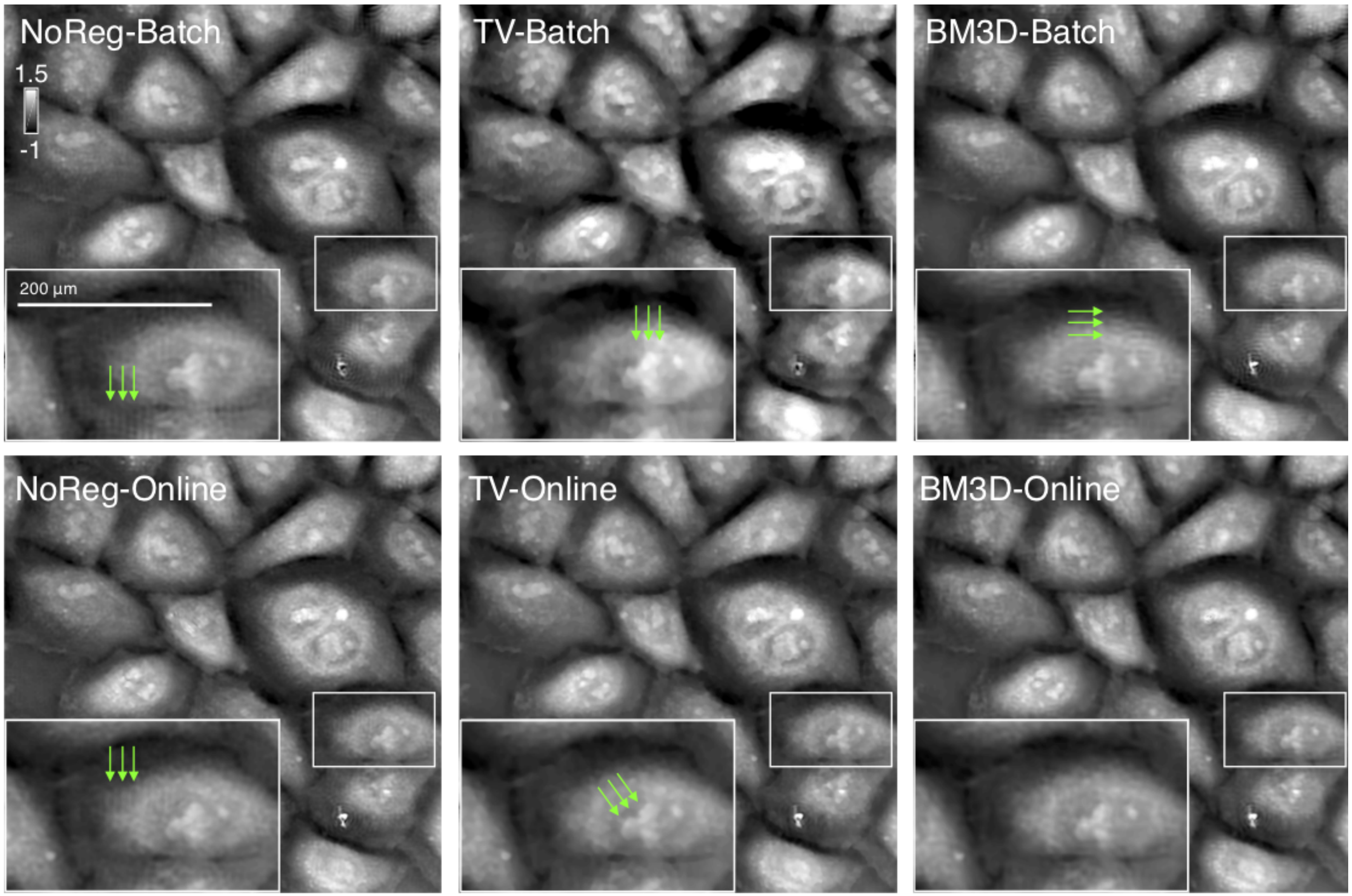
$$\boldsymbol{s}^k \leftarrow \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$$



PnP-SGD enables fast regularized inversion in **optical tomography (OT)**

PnP-SGD enables to scale PnP-FISTA to a large number of tomographic measurements

Experimental Fourier Ptychographic Microscopy (FPM) data



Using 60 (out of total 293) illuminations per iteration

Sun *et al.*, "Regularized Fourier Ptychography using an Online Plug-and-Play Algorithm," *Proc. IEEE ICASSP 2019*

On-RED is a stochastic variant of GM-RED suitable for large-scale tomographic problems

On-RED is a stochastic variant of GM-RED suitable for large-scale tomographic problems

Algorithm 1 GM-RED

1: **input:** $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\tau > 0$, and $\sigma > 0$
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad \nabla g(\boldsymbol{x}^{k-1}) \leftarrow \mathsf{fullGradient}(\boldsymbol{x}^{k-1})$
4: $\quad \mathsf{G}(\boldsymbol{x}^{k-1}) \leftarrow \nabla g(\boldsymbol{x}^{k-1}) + \tau(\boldsymbol{x}^{k-1} - \mathsf{D}_\sigma(\boldsymbol{x}^{k-1}))$
5: $\quad \boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \mathsf{G}(\boldsymbol{x}^{k-1})$
6: **end for**

Algorithm 2 On-RED

1: **input:** $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\tau > 0$, $\sigma > 0$, and $B \geq 1$
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad \widehat{\nabla} g(\boldsymbol{x}^{k-1}) \leftarrow \mathsf{minibatchGradient}(\boldsymbol{x}^{k-1}, B)$
4: $\quad \widehat{\mathsf{G}}(\boldsymbol{x}^{k-1}) \leftarrow \widehat{\nabla} g(\boldsymbol{x}^{k-1}) + \tau(\boldsymbol{x}^{k-1} - \mathsf{D}_\sigma(\boldsymbol{x}^{k-1}))$
5: $\quad \boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \widehat{\mathsf{G}}(\boldsymbol{x}^{k-1})$
6: **end for**

On-RED uses only B << L measurements per iteration

Theorem 1. Run GM-RED for $t \geq 1$ iterations under Assumptions 1-3 using a fixed step-size $0 < \gamma \leq 1/(L + 2\tau)$ and a fixed minibatch size $B = t$. Then, we have

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=1}^{t} \|\mathsf{G}(\boldsymbol{x}^{k-1})\|_2^2\right] \leq \frac{C}{\sqrt{t}},$$

where $C > 0$ is a constant.

Convergence similar to PnP-SGD

Wu *et al.*, "Online Regularization by Denoising with Applications to Phase Retrieval," *Proc. ICCVW*, 2019

Washington University in St. Louis

On-RED is a stochastic variant of GM-RED suitable for large-scale tomographic problems

Algorithm 1 GM-RED

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\tau > 0$, and $\sigma > 0$
for $k = 1, 2, \dots$ **do**
  $\nabla g(\boldsymbol{x}^{k-1}) \leftarrow \mathsf{fullGradient}(\boldsymbol{x}^{k-1})$
  $\mathsf{G}(\boldsymbol{x}^{k-1}) \leftarrow \nabla g(\boldsymbol{x}^{k-1}) + \tau(\boldsymbol{x}^{k-1} - \mathsf{D}_\sigma(\boldsymbol{x}^{k-1}))$
  $\boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \mathsf{G}(\boldsymbol{x}^{k-1})$
end for

Algorithm 2 On-RED

input: $\boldsymbol{x}^0 \in \mathbb{R}^n$, $\tau > 0$, $\sigma > 0$, and $B \geq 1$
for $k = 1, 2, \dots$ **do**
  $\widehat{\nabla} g(\boldsymbol{x}^{k-1}) \leftarrow \mathsf{minibatchGradient}(\boldsymbol{x}^{k-1}, B)$
  $\widehat{\mathsf{G}}(\boldsymbol{x}^{k-1}) \leftarrow \widehat{\nabla} g(\boldsymbol{x}^{k-1}) + \tau(\boldsymbol{x}^{k-1} - \mathsf{D}_\sigma(\boldsymbol{x}^{k-1}))$
  $\boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \widehat{\mathsf{G}}(\boldsymbol{x}^{k-1})$
end for

Table 3. Optimized SNR for each test image in dB

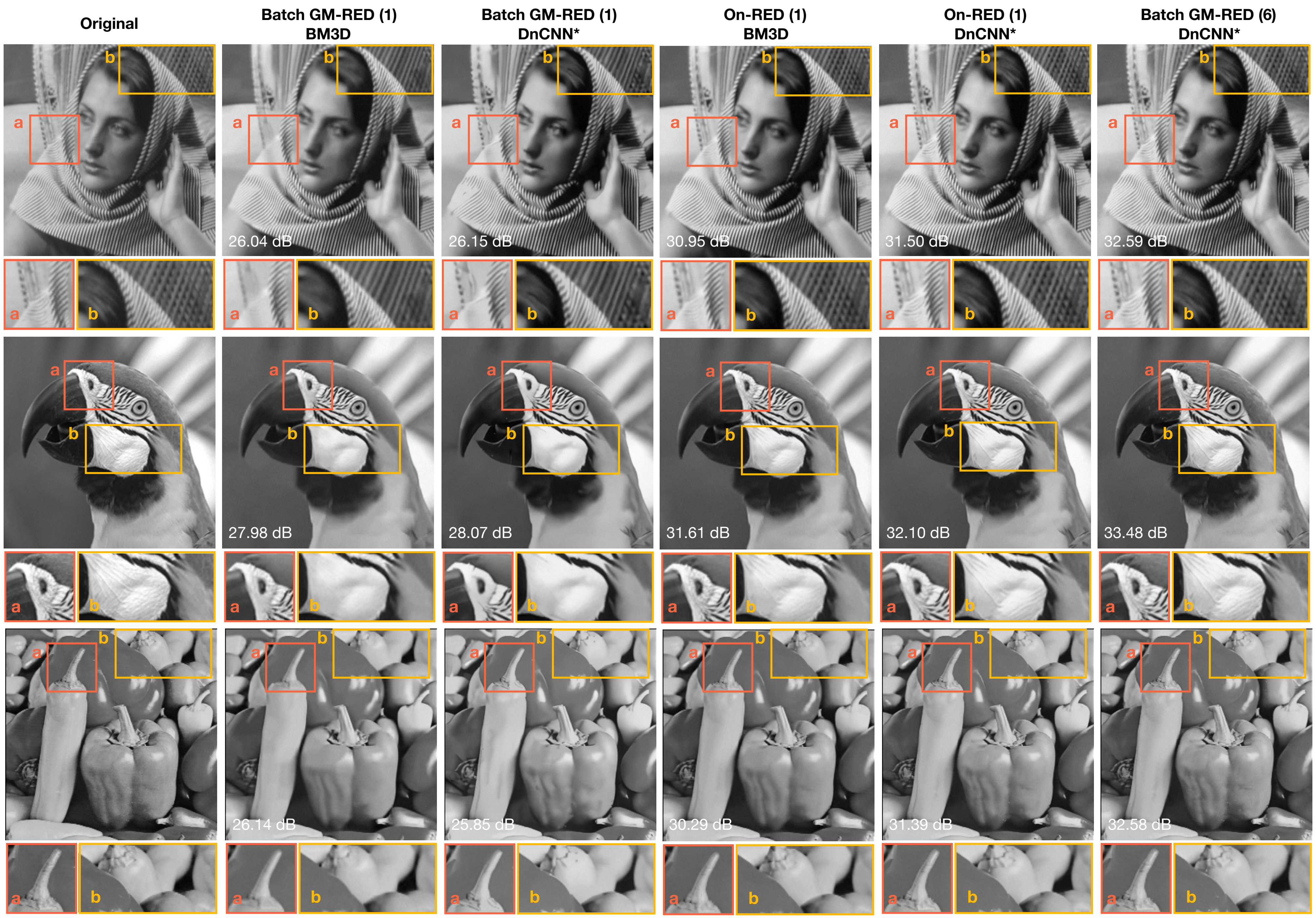
Algorithms ($I = 6$)	SGM ($B = 1$)	GM-RED (fixed 1)		On-RED ($B = 1$)		GM-RED (fixed 6)
Denoisers	—	BM3D	DnCNN*	BM3D	DnCNN*	DnCNN*
Barbara	27.37	26.04	26.15	30.95	31.50	32.59
Boat	27.68	26.90	27.53	31.65	32.61	33.17
Lenna	27.65	26.55	27.58	31.47	32.54	33.20
Monarch	27.51	24.76	26.34	29.66	31.31	32.63
Parrot	27.20	27.98	28.07	31.61	32.10	33.48
Pepper	27.08	26.14	25.85	30.29	31.39	32.58
Average	27.42	26.40	26.92	30.94	31.91	32.94

Phase retrieval from coded diffraction pattern (CDP) measurements

Metzler *et al.*, "prDeep: Robust Phase Retrieval with a Flexible Deep Network ," *Proc. ICML*, 2018

Wu *et al.*, "Online Regularization by Denoising with Applications to Phase Retrieval," *Proc. ICCVW*, 2019

On-RED is a stochastic variant of GM-RED suitable for large-scale tomographic problems



Wu *et al.*, “Online Regularization by Denoising with Applications to Phase Retrieval,” *Proc. ICCVW*, Oct 2019

BC-RED is a block-coordinate variant of GM-RED suitable for processing very large images

\

BC-RED is a block-coordinate variant of GM-RED suitable for processing very large images

Algorithm 1 Block Coordinate Regularization by Denoising (BC-RED)

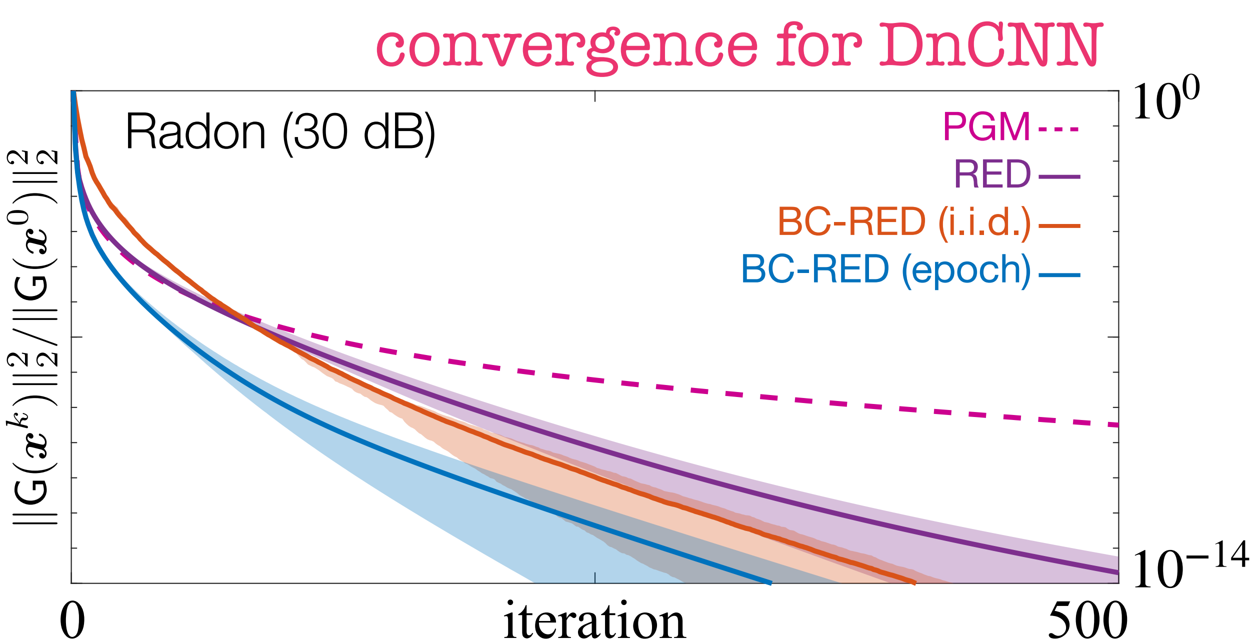
input: initial value $\boldsymbol{x}^0 \in \mathbb{R}^n$, parameter $\tau > 0$, and step-size $\gamma > 0$.
for $k = 1, 2, 3, \dots$ **do**
  Choose an index $i_k \in \{1, \dots, b\}$
  $\boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \mathsf{U}_{i_k} \mathsf{G}_{i_k}(\boldsymbol{x}^{k-1})$
    where $\mathsf{G}_i(\boldsymbol{x}) := \mathsf{U}_i^\mathsf{T} \mathsf{G}(\boldsymbol{x})$ with $\mathsf{G}(\boldsymbol{x}) := \nabla g(\boldsymbol{x}) + \tau(\boldsymbol{x} - \mathsf{D}(\boldsymbol{x}))$.
end for

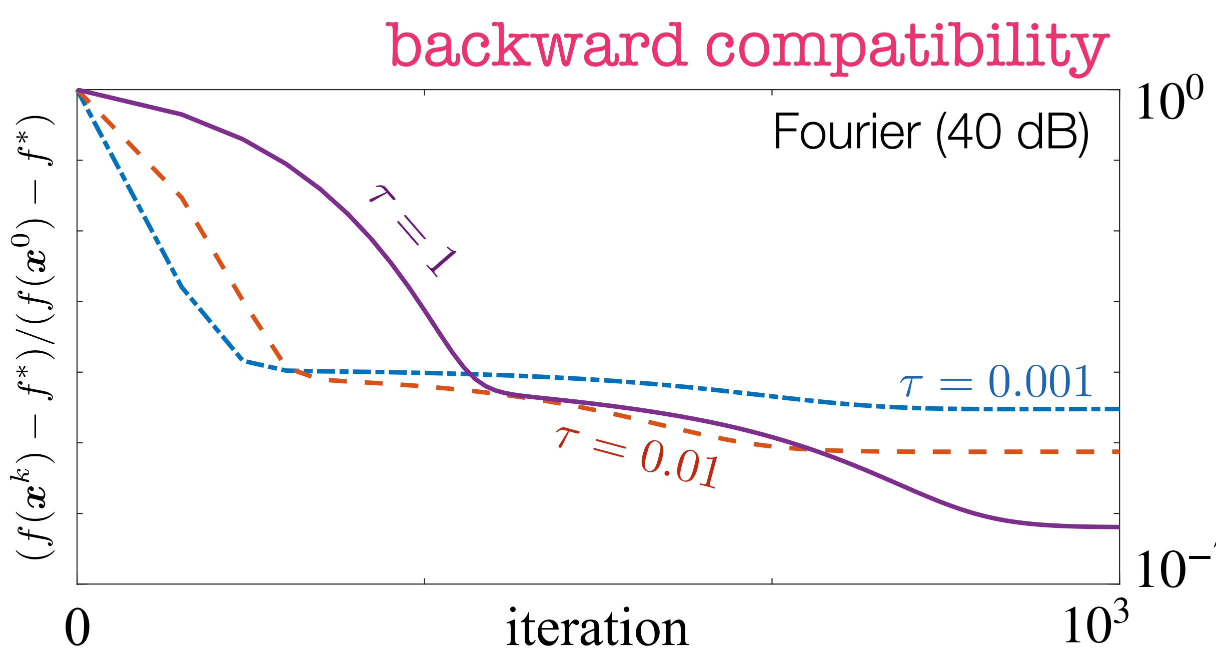
BC-RED can be faster than GM-RED

Download the code here:
github.com/wustl-cig/bcred

Sun *et al.*, “Block Coordinate Regularization by Denoising,” *Proc. NeurIPS*, Dec 2019

BC-RED is a block-coordinate variant of GM-RED suitable for processing very large images





Download the code here:
github.com/wustl-cig/bcred

Sun *et al.*, "Block Coordinate Regularization by Denoising," *Proc. NeurIPS*, Dec 2019

BC-RED is a block-coordinate variant of GM-RED suitable for processing very large images

Table 1: Average SNRs obtained for different measurement matrices and image priors.

Methods	**Radon**		**Random**		**Fourier**	
	30 dB	**40 dB**	**30 dB**	**40 dB**	**30 dB**	**40 dB**
PGM (TV)	20.66	24.40	26.07	***28.42***	28.74	29.99
U-Net	***21.90***	21.72	16.37	16.40	22.11	22.11
RED (TV)	20.79	24.46	25.64	28.30	28.67	29.97
BC-RED (TV)	20.78	24.42	25.70	28.40	28.71	29.99
RED (BM3D)	21.55	***25.24***	26.46	27.82	28.89	29.79
BC-RED (BM3D)	21.56	25.16	26.52	27.89	28.85	29.80
RED (DnCNN*)	20.89	24.38	26.53	28.05	29.33	*30.32*
BC-RED (DnCNN*)	20.88	24.42	***26.60***	28.12	***29.40***	***30.39***

Download the code here:
github.com/wustl-cig/bcred

Sun *et al.*, "Block Coordinate Regularization by Denoising," *Proc. NeurIPS*, Dec 2019

To conclude

- We increasingly rely on implicit regularization using nonlinear operators, such as deep neural nets
- PnP and RED are a theoretically sound algorithm that can leverage denoising neural nets as priors
- For certain large-scale problems, it is beneficial to consider scalable PnP-SGD, BC-RED, and On-RED

Washington University in St. Louis

Computational Imaging Group (CIG) at Washington University in St. Louis (WashU)

CONTACT INFO

Prof. Ulugbek Kamilov
Email: kamilov@wustl.edu
Personal Twitter: @ukmlv

Web: http://cigroup.wustl.edu
Group Twitter: @wustlcig

Support:

CCF-181391

point your camera
get the slides

Acknowledgements

- This material is based upon work supported by the National Science Foundation under Grant No. 1813910. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.