Plug-and-Play Models for Large-Scale Computational Imaging

Ulugbek S. Kamilov
Computational Imaging Group (CIG)
Twitter: **@ukmlv**

PnP Workshop – 8 Dec 2022 – Paris, France

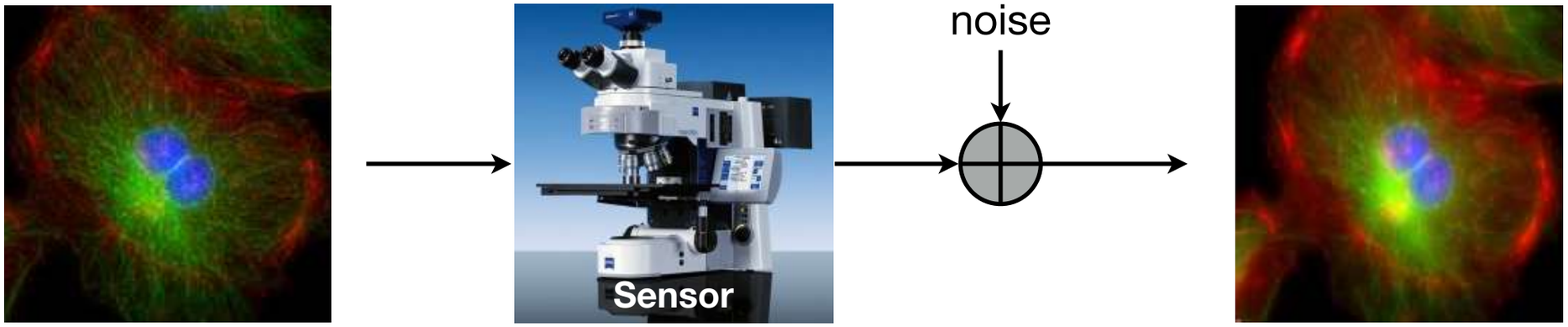
Washington University in St. Louis

Big **Thank You** to the members of the WashU **Computational Imaging Group (CIG)**

From left to right in the photo:
Weijie Gan, Tomas Kerepecky (alum), Xiaojian Xu (alum), Jiaming Liu, Flora Sun, Yu Sun (alum), Yuyang Hu, and Shirin Shoushtari.

Many **computational imaging** problems
can be formulated as **inverse problems**

Many computational imaging problems can be formulated as inverse problems

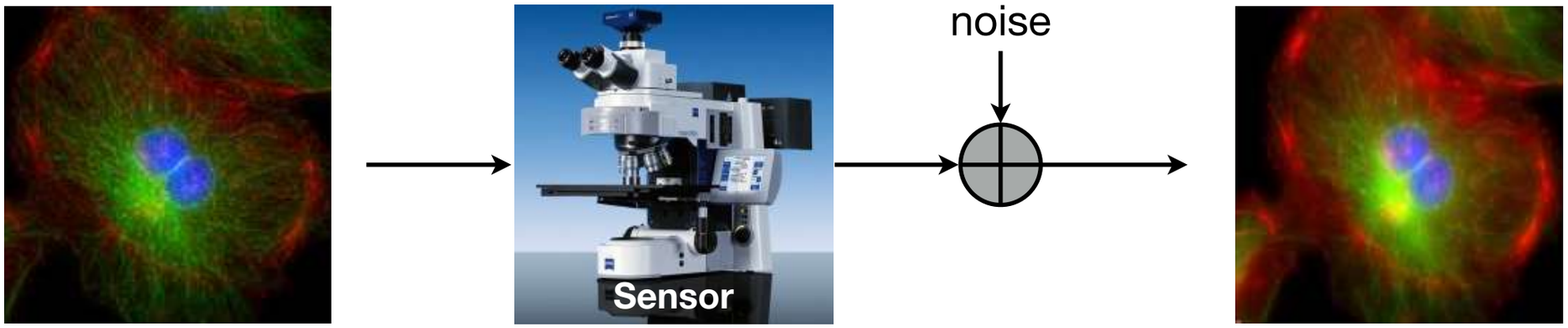


$\boldsymbol{x}$ $\quad$ $\boldsymbol{A}$ $\quad$ $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}) + \boldsymbol{e}$

Source: Michael Unser

Many computational imaging problems can be formulated as inverse problems

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

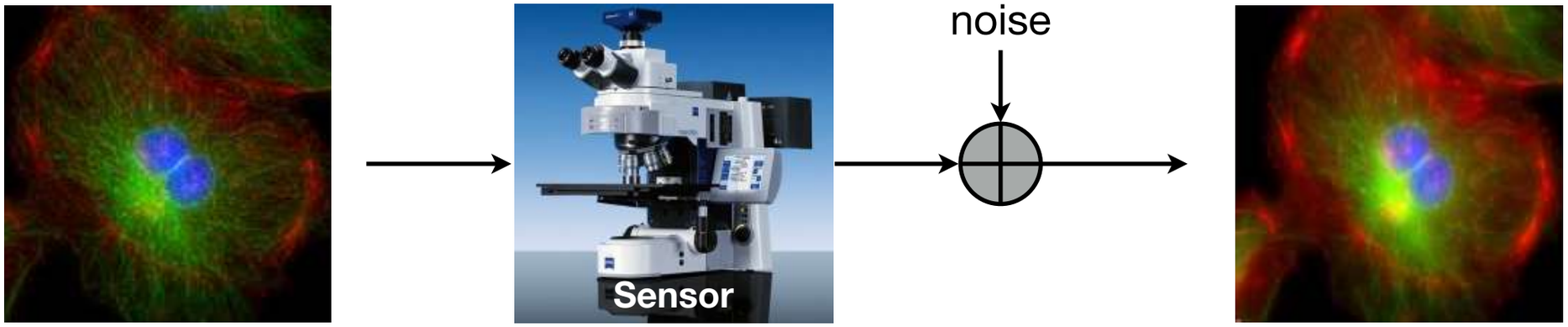


$\boldsymbol{x}$ $\quad$ $\boldsymbol{A}$ $\quad$ $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}) + \boldsymbol{e}$

Source: Michael Unser

Many computational imaging problems can be formulated as inverse problems

Forward problem: generate y from x



x $\quad$ A $\quad$ $y = A(x) + e$

Inverse problem: recover x from y

Source: Michael Unser

Many computational imaging problems can be formulated as inverse problems

$$y = A(x) + e$$

Image restoration

Denoising

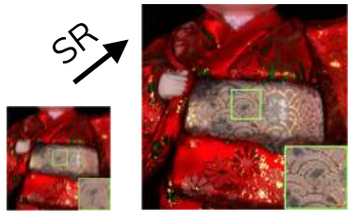
$A =$ (Identity)

Deblurring

$A =$

Super-Resolution

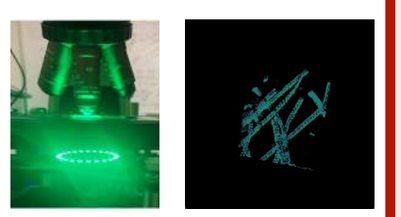
$A =$



Bio Imaging

Optical Microscopy

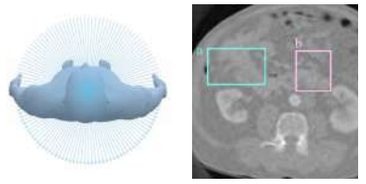
$A =$



Medical Imaging

Sparse-View CT

$A =$



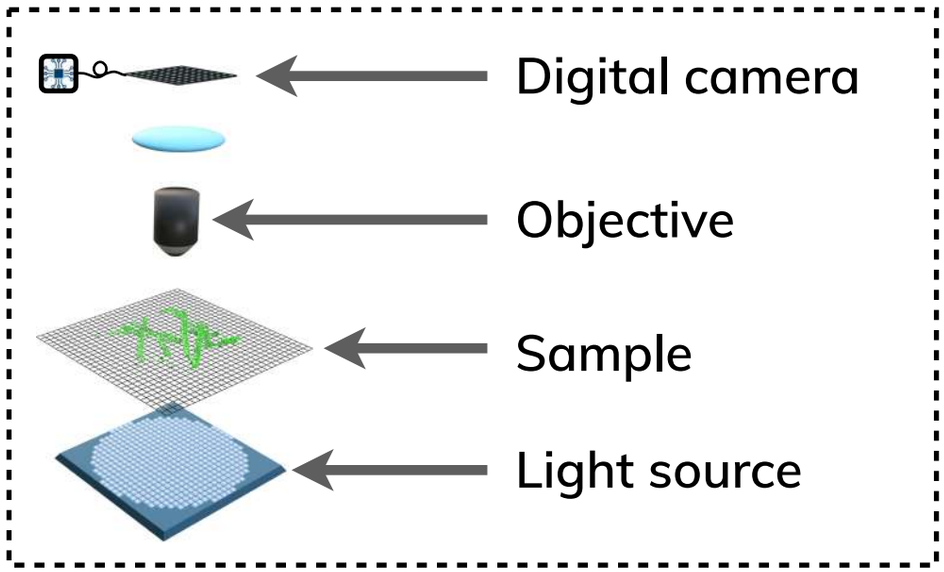
Accelerated MRI

$A =$

Source: Yu Sun

Example: Intensity diffraction tomography (IDT)
collects intensity measurements of scattered light

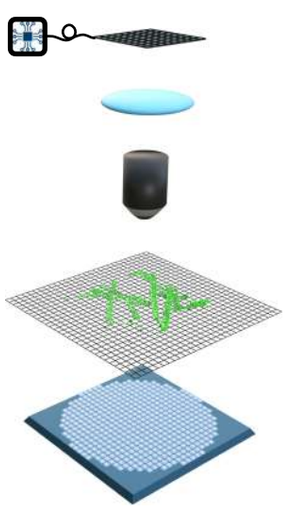
Example: Intensity diffraction tomography (IDT) collects intensity measurements of scattered light



IDT is a relatively cheap and simple optical microscopy system

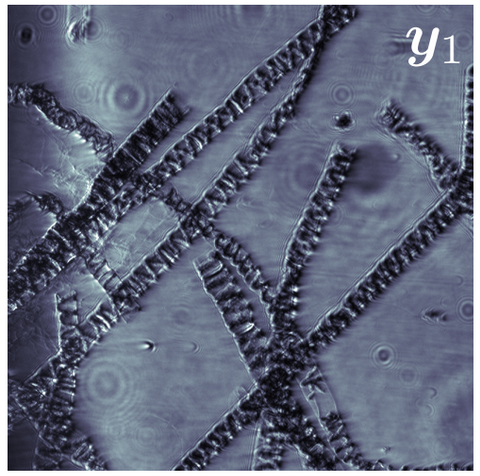
Liu et al, "Recovery of continuous 3D refractive index maps from discrete intensity-only measurements using neural fields," ***Nature Machine Intelligence***, 2022

Example: Intensity diffraction tomography (IDT) collects intensity measurements of scattered light

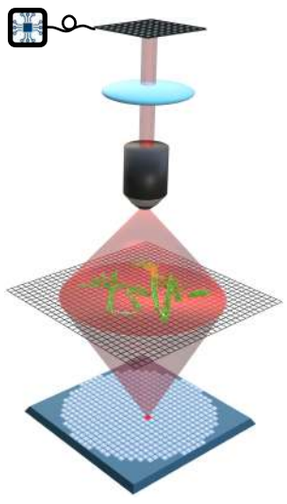


Liu et al, "Recovery of continuous 3D refractive index maps from discrete intensity-only measurements using neural fields," ***Nature Machine Intelligence***, 2022

Example: Intensity diffraction tomography (IDT) collects intensity measurements of scattered light

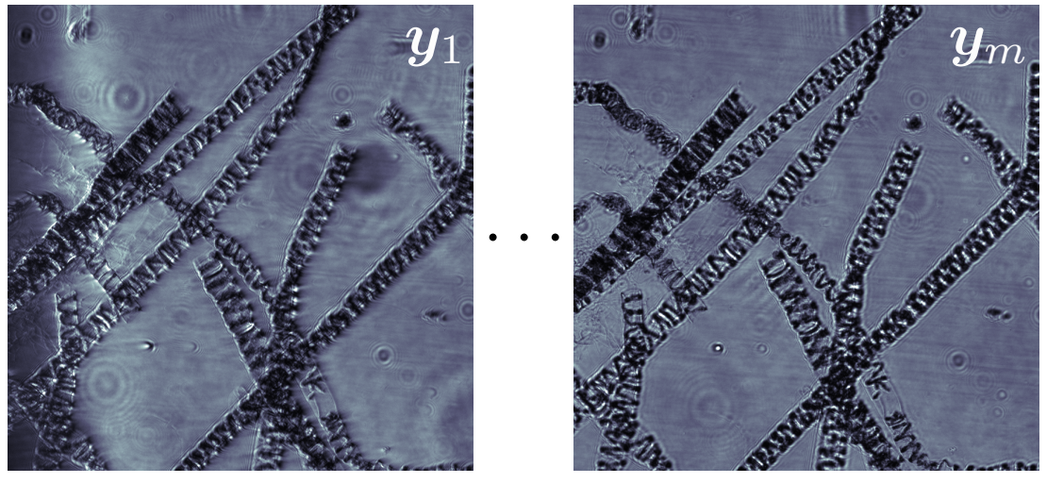


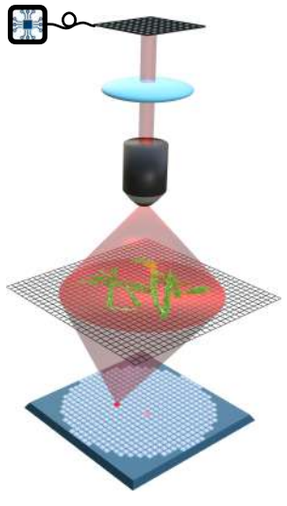
intensity image 1



Liu et al, "Recovery of continuous 3D refractive index maps from discrete intensity-only measurements using neural fields," ***Nature Machine Intelligence***, 2022

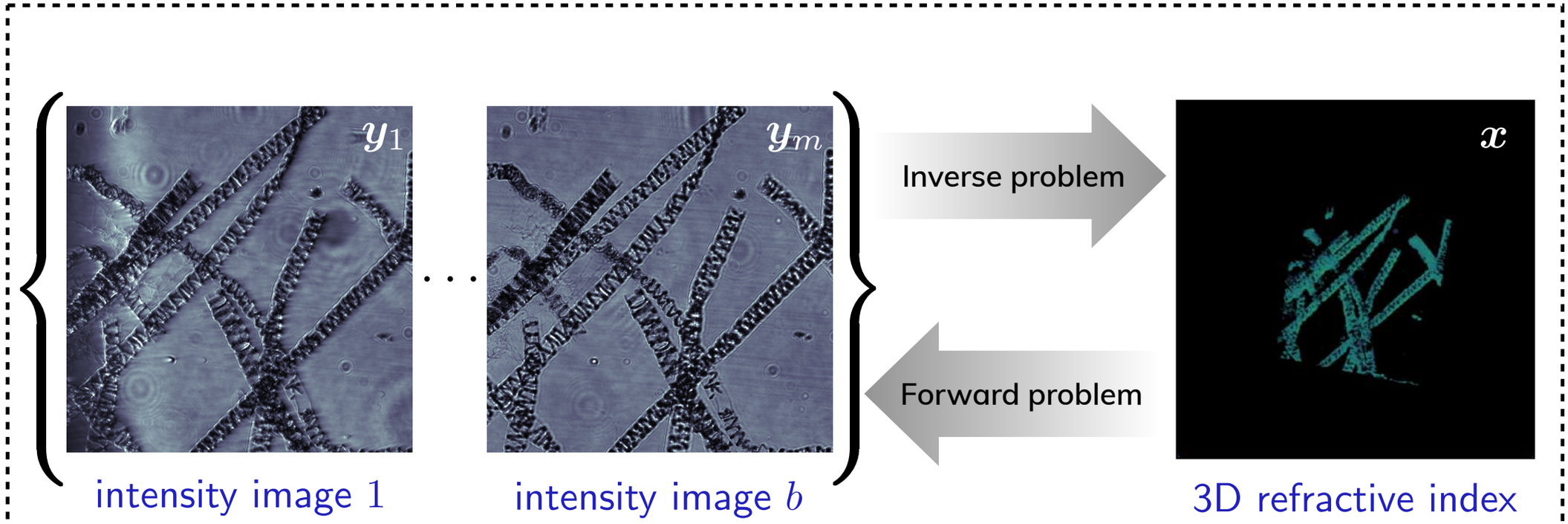
Example: Intensity diffraction tomography (IDT) collects intensity measurements of scattered light



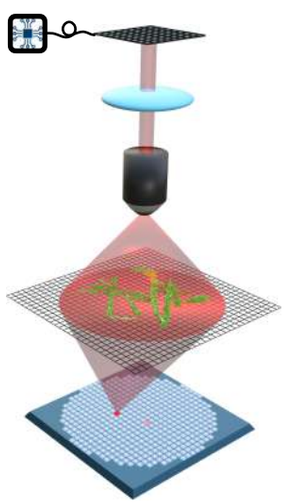


Liu et al, "Recovery of continuous 3D refractive index maps from discrete intensity-only measurements using neural fields," ***Nature Machine Intelligence***, 2022

Example: Intensity diffraction tomography (IDT) collects intensity measurements of scattered light



IDT is a data-intensive limited-angle tomography under **light-scattering** and **phase-loss**

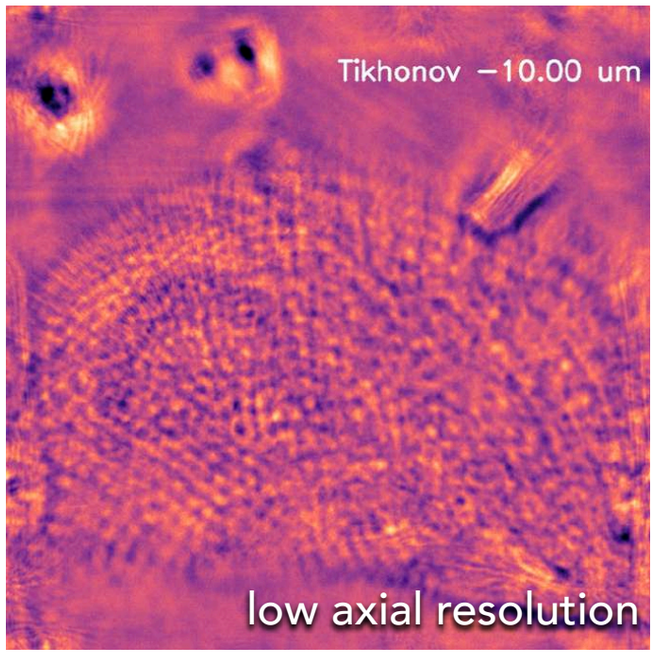


Liu et al, "Recovery of continuous 3D refractive index maps from discrete intensity-only measurements using neural fields," ***Nature Machine Intelligence***, 2022

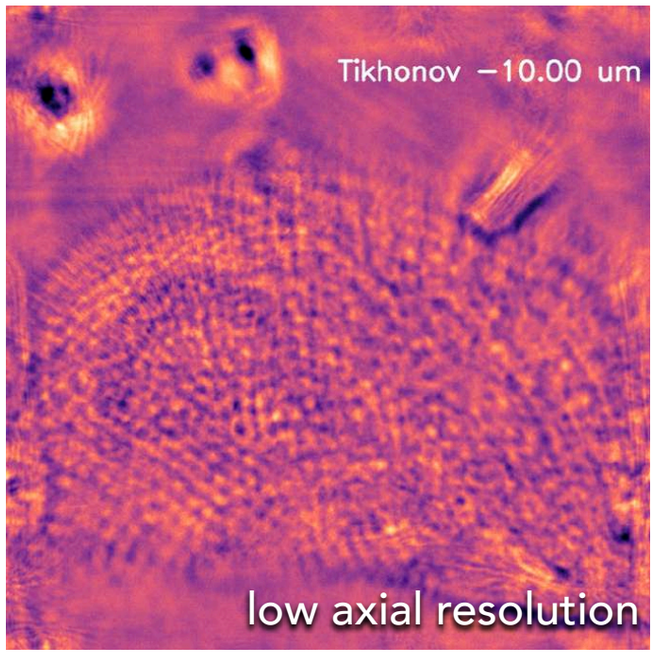
Washington University in St. Louis

Classical issues in the context of **inverse problems**: **costly acquisition**, **imaging artifacts**, and **big data**

Classical issues in the context of inverse problems: costly acquisition, imaging artifacts, and big data

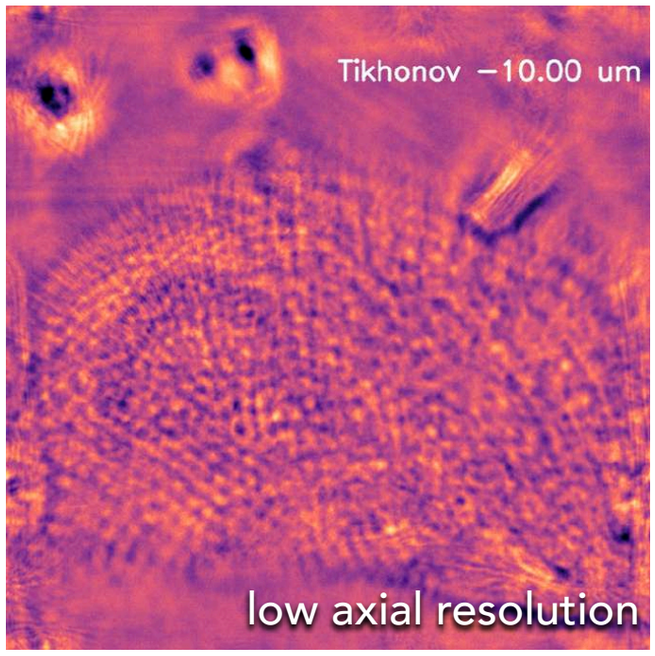


Classical issues in the context of inverse problems: costly acquisition, imaging artifacts, and big data



Challenge #1: Acquisition is too slow or costly:
Due to sequential and indirect acquisition of data

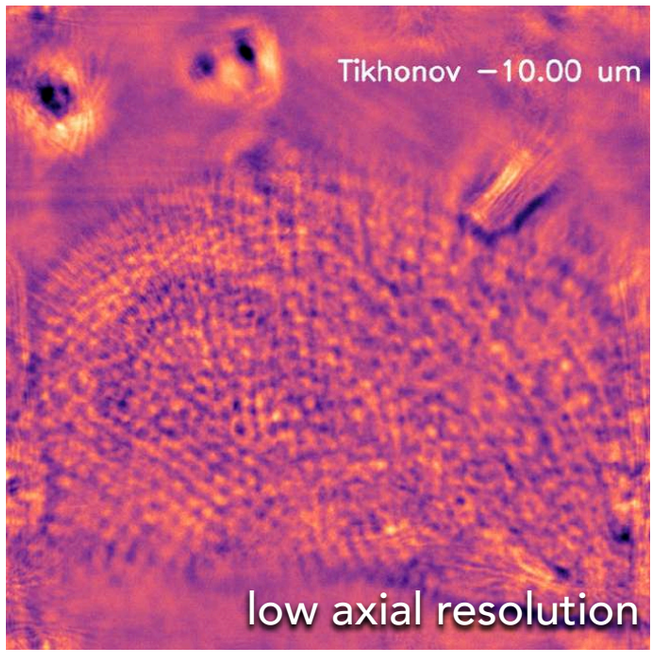
Classical issues in the context of inverse problems: costly acquisition, imaging artifacts, and big data



Challenge #1: Acquisition is too slow or costly:
Due to sequential and indirect acquisition of data

Challenge #2: Reconstructed images contain artifacts:
Due to undersampling, motion, model mismatch, and noise

Classical issues in the context of inverse problems: costly acquisition, imaging artifacts, and big data



Challenge #1: Acquisition is too slow or costly:
Due to sequential and indirect acquisition of data

Challenge #2: Reconstructed images contain artifacts:
Due to undersampling, motion, model mismatch, and noise

Challenge #3: High computational/memory requirements:
Due to large volumes of data to process in 3D, 4D, or 5D

Outline for the rest of the talk

- Plug-and-Play Methods for Inverse Problems (**IEEE SPM 2022**)
  Integrating physical models and learned deep priors

- Online Deep Equilibrium Learning (**NeurIPS 2022**)
  A new PnP framework for efficient prior learning

- Deep Continuous Artifact-Free Fields (**Nature MI 2022**)
  A new PnP framework for continuous image recovery

Outline for the rest of the talk

- Plug-and-Play Methods for Inverse Problems (**IEEE SPM 2022**)
  Integrating physical models and learned deep priors

- Online Deep Equilibrium Learning (**NeurIPS 2022**)
  A new PnP framework for efficient prior learning

- Deep Continuous Artifact-Free Fields (**Nature MI 2022**)
  A new PnP framework for continuous image recovery

Washington University in St.Louis

Plug-and-Play Methods (PnP) tutorial for IEEE Signal Processing Magazine is available online

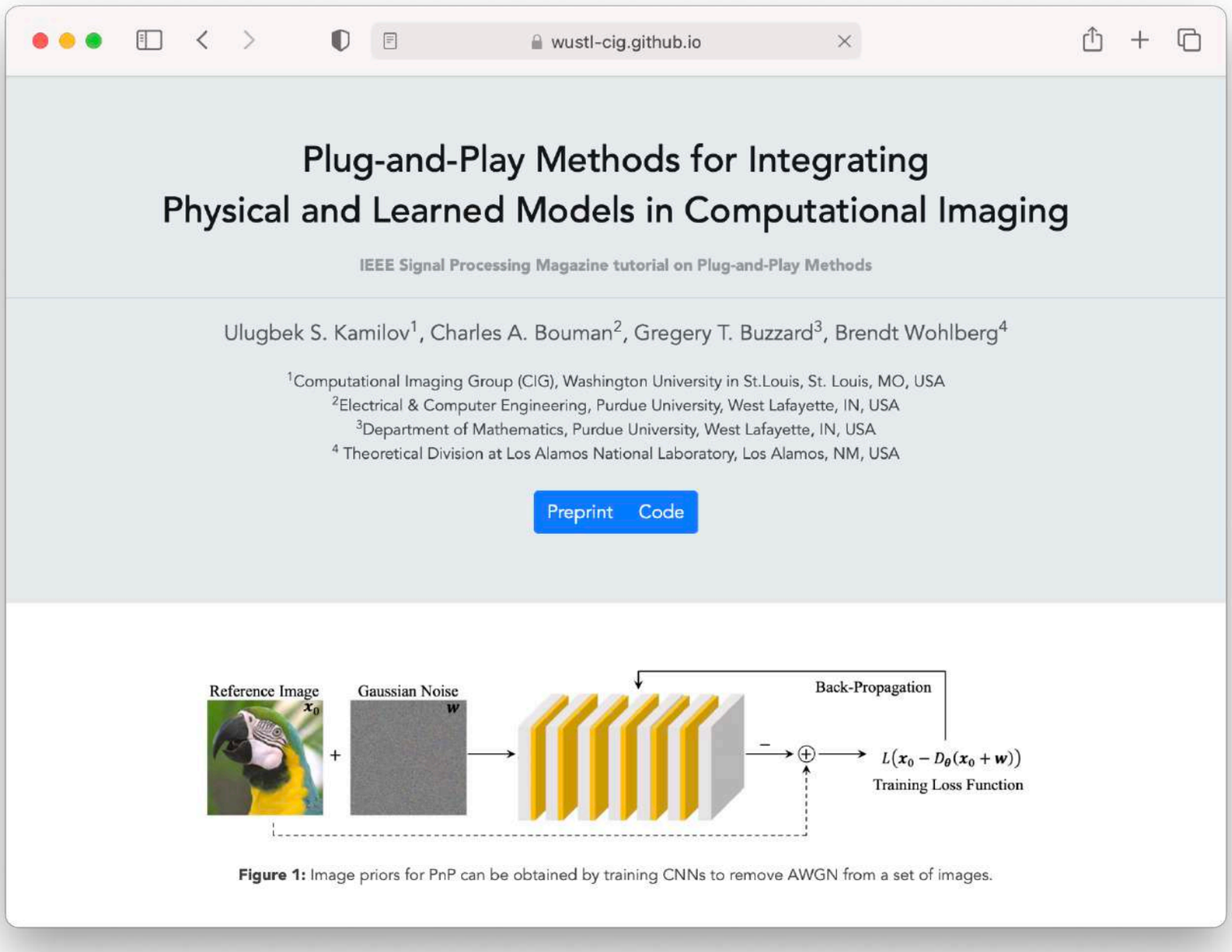


Figure 1: Image priors for PnP can be obtained by training CNNs to remove AWGN from a set of images.

Scan the QR code to see the tutorial

Kamilov *et al*, "Plug-and-Play Methods for Integrating Physical and Learned Models in Computational Imaging," ***IEEE Signal Processing Magazine***, 2022

Bayesian perspective is common, but the view on how to best represent image priors is still evolving

Bayesian perspective is common, but the view on how to best represent image priors is still evolving

Probabilistic interpretation of an inverse problem

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

forward model

$$\boldsymbol{x} \sim p_{\boldsymbol{x}}$$

image prior

$$\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$

noise model

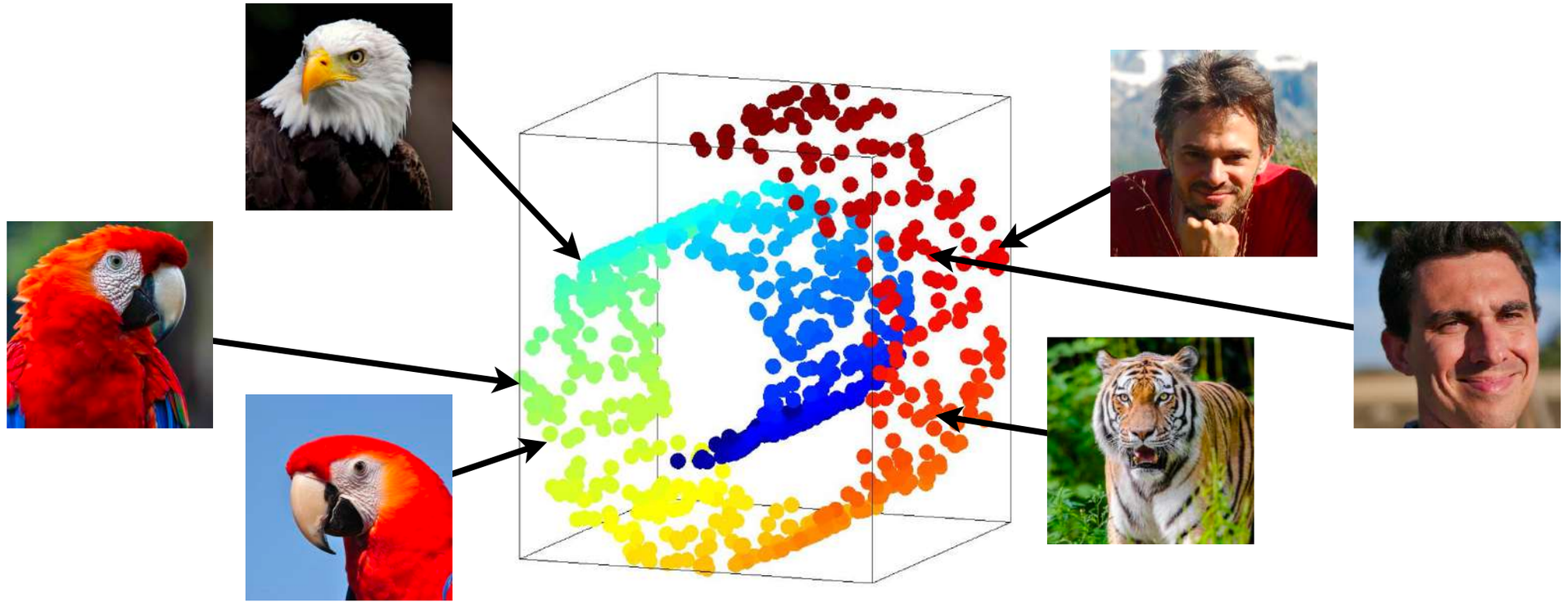
Bayesian perspective is common, but the view on how to best represent image priors is still evolving

Probabilistic interpretation of an inverse problem

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}) + \boldsymbol{e}$$ forward model

$$\boldsymbol{x} \sim p_{\boldsymbol{x}}$$ image prior

$$\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$ noise model



Remark: Generating images from $p_{\boldsymbol{x}}$ is equivalent to sampling from a lower-dimensional and non-linear subset $\mathcal{X} \subset \mathbb{R}^n$ of a high-dimensional space

Bayesian perspective is common, but the view on how to best represent image priors is still evolving

Probabilistic interpretation of an inverse problem

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}) + \boldsymbol{e} \qquad \boldsymbol{x} \sim p_{\boldsymbol{x}} \qquad \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$

MAP and **MMSE** statistical estimators can be expressed as **model-based optimization**

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\mathsf{arg\,min}} \left\{ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}(\boldsymbol{x})\|_2^2 + \sigma^2 h(\boldsymbol{x}) \right\}$$

$$h_{\mathsf{MAP}}(\boldsymbol{x}) = -\mathsf{log}(p_{\boldsymbol{x}}(\boldsymbol{x}))$$

Remark: Maximum a posteriori probability (MAP) estimator returns an image that maximizes $p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})$

Bayesian perspective is common, but the view on how to best represent image priors is still evolving

Probabilistic interpretation of an inverse problem

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}) + \boldsymbol{e} \qquad \boldsymbol{x} \sim p_{\boldsymbol{x}} \qquad \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$

MAP and **MMSE** statistical estimators can be expressed as **model-based optimization**

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}(\boldsymbol{x})\|_2^2 + \sigma^2 h(\boldsymbol{x}) \right\}$$

$h_{\text{MAP}}(\boldsymbol{x}) = -\log(p_{\boldsymbol{x}}(\boldsymbol{x}))$

$h_{\text{MMSE}}(\boldsymbol{x}) =$ some expression

Remark: Minimum mean squared error (MMSE) estimator returns the conditional mean $\mathbb{E}[\boldsymbol{x}|\boldsymbol{y}]$, which is the solution that maximizes the signal-to-noise ratio (SNR)

Gribonval and Machart, "Reconciling "priors" & "priors" without prejudice?" ***Proc NeurIPS***, 2013

Bayesian perspective is common, but the view on how to best represent image priors is still evolving

Probabilistic interpretation of an inverse problem

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{x}) + \boldsymbol{e} \qquad \boldsymbol{x} \sim p_{\boldsymbol{x}} \qquad \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$$

MAP and MMSE statistical estimators can be expressed as model-based optimization

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}(\boldsymbol{x})\|_2^2 + \sigma^2 h(\boldsymbol{x}) \right\}$$

$$h_{\text{MAP}}(\boldsymbol{x}) = -\log(p_{\boldsymbol{x}}(\boldsymbol{x}))$$

$$h_{\text{MMSE}}(\boldsymbol{x}) = \text{some expression}$$

Image denoising is a special inverse problem where the forward model is an identity

$$\mathsf{D}_\sigma(\boldsymbol{y}) = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \sigma^2 h(\boldsymbol{x}) \right\}$$

Proximal operator is a convenient proxy for the prior from the algorithmic perspective

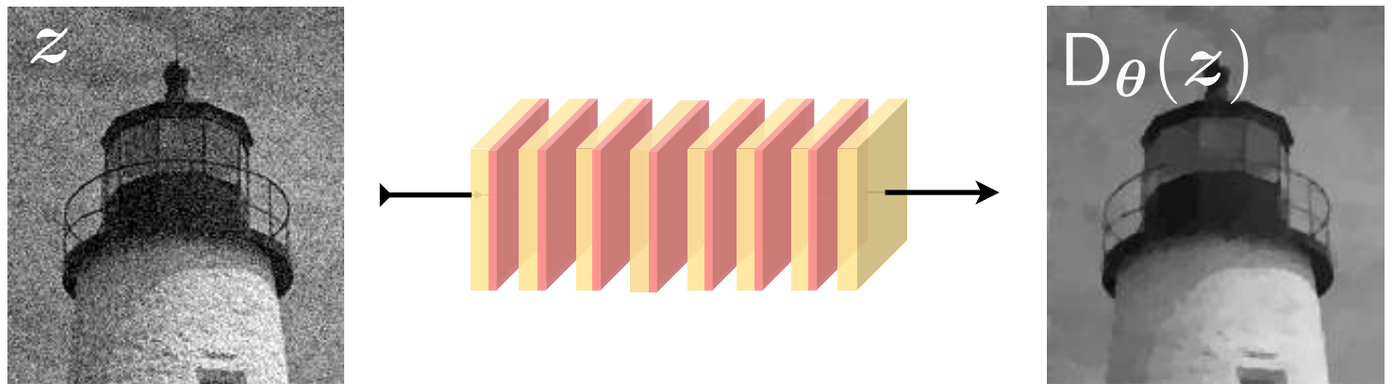
Washington University in St. Louis

Plug-and-Play Methods (PnP) are flexible deep models that use image denoisers as image priors

Plug-and-Play Methods (PnP) are flexible deep models that use image denoisers as image priors

Learned model: Pre-trained **image denoising** neural network

$\mathsf{D}_{\boldsymbol{\theta}}$: more noisy image $\mapsto$ less noisy image



Remark: PnP is a self-supervised learning framework since the image prior is learned on a "pretext task"

Plug-and-Play Methods (PnP) are flexible deep models that use image denoisers as image priors

Learned model: Pre-trained image denoising neural network

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

Physical model: Infuses information from the **forward model**

$$\mathsf{I} - \gamma \nabla g : \text{less measurement consistent} \mapsto \text{more measurement consistent}$$

Remark: Smaller values of $g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}(\boldsymbol{x})\|_2^2$ are more consistent with the measured data

Plug-and-Play Methods (PnP) are flexible deep models that use image denoisers as image priors

Learned model: Pre-trained image denoising neural network

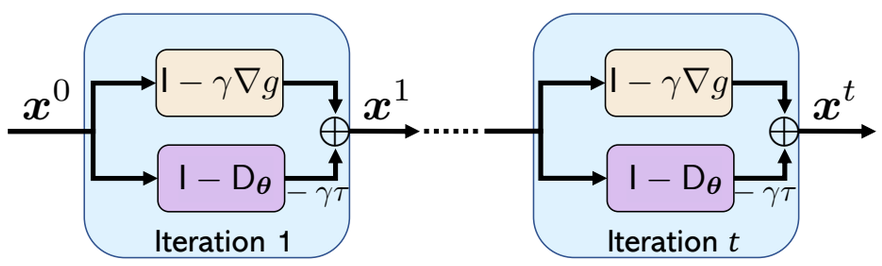
$$\mathsf{D}_{\boldsymbol{\theta}} : \text{more noisy image} \mapsto \text{less noisy image}$$

Physical model: Infuses information from the forward model

$$\mathsf{I} - \gamma \nabla g : \text{less measurement consistent} \mapsto \text{more measurement consistent}$$

PnP methods integrate both models into a **deep model-based architecture (DMBA)** that can have infinitely many layers

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



Romano et al, "The Little Engine That Could: Regularization by Denoising (RED)," ***SIAM J Imaging Science***, 2017

Plug-and-Play Methods (PnP) are flexible deep models that use image denoisers as image priors

Learned model: Pre-trained image denoising neural network

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

Physical model: Infuses information from the forward model

$$\mathsf{I} - \gamma \nabla g : \text{less measurement consistent} \mapsto \text{more measurement consistent}$$

PnP methods integrate both models into a deep model-based architecture (DMBA) that can have infinitely many layers

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

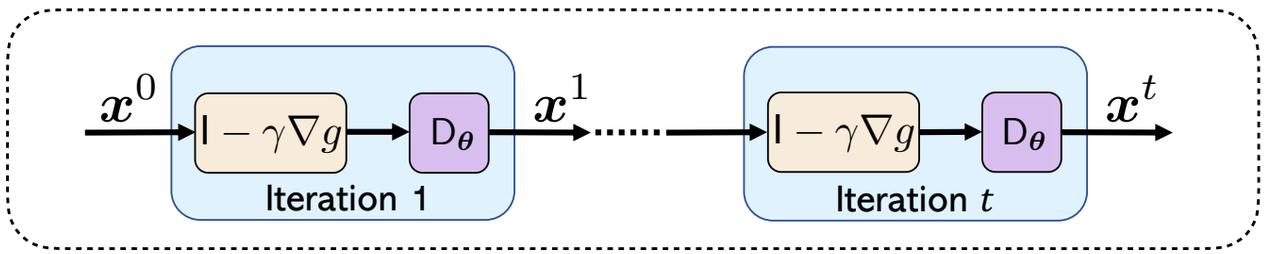
PnP methods have become influential in the context of inverse problems due to their mathematical elegance, flexibility, and performance

PnP does not have to be restricted to **PnP-ADMM**, **MAP**, **implicit priors**, **denoisers,** and **point estimates**

PnP does not have to be restricted to PnP-ADMM, MAP, implicit priors, denoisers, and point estimates

- While PnP-ADMM was the first PnP method, there are variants based on other algorithms (such as PnP-FISTA)

Example: Architecture of PnP-ISTA



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

PnP does not have to be restricted to PnP-ADMM, MAP, implicit priors, denoisers, and point estimates

- While PnP-ADMM was the first PnP method, there are variants based on other algorithms (such as PnP-FISTA)
- While original PnP is interpreted as using implicit priors, there are formulations based on explicit regularizers (such as RED)

Example: RED regularizer and its gradient

$$h(\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}}(\boldsymbol{x} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \Rightarrow \quad \nabla h(\boldsymbol{x}) = \boldsymbol{x} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Romano et al, "The Little Engine That Could: Regularization by Denoising (RED)," ***SIAM J Imaging Science***, 2017

Reehorst and Schniter, "Regularization by Denoising: Clarifications and New Interpretations," ***IEEE Trans Comput Imag***, 2019

PnP does not have to be restricted to PnP-ADMM, MAP, implicit priors, denoisers, and point estimates

- While PnP-ADMM was the first PnP method, there are variants based on other algorithms (such as PnP-FISTA)
- While original PnP is interpreted as using implicit priors, there are formulations based on explicit regularizers (such as RED)
- While the MAP view of the proximal operator is common, CNNs are often trained to act as MMSE estimators over the training data

$$\mathsf{D}_{\sigma}^{*}(\boldsymbol{z}) = \arg\min_{\mathsf{D}} \mathbb{E}\left[\|\boldsymbol{x} - \mathsf{D}(\boldsymbol{z})\|_2^2\right]$$

MMSE estimator for $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{e}$

PnP does not have to be restricted to PnP-ADMM, MAP, implicit priors, denoisers, and point estimates

- While PnP-ADMM was the first PnP method, there are variants based on other algorithms (such as PnP-FISTA)
- While original PnP is interpreted as using implicit priors, there are formulations based on explicit regularizers (such as RED)
- While the MAP view of the proximal operator is common, CNNs are often trained to act as MMSE estimators over the training data

Theorem: PnP-ISTA using D_σ^* monotonically converges to a stationary point of the following composite function

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + h_{\text{mmse}}(\boldsymbol{x})$$

- generally not convex
- $\neq h_{\text{map}}(\mathbf{x}) = -\log(p_x(\mathbf{x}))$
- MAP estimator for the prior: $p(\mathbf{x}) \propto \exp(-h_{\text{mmse}}(\mathbf{x}))$

Xu et al, "Provable Convergence of Plug-and-Play Priors with MMSE denoisers," ***IEEE SPL***, vol. 27, 2020

PnP does not have to be restricted to PnP-ADMM, MAP, implicit priors, denoisers, and point estimates

- While PnP-ADMM was the first PnP method, there are variants based on other algorithms (such as PnP-FISTA)
- While original PnP is interpreted as using implicit priors, there are formulations based on explicit regularizers (such as RED)
- While the MAP view of the proximal operator is common, CNNs are often trained to act as MMSE estimators over the training data

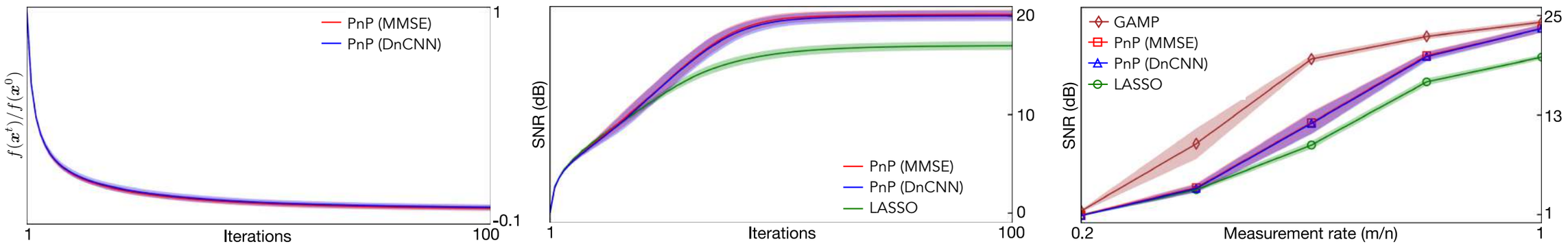
$$\|\mathsf{D}_\sigma(\boldsymbol{z}) - \mathsf{D}^*_\sigma(\boldsymbol{z})\|_2 \leq \sigma\varepsilon \quad \Rightarrow \quad \min_{1\leq i\leq t} \|\mathsf{G}^*(\boldsymbol{x}^t)\|_2^2 \leq \frac{C_1}{t} + \tau\sigma\varepsilon C_2$$

Remark: Even when the denoiser is not exact, PnP converges up to an error bound!

Shoushtari et al, "Deep Model-Based Architectures for Inverse Problems under Mismatched Priors," ***IEEE JSAIT***, 2022

PnP does not have to be restricted to PnP-ADMM, MAP, implicit priors, denoisers, and point estimates

- While PnP-ADMM was the first PnP method, there are variants based on other algorithms (such as PnP-FISTA)
- While original PnP is interpreted as using implicit priors, there are formulations based on explicit regularizers (such as RED)
- While the MAP view of the proximal operator is common, CNNs are often trained to act as MMSE estimators over the training data



Note how DnCNN, trained to approximate the MMSE denoiser, perfectly agrees with the results using the true MMSE denoiser!

Xu et al, "Provable Convergence of Plug-and-Play Priors with MMSE denoisers," ***IEEE SPL***, vol. 27, 2020

PnP does not have to be restricted to PnP-ADMM, MAP, implicit priors, denoisers, and point estimates

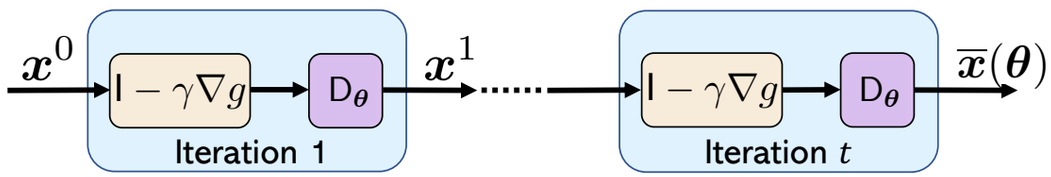
- While PnP-ADMM was the first PnP method, there are variants based on other algorithms (such as PnP-FISTA)

- While original PnP is interpreted as using implicit priors, there are formulations based on explicit regularizers (such as RED)

- While the MAP view of the proximal operator is common, CNNs are often trained to act as MMSE estimators over the training data

- While original PnP was restricted to image denoisers, there are tools for efficiently learning priors end-to-end (such as DEQ)

Example: DEQ formulation for the end-to-end optimization of PnP-ISTA



$$\Rightarrow \quad \ell(\boldsymbol{\theta}) = \frac{1}{2}\|\overline{\boldsymbol{x}}(\boldsymbol{\theta}) - \boldsymbol{x}^*\|_2^2$$

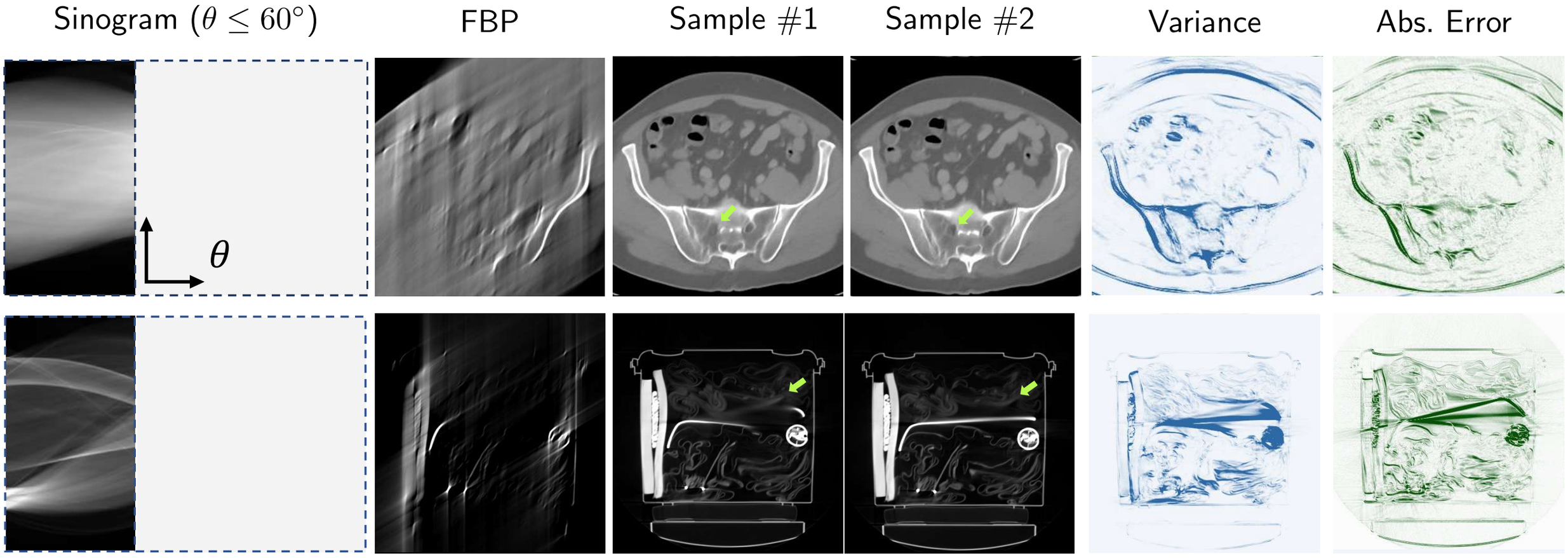
Gilton et al, "Deep equilibrium architectures for inverse problems in imaging," ***IEEE Trans Comput Imag***, 2021

PnP does not have to be restricted to PnP-ADMM, MAP, implicit priors, denoisers, and point estimates

- While PnP-ADMM was the first PnP method, there are variants based on other algorithms (such as PnP-FISTA)
- While original PnP is interpreted as using implicit priors, there are formulations based on explicit regularizers (such as RED)
- While the MAP view of the proximal operator is common, CNNs are often trained to act as MMSE estimators over the training data
- While original PnP was restricted to image denoisers, there are tools for efficiently learning priors end-to-end (such as DEQ)
- While original PnP was restricted to point estimates, there are tools for efficiently sampling from priors (such as PnP-ULA)

Laumont et al, "Bayesian imaging using Plug & Play priors: when Langevin meets Tweedie," ***SIAM J Imag Science***, 2022

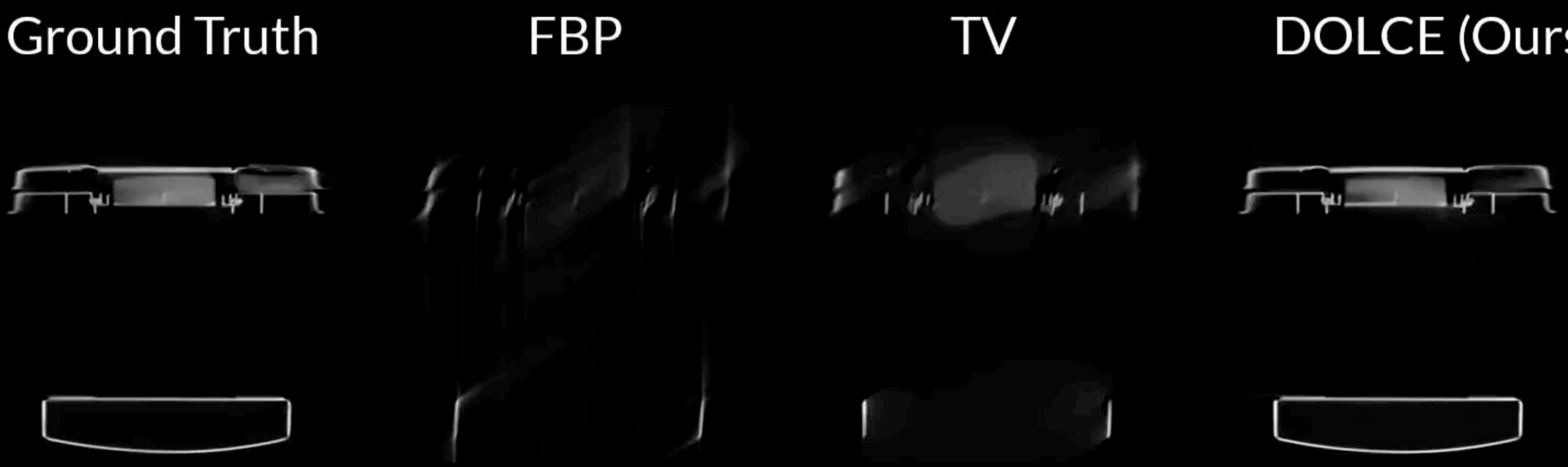
Training PnP denoisers as **diffusion models** can enable sampling realistic images with little data



Ground-truth suitcaise

Liu et al, "DOLCE: A Model-Based Probabilistic Diffusion Framework for Limited-Angle CT Reconstruction," ***arXiv:2211.12340***, 2022

Training PnP denoisers as diffusion models can enable sampling realistic images with little data



Limited-angle CT with $\theta_{\text{max}} \leq 60^{\circ}$. Note how DOLCE can synthesize realistic-looking 3D reconstruction from very little data!

Liu et al, "DOLCE: A Model-Based Probabilistic Diffusion Framework for Limited-Angle CT Reconstruction," ***arXiv:2211.12340***, 2022

Outline for the rest of the talk

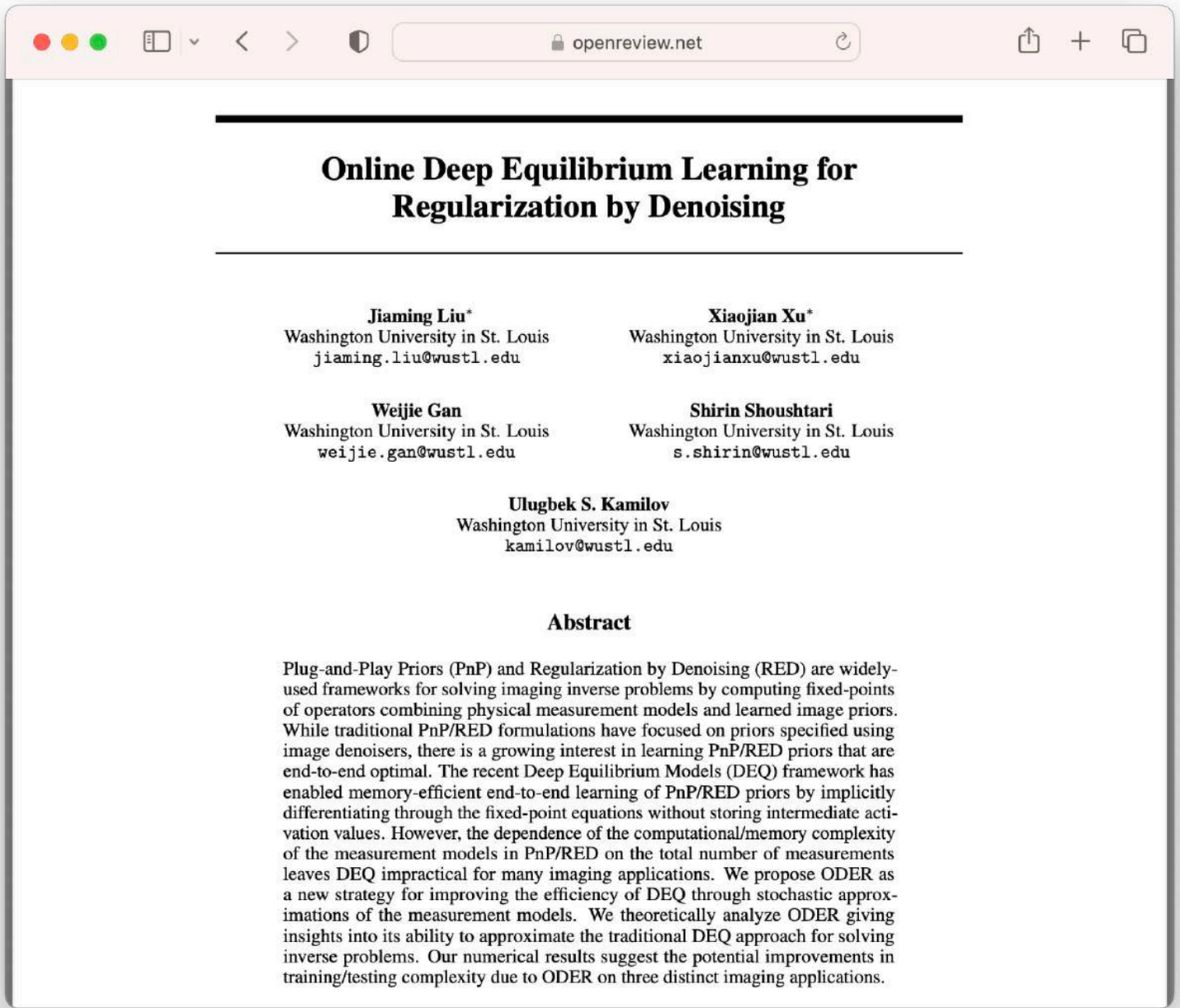
- Plug-and-Play Methods for Inverse Problems (**IEEE SPM 2022**)
  Integrating physical models and learned deep priors

- Online Deep Equilibrium Learning (**NeurIPS 2022**)
  A new PnP framework for efficient prior learning

- Deep Continuous Artifact-Free Fields (**Nature MI 2022**)
  A new PnP framework for continuous image recovery

Washington University in St. Louis

ODER is a PnP method for learning end-to-end optimal image priors for **large-scale problems**



openreview.net

Online Deep Equilibrium Learning for Regularization by Denoising

Jiaming Liu*
Washington University in St. Louis
jiaming.liu@wustl.edu

Xiaojian Xu*
Washington University in St. Louis
xiaojianxu@wustl.edu

Weijie Gan
Washington University in St. Louis
weijie.gan@wustl.edu

Shirin Shoushtari
Washington University in St. Louis
s.shirin@wustl.edu

Ulugbek S. Kamilov
Washington University in St. Louis
kamilov@wustl.edu

Abstract

Plug-and-Play Priors (PnP) and Regularization by Denoising (RED) are widely-used frameworks for solving imaging inverse problems by computing fixed-points of operators combining physical measurement models and learned image priors. While traditional PnP/RED formulations have focused on priors specified using image denoisers, there is a growing interest in learning PnP/RED priors that are end-to-end optimal. The recent Deep Equilibrium Models (DEQ) framework has enabled memory-efficient end-to-end learning of PnP/RED priors by implicitly differentiating through the fixed-point equations without storing intermediate activation values. However, the dependence of the computational/memory complexity of the measurement models in PnP/RED on the total number of measurements leaves DEQ impractical for many imaging applications. We propose ODER as a new strategy for improving the efficiency of DEQ through stochastic approximations of the measurement models. We theoretically analyze ODER giving insights into its ability to approximate the traditional DEQ approach for solving inverse problems. Our numerical results suggest the potential improvements in training/testing complexity due to ODER on three distinct imaging applications.

Scan the QR code to see the paper

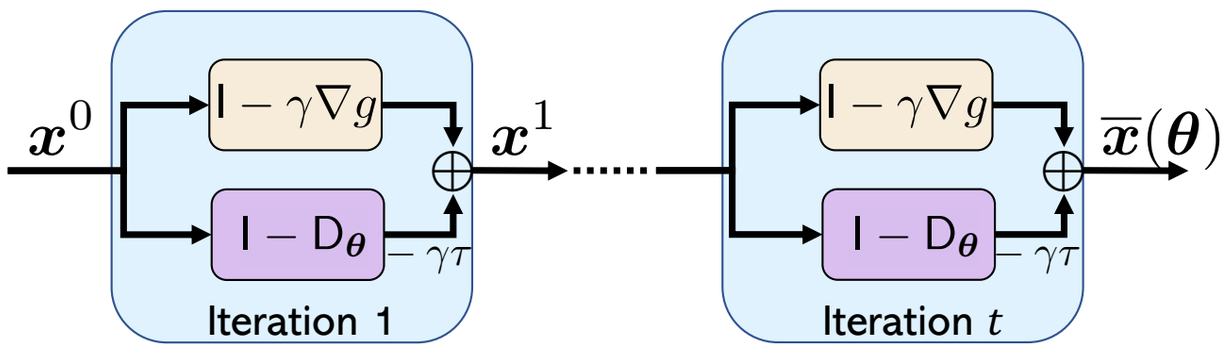
Liu et al, "Online Deep Equilibrium Learning for Regularization by Denoising," ***Proc NeurIPS***, 2022

PnP models achieve **SOTA performance**
when the image prior is trained at the fixed points

PnP models achieve SOTA performance when the image prior is trained at the fixed points

PnP can be interpreted as a **model-based implicit network**

$$\boldsymbol{x}^t \leftarrow \mathsf{T}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma \left(\nabla g(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1})) \right)$$



Two types of operations:

physical forward model

learned CNN prior

PnP models achieve SOTA performance when the image prior is trained at the fixed points

PnP can be interpreted as a model-based implicit network

$$\boldsymbol{x}^t \leftarrow \mathsf{T}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma \left(\nabla g(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1})) \right)$$

Performance of PnP models is significantly improved by going beyond AWGN denoisers to **artifact removing (AR) priors**

$\mathsf{D}_{\boldsymbol{\theta}}$: more image artifacts $\mapsto$ less image artifacts

Remark: AWGN denoisers are suboptimal at intermediate PnP layers!

PnP models achieve SOTA performance when the image prior is trained at the fixed points

PnP can be interpreted as a model-based implicit network

$$\boldsymbol{x}^t \leftarrow \mathsf{T}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma \left(\nabla g(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1})) \right)$$

Performance of PnP models is significantly improved by going beyond AWGN denoisers to artifact removing (AR) priors

$\mathsf{D}_{\boldsymbol{\theta}}$: more image artifacts $\mapsto$ less image artifacts

Deep equilibrium models (DEQ) enable efficient training of AR operators at PnP fixed points using implicit differentiation

$$\ell(\boldsymbol{\theta}) = \frac{1}{2}\|\overline{\boldsymbol{x}}(\boldsymbol{\theta}) - \boldsymbol{x}^*\|_2^2 \qquad \overline{\boldsymbol{x}}(\boldsymbol{\theta}) \in \mathsf{Fix}(\mathsf{T}_{\boldsymbol{\theta}})$$

$$\Rightarrow \quad \nabla \ell(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{\theta}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{\mathsf{T}} (\mathsf{I} - \nabla_{\boldsymbol{x}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{-\mathsf{T}} (\overline{\boldsymbol{x}} - \boldsymbol{x}^*)$$

Gilton et al, "Deep equilibrium architectures for inverse problems in imaging," ***IEEE Trans Comput Imag***, 2021

PnP models achieve SOTA performance when the image prior is trained at the fixed points

PnP can be interpreted as a model-based implicit network

$$\boldsymbol{x}^t \leftarrow \mathsf{T}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma \left(\nabla g(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1})) \right) \quad \text{forward pass}$$

Performance of PnP models is significantly improved by going beyond AWGN denoisers to artifact removing (AR) priors

$$\mathsf{D}_{\boldsymbol{\theta}} : \text{more image artifacts} \mapsto \text{less image artifacts}$$

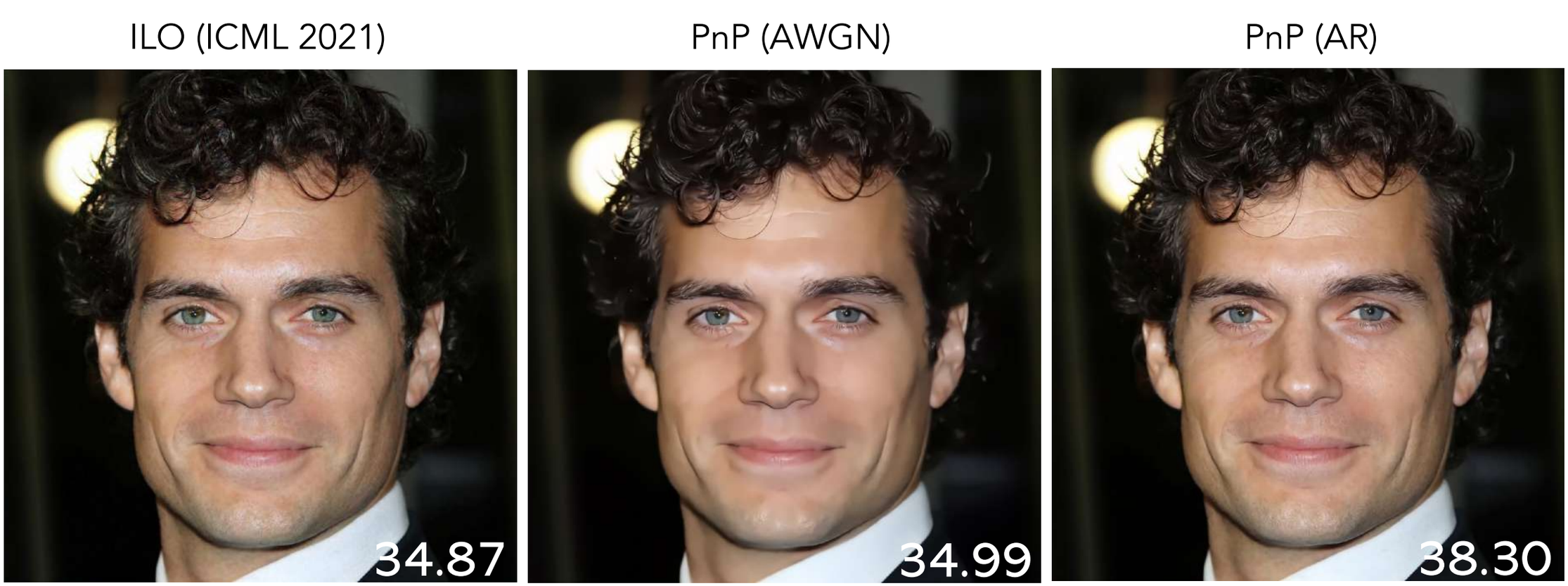
Deep equilibrium models (DEQ) enable efficient training of AR operators at PnP fixed points using implicit differentiation

$$\ell(\boldsymbol{\theta}) = \frac{1}{2}\|\overline{\boldsymbol{x}}(\boldsymbol{\theta}) - \boldsymbol{x}^*\|_2^2 \qquad \overline{\boldsymbol{x}}(\boldsymbol{\theta}) \in \mathsf{Fix}(\mathsf{T}_{\boldsymbol{\theta}})$$

$$\Rightarrow \quad \nabla \ell(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{\theta}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{\mathsf{T}} (\mathsf{I} - \nabla_{\boldsymbol{x}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{-\mathsf{T}} (\overline{\boldsymbol{x}} - \boldsymbol{x}^*)$$

$$\Rightarrow \quad \boldsymbol{b}^t \leftarrow \mathsf{F}(\boldsymbol{b}^{t-1}) = (\nabla_{\boldsymbol{x}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{\mathsf{T}} \boldsymbol{b}^{t-1} + (\overline{\boldsymbol{x}} - \boldsymbol{x}^*) \quad \text{backward pass}$$

PnP models achieve SOTA performance when the image prior is trained at the fixed points



Example: In compressive sensing from random projections with 10% subsampling, PnP (AR) significantly outperforms PnP (AWGN)!

Liu et al, "Recovery Analysis for Plug-and-Play Priors using the Restricted Eigenvalue Condition," ***Proc. NeurIPS***, 2021

PnP models achieve SOTA performance when the image prior is trained at the fixed points

ILO (ICML 2021)

PnP (AWGN)

PnP (AR)

Example: In compressive sensing from random projections with 10% subsampling, PnP (AR) significantly outperforms PnP (AWGN)!

Liu et al, "Recovery Analysis for Plug-and-Play Priors using the Restricted Eigenvalue Condition," ***Proc. NeurIPS***, 2021

PnP models achieve SOTA performance when the image prior is trained at the fixed points

Table 3: Average PSNR (dB) values for several algorithms on test images from CelebA HQ.

Method \ CS Ratio	10%	20%	30%	40%	50%
TV	32.13	35.24	37.41	39.35	41.29
PULSE [36]	27.45	29.98	33.06	34.25	34.77
ILO [37]	36.15	40.98	43.46	47.89	48.21
PnP (denoising)	35.61	41.51	45.71	48.05	52.24
PnP (AR)	**39.19**	**44.20**	**48.66**	**51.32**	**53.89**

Example: In compressive sensing from random projections with 10% subsampling, PnP (AR) significantly outperforms PnP (AWGN)!

Liu et al, "Recovery Analysis for Plug-and-Play Priors using the Restricted Eigenvalue Condition," ***Proc. NeurIPS***, 2021

Training **deep equilibrium models** becomes expensive as the **number of measurements grows**

Training deep equilibrium models becomes expensive as the number of measurements grows

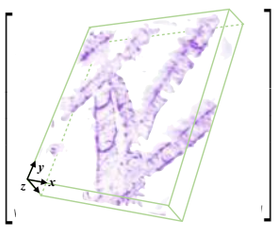
Consider **data-fidelity terms** that can be expressed as

$$g(\boldsymbol{x}) = \frac{1}{b}\sum_{i=1}^{b} g_i(\boldsymbol{x})$$

each term g_i depends on a subset of measurements $\boldsymbol{y}_i$

Example in IDT: $g_i(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{A}_i(\boldsymbol{x})\|_2^2$

$$\boldsymbol{y} = \left[\; \boldsymbol{y}_1 \;\cdots\; \boldsymbol{y}_b \;\right] \qquad \boldsymbol{x} =$$



Training deep equilibrium models becomes expensive as the number of measurements grows

Consider data-fidelity terms that can be expressed as

$$g(\boldsymbol{x}) = \frac{1}{b}\sum_{i=1}^{b} g_i(\boldsymbol{x})$$

DEQ gradient becomes impractical to compute when the number of measurements grows

$$\nabla \ell(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{\theta}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{\mathsf{T}} \overline{\boldsymbol{b}}$$

$$\boldsymbol{x}^t \leftarrow \mathsf{T}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma \left(\nabla g(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1})) \right)$$

$$\boldsymbol{b}^t \leftarrow \mathsf{F}(\boldsymbol{b}^{t-1}) = (\nabla_{\boldsymbol{x}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{\mathsf{T}} \boldsymbol{b}^{t-1} + (\overline{\boldsymbol{x}} - \boldsymbol{x}^*)$$

impractical per-iteration computational and memory complexity when $b \to \infty$

Training deep equilibrium models becomes expensive as the number of measurements grows

Consider data-fidelity terms that can be expressed as

$$g(\boldsymbol{x}) = \frac{1}{b}\sum_{i=1}^{b} g_i(\boldsymbol{x})$$

DEQ gradient becomes impractical to compute when the number of measurements grows

$$\nabla \ell(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{\theta}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{\mathsf{T}} \overline{\boldsymbol{b}}$$

$$\boldsymbol{x}^t \leftarrow \mathsf{T}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma \left(\nabla g(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{R}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1})) \right)$$

$$\boldsymbol{b}^t \leftarrow \mathsf{F}(\boldsymbol{b}^{t-1}) = (\nabla_{\boldsymbol{x}} \mathsf{T}_{\boldsymbol{\theta}}(\overline{\boldsymbol{x}}))^{\mathsf{T}} \boldsymbol{b}^{t-1} + (\overline{\boldsymbol{x}} - \boldsymbol{x}^*)$$

Question: Can we train AR priors for PnP with complexity independent of b?

Stochastic PnP methods are useful for reducing the per-iteration cost **during inference**

Stochastic PnP methods are useful for reducing the per-iteration cost during inference

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w} \sum_{s=1}^{w} g_{i_s}(\boldsymbol{x})$$

- $w \ll b$ is the mini-batch size
- complexity independent of b

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

Wu et al, "SIMBA: Scalable Inversion in Optical Tomography using Deep Denoising Priors," ***IEEE J Sel Topics Signal Process***, 2020

Stochastic PnP methods are useful for reducing the per-iteration cost during inference

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w} \sum_{s=1}^{w} g_{i_s}(\boldsymbol{x})$$

$$\mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

if $\{i_1, \ldots, i_w\}$ are iid uniform random variables

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

Wu et al, "SIMBA: Scalable Inversion in Optical Tomography using Deep Denoising Priors," ***IEEE J Sel Topics Signal Process***, 2020

Stochastic PnP methods are useful for reducing the per-iteration cost during inference

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w}\sum_{s=1}^{w} g_{i_s}(\boldsymbol{x}) \qquad \mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

if $\{i_1, \ldots, i_w\}$ are iid uniform random variables

SIMBA is a type of PnP-SGD algorithm

$$\boldsymbol{x}^t \leftarrow \widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma\left(\nabla\widehat{g}(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}))\right)$$

Remark: Complexity of the SIMBA iteration is independent of b!

Wu et al, "SIMBA: Scalable Inversion in Optical Tomography using Deep Denoising Priors," ***IEEE J Sel Topics Signal Process***, 2020

Stochastic PnP methods are useful for reducing the per-iteration cost during inference

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w}\sum_{s=1}^{w} g_{i_s}(\boldsymbol{x}) \qquad \mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

if $\{i_1, \ldots, i_w\}$ are iid uniform random variables

SIMBA is a type of PnP-SGD algorithm

$$\boldsymbol{x}^t \leftarrow \widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma\left(\nabla\widehat{g}(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}))\right)$$

A theoretical error bound on the convergence of SIMBA

$$\mathbb{E}\left[\|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2^2\right] \leq \eta^k R + \frac{\gamma\nu}{(1-\eta)\sqrt{w}}$$

- $0 < \eta < 1$
- $\bar{\boldsymbol{x}} \in \mathsf{Fix}(\mathsf{T})$

Liu et al, "Online Deep Equilibrium Learning for Regularization by Denoising," ***Proc NeurIPS***, 2022

Stochastic PnP methods are useful for reducing the per-iteration cost during inference

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w}\sum_{s=1}^{w} g_{i_s}(\boldsymbol{x}) \qquad \mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

if $\{i_1, \ldots, i_w\}$ are iid uniform random variables

SIMBA is a type of PnP-SGD algorithm

$$\boldsymbol{x}^t \leftarrow \widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma\left(\nabla\widehat{g}(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}))\right)$$

A theoretical error bound on the convergence of SIMBA

$$\mathbb{E}\left[\|\boldsymbol{x}^k - \bar{\boldsymbol{x}}\|_2^2\right] \leq \eta^k R + \frac{\gamma\nu}{(1-\eta)\sqrt{w}}$$

Remark: SIMBA can in expectation approximate the fixed points of the full PnP algorithm up to an error term that depends on γ and w!

ODER extends stochastic PnP for **efficient training** of AR priors with theoretical guarantees

ODER extends stochastic PnP for efficient training of AR priors with theoretical guarantees

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w} \sum_{s=1}^{w} g_{i_s}(\boldsymbol{x})$$

$$\mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

Liu et al, "Online Deep Equilibrium Learning for Regularization by Denoising," ***Proc NeurIPS***, 2022

ODER extends stochastic PnP for efficient training of AR priors with theoretical guarantees

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w}\sum_{s=1}^{w} g_{i_s}(\boldsymbol{x}) \qquad\qquad \mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

ODER seeks to approximate the implicit gradient as

$$\nabla\hat{\ell}(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{\theta}}\widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^T))^{\mathsf{T}}\boldsymbol{b}^T \qquad T \geq 1 \text{ forward and backward iterations}$$

$$\boldsymbol{x}^t \leftarrow \widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma\left(\nabla\widehat{g}(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{R}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}))\right)$$

$$\boldsymbol{b}^t \leftarrow \widehat{\mathsf{F}}(\boldsymbol{b}^{t-1}) = (\nabla_{\boldsymbol{x}}\widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^T))^{\mathsf{T}}\boldsymbol{b}^{t-1} + (\boldsymbol{x}^T - \boldsymbol{x}^*)$$

Liu et al, "Online Deep Equilibrium Learning for Regularization by Denoising," ***Proc NeurIPS***, 2022

ODER extends stochastic PnP for efficient training of AR priors with theoretical guarantees

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w}\sum_{s=1}^{w} g_{i_s}(\boldsymbol{x}) \qquad\qquad \mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

ODER seeks to approximate the implicit gradient as

$$\nabla\hat{\ell}(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{\theta}}\widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^T))^{\mathsf{T}}\boldsymbol{b}^T$$

$$\boldsymbol{x}^t \leftarrow \widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma\left(\nabla\widehat{g}(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{R}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}))\right)$$

$$\boldsymbol{b}^t \leftarrow \widehat{\mathsf{F}}(\boldsymbol{b}^{t-1}) = (\nabla_{\boldsymbol{x}}\widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^T))^{\mathsf{T}}\boldsymbol{b}^{t-1} + (\boldsymbol{x}^T - \boldsymbol{x}^*)$$

Remark: Complexity of ODER is independent of b!

Liu et al, "Online Deep Equilibrium Learning for Regularization by Denoising," ***Proc NeurIPS***, 2022

ODER extends stochastic PnP for efficient training of AR priors with theoretical guarantees

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w}\sum_{s=1}^{w} g_{i_s}(\boldsymbol{x}) \qquad\qquad \mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

ODER seeks to approximate the implicit gradient as

$$\nabla\hat{\ell}(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{\theta}}\widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^T))^{\mathsf{T}}\boldsymbol{b}^T$$

$$\boldsymbol{x}^t \leftarrow \widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma\left(\nabla\widehat{g}(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{R}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}))\right)$$

$$\boldsymbol{b}^t \leftarrow \widehat{\mathsf{F}}(\boldsymbol{b}^{t-1}) = (\nabla_{\boldsymbol{x}}\widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^T))^{\mathsf{T}}\boldsymbol{b}^{t-1} + (\boldsymbol{x}^T - \boldsymbol{x}^*)$$

A **theoretical error bound** on SGD learning using ODER gradients

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla\ell(\boldsymbol{\theta}^k)\|_2^2\right] \leq \frac{2(\ell(\boldsymbol{\theta}^0) - \ell(\boldsymbol{\theta}^*))}{\beta K} + \frac{C_2}{\sqrt{w}} + \beta C_2$$

- $\beta > 0$ learning rate
- $K \geq 1$ SGD iterations

Liu et al, "Online Deep Equilibrium Learning for Regularization by Denoising," ***Proc NeurIPS***, 2022

ODER is a framework for efficiently training AR priors for PnP with theoretical guarantees

Consider a mini-batch approximation of the data-fidelity term

$$\widehat{g}(\boldsymbol{x}) = \frac{1}{w}\sum_{s=1}^{w} g_{i_s}(\boldsymbol{x}) \qquad\qquad \mathbb{E}\left[\widehat{g}(\boldsymbol{x})\right] = g(\boldsymbol{x})$$

ODER seeks to approximate the implicit gradient as

$$\nabla\hat{\ell}(\boldsymbol{\theta}) = (\nabla_{\boldsymbol{\theta}}\widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^T))^{\mathsf{T}}\boldsymbol{b}^T$$

$$\boldsymbol{x}^t \leftarrow \widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}) = \boldsymbol{x}^{t-1} - \gamma\left(\nabla\widehat{g}(\boldsymbol{x}^{t-1}) + \tau(\boldsymbol{x}^{t-1} - \mathsf{R}_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1}))\right)$$

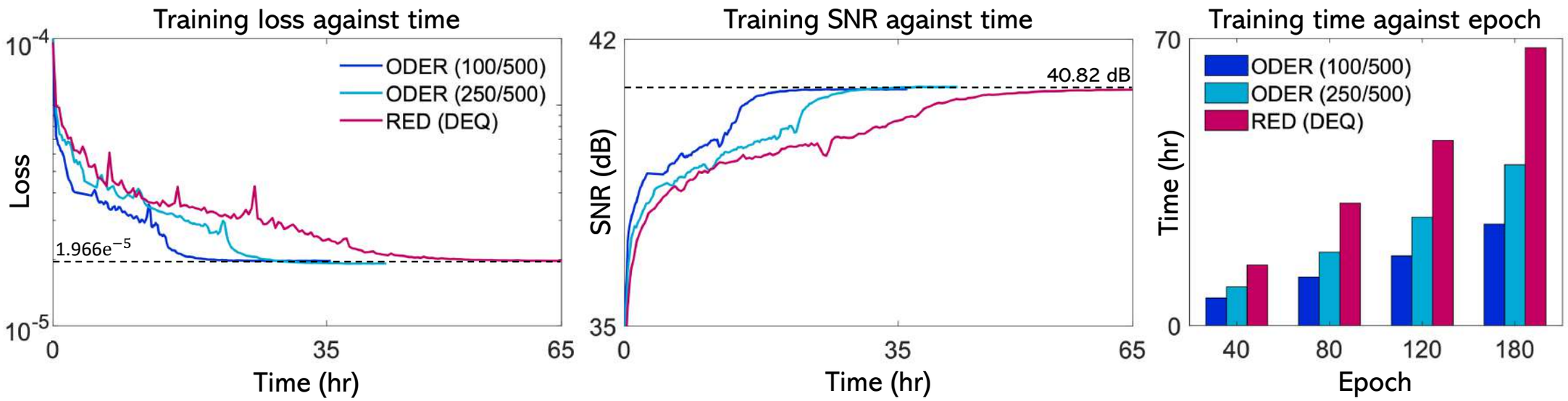
$$\boldsymbol{b}^t \leftarrow \widehat{\mathsf{F}}(\boldsymbol{b}^{t-1}) = (\nabla_{\boldsymbol{x}}\widehat{\mathsf{T}}_{\boldsymbol{\theta}}(\boldsymbol{x}^T))^{\mathsf{T}}\boldsymbol{b}^{t-1} + (\boldsymbol{x}^T - \boldsymbol{x}^*)$$

A **theoretical error bound** on SGD learning using ODER gradients

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla\ell(\boldsymbol{\theta}^k)\|_2^2\right] \leq \frac{2(\ell(\boldsymbol{\theta}^0) - \ell(\boldsymbol{\theta}^*))}{\beta K} + \frac{C_2}{\sqrt{w}} + \beta C_2$$

ODER can approximate the DEQ learning with controllable accuracy!

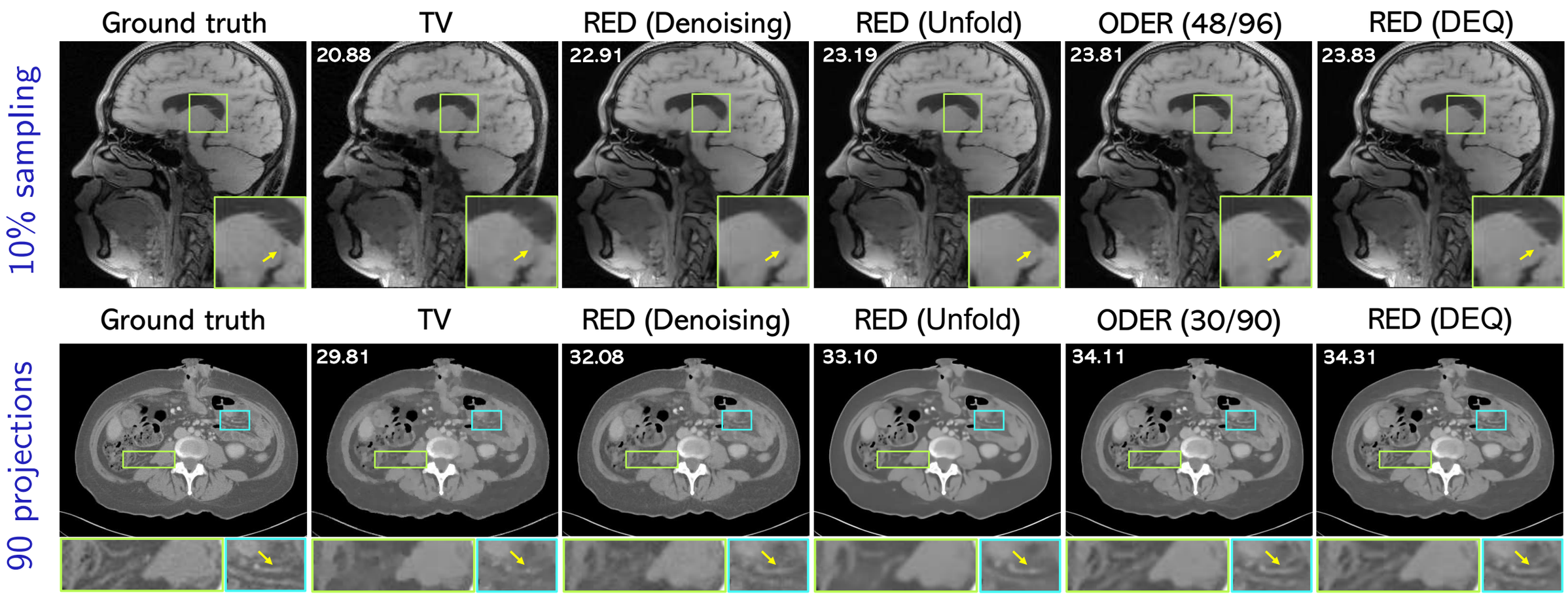
ODER achieves nearly **2.5x improvement in training time** on IDT for the same image quality



Remark: 2.5× Faster training in IDT for a similar loss and SNR!

Liu et al, "Online Deep Equilibrium Learning for Regularization by Denoising," ***Proc NeurIPS***, 2022

ODER achieves SOTA reconstruction performance in parallel MRI and sparse-view CT



Note the similar performance of ODER and RED (DEQ), and the improvement over RED (Denoising) and RED (Unfold) due to DEQ learning!

Liu et al, "Online Deep Equilibrium Learning for Regularization by Denoising," ***Proc NeurIPS***, 2022

Outline for the rest of the talk

- Plug-and-Play Methods for Inverse Problems (**IEEE SPM 2022**)
  Integrating physical models and learned deep priors

- Online Deep Equilibrium Learning (**NeurIPS 2022**)
  A new PnP framework for efficient prior learning

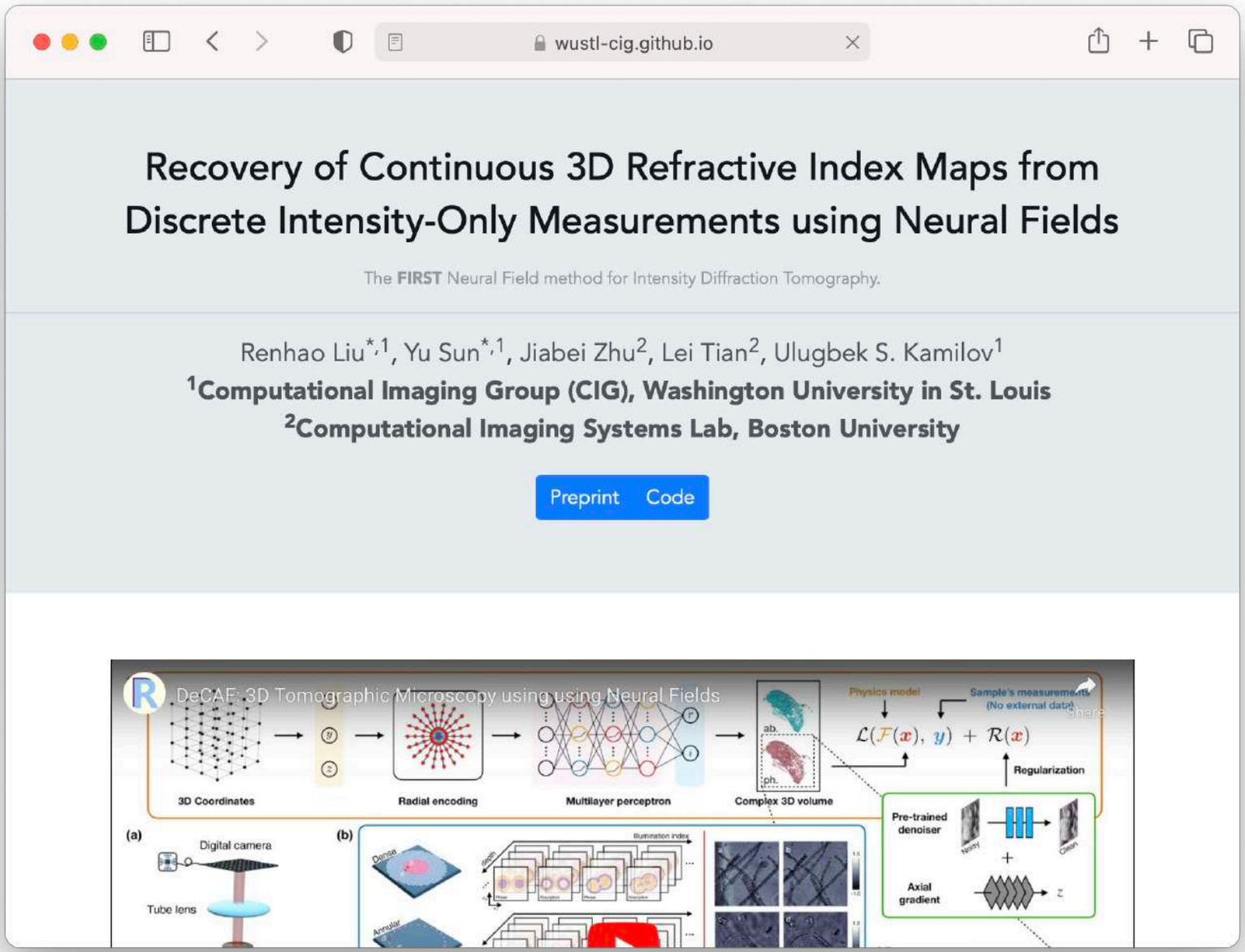
- Deep Continuous Artifact-Free Fields (**Nature MI 2022**)
  A new PnP framework for continuous image recovery

Washington University in St. Louis

DeCAF is a PnP method that enables the recovery of **continuous images** represented by neural fields

Washington University in St. Louis

DeCAF is a PnP method that enables the recovery of **continuous images** represented by neural fields



Scan the QR code to see the project

Liu et al, "Recovery of Continuous 3D Refractive Index Maps from Discrete Intensity-Only Measurements using Neural Fields," ***Nature Machine Intelligence***, 2022

DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

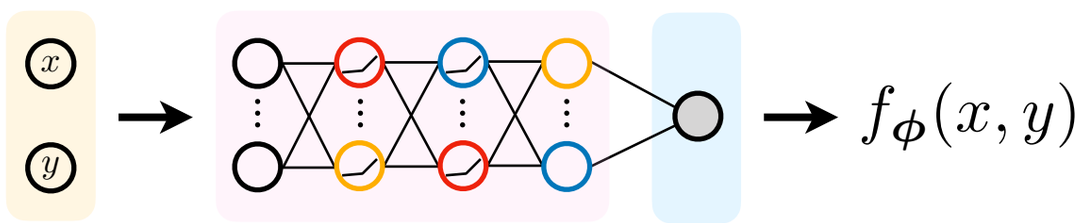
- Traditional PnP formulations seek to reconstruct images represented using a **pre-defined pixel grid**

$$\boldsymbol{x} = (x_1, \cdots, x_n)$$

DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

- Traditional PnP formulations seek to reconstruct images represented using a pre-defined pixel grid

- DeCAF is a variant of PnP that continuously represents the desired image by using a **coordinate-based neural network**



DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

- Traditional PnP formulations seek to reconstruct images represented using a pre-defined pixel grid

- DeCAF is a variant of PnP that continuously represents the desired image by using a coordinate-based neural network

- Continuous representation in DeCAF **decouples** the representation of the solution from any pre-defined pixel grid

DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

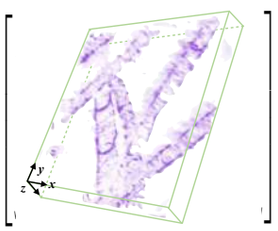
- Traditional PnP formulations seek to reconstruct images represented using a pre-defined pixel grid

- DeCAF is a variant of PnP that continuously represents the desired image by using a coordinate-based neural network

- Continuous representation in DeCAF decouples the representation of the solution from any pre-defined pixel grid

- We tested DeCAF on a large-scale 3D IDT problem, where it is difficult to have ground-truth training data for the refractive index

$$\boldsymbol{y} = \left[\boldsymbol{y}_1 \cdots \boldsymbol{y}_b \right] \qquad \boldsymbol{x} =$$



Washington University in St. Louis

DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

DeCAF maps **coordinates** to **image values** using a deep network

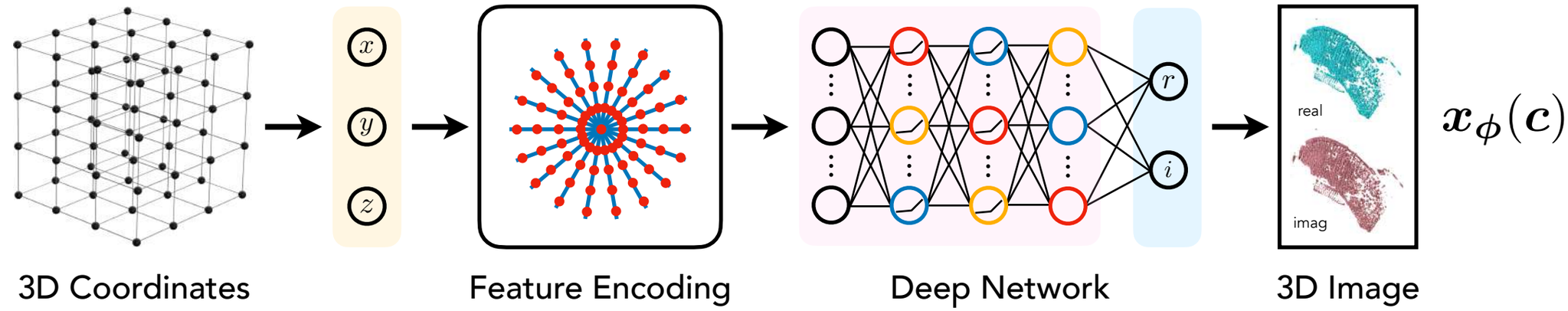
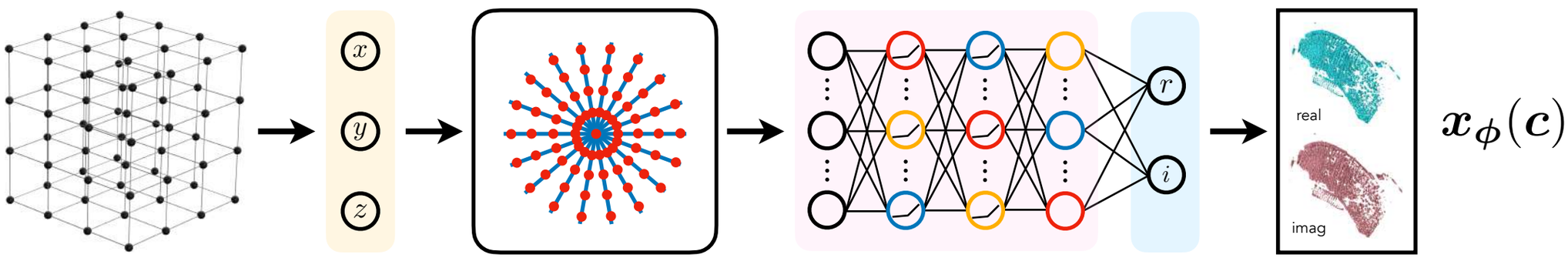


Image is represented as $\boldsymbol{x}_{\boldsymbol{\phi}}(\boldsymbol{c})$ where $\boldsymbol{\phi}$ are weights and $\boldsymbol{c} = \{(x_i, y_i, z_i)\}$ are coordinates

Washington University in St. Louis

DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

DeCAF maps coordinates to image values using a deep network



3D Coordinates | Feature Encoding | Deep Network | 3D Image

Image formation is formulated as a stochastic PnP algorithm

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \left\{ \frac{1}{b} \sum_{i=1}^{b} g_i(\boldsymbol{x}_{\boldsymbol{\phi}}) + h(\boldsymbol{x}_{\boldsymbol{\phi}}) \right\}$$

Minimization over network weights; not image pixel values

Quadratic data-fidelity

$$g(\boldsymbol{x}_{\boldsymbol{\phi}}) = \frac{1}{2} \left\| \boldsymbol{y}_i - \boldsymbol{A}_i(\boldsymbol{x}_{\boldsymbol{\phi}}) \right\|_2^2$$

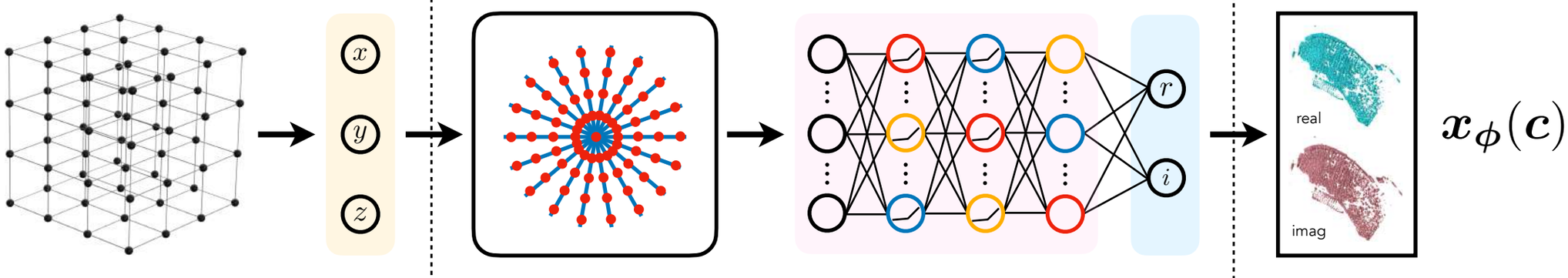
Pre-trained deep prior

$$h(\boldsymbol{x}_{\boldsymbol{\phi}}) = \frac{\tau}{2} \left\| \boldsymbol{x}_{\boldsymbol{\phi}} - \mathsf{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\boldsymbol{\phi}}) \right\|_2^2$$

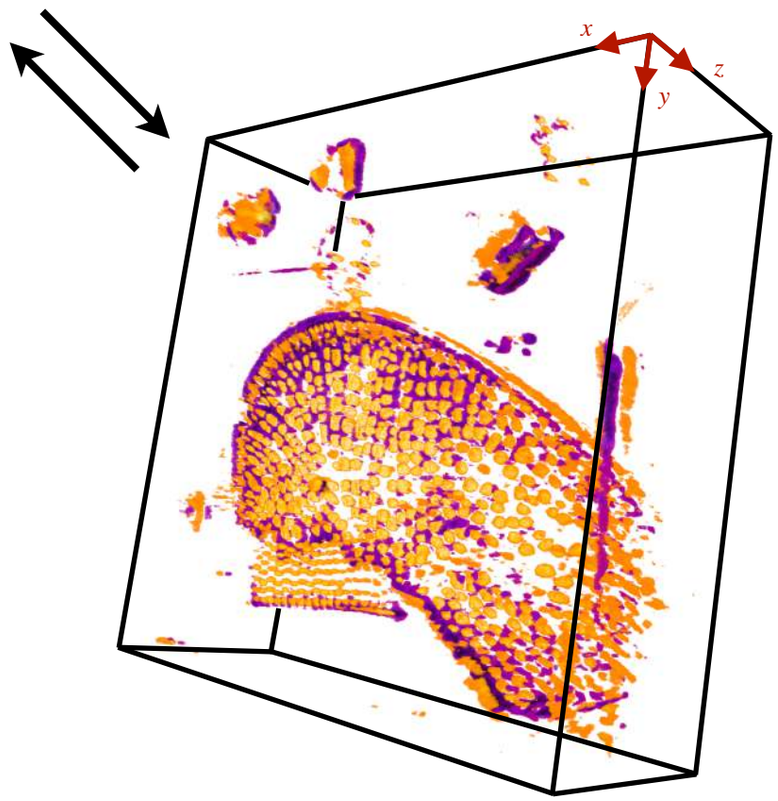
Washington University in St. Louis

DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

DeCAF maps coordinates to image values using a deep network

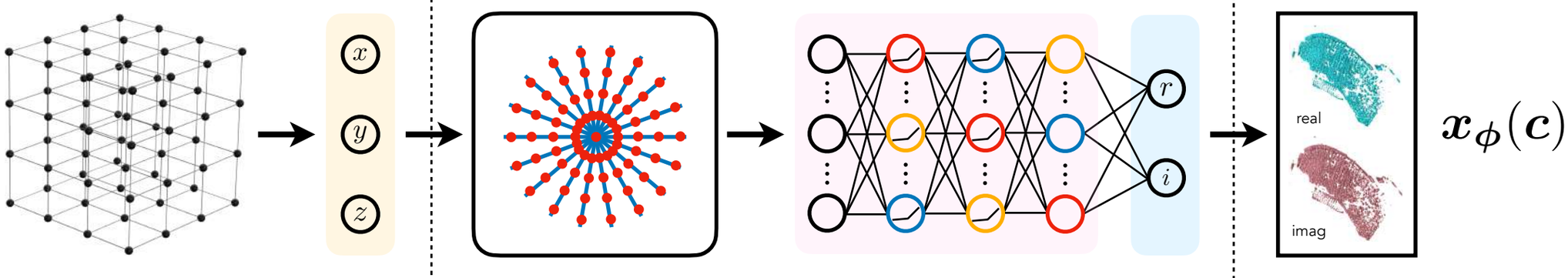


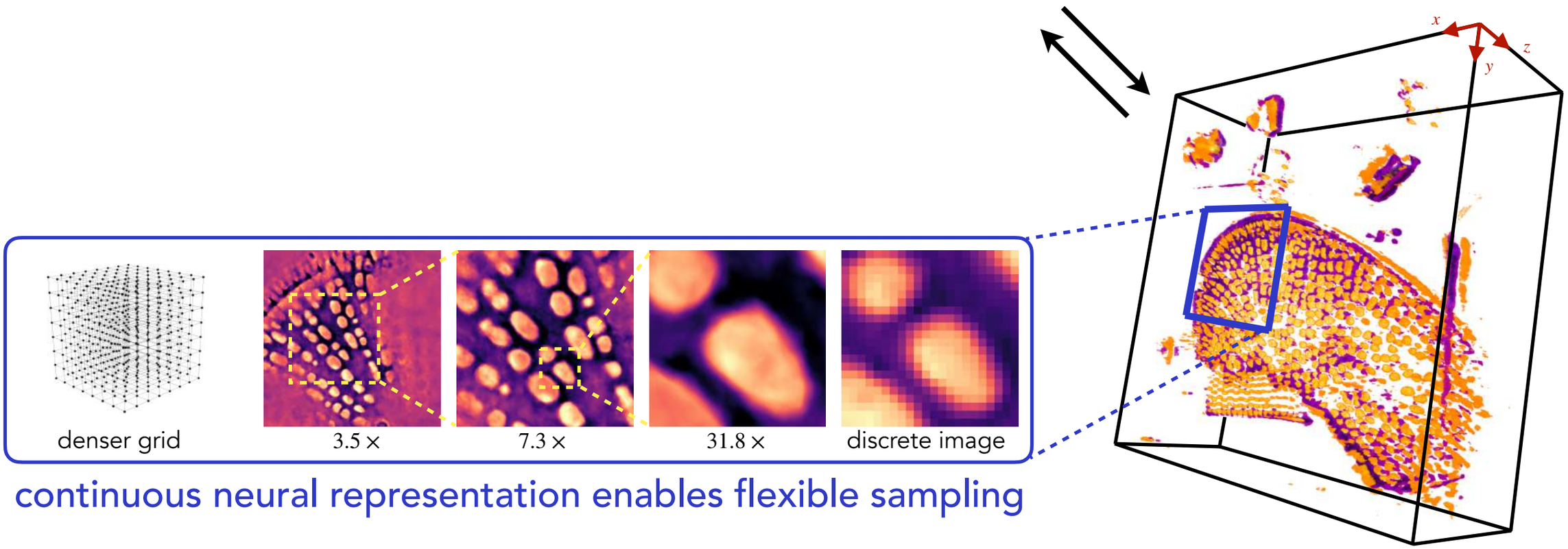
3D Coordinates | Feature Encoding | Deep Network | 3D Image



DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

DeCAF maps coordinates to image values using a deep network





continuous neural representation enables flexible sampling

Washington University in St. Louis

DeCAF is a PnP method that enables the recovery of continuous images represented by neural fields

DeCAF maps coordinates to image values using a deep network

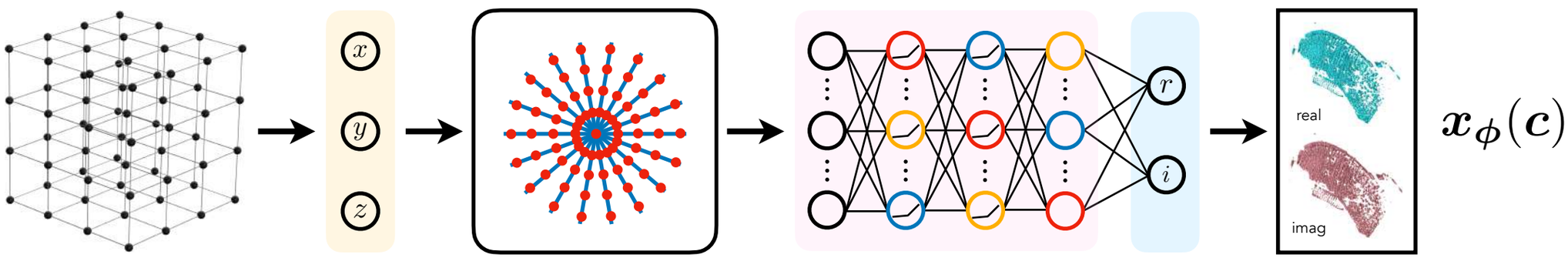


Image formation is performed by integrating DeCAF into PnP

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \{g(\boldsymbol{x}_{\boldsymbol{\phi}}) + h(\boldsymbol{x}_{\boldsymbol{\phi}})\}$$

The key benefits of DeCAF include:

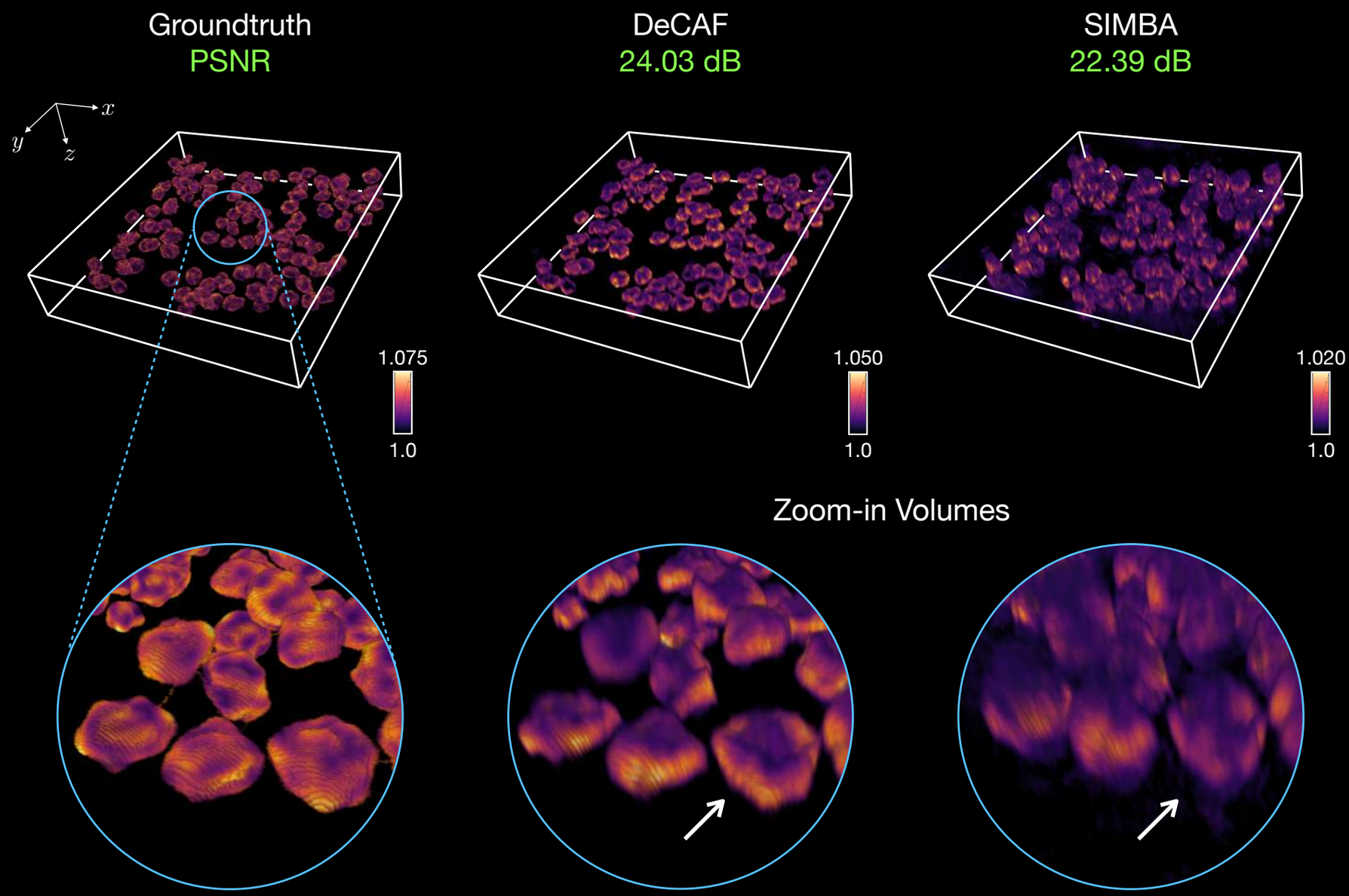
(a) representation is decoupled from any pre-defined voxel grid

(b) subimages can be synthesized at will during optimization

(c) image prior is trained on natural images

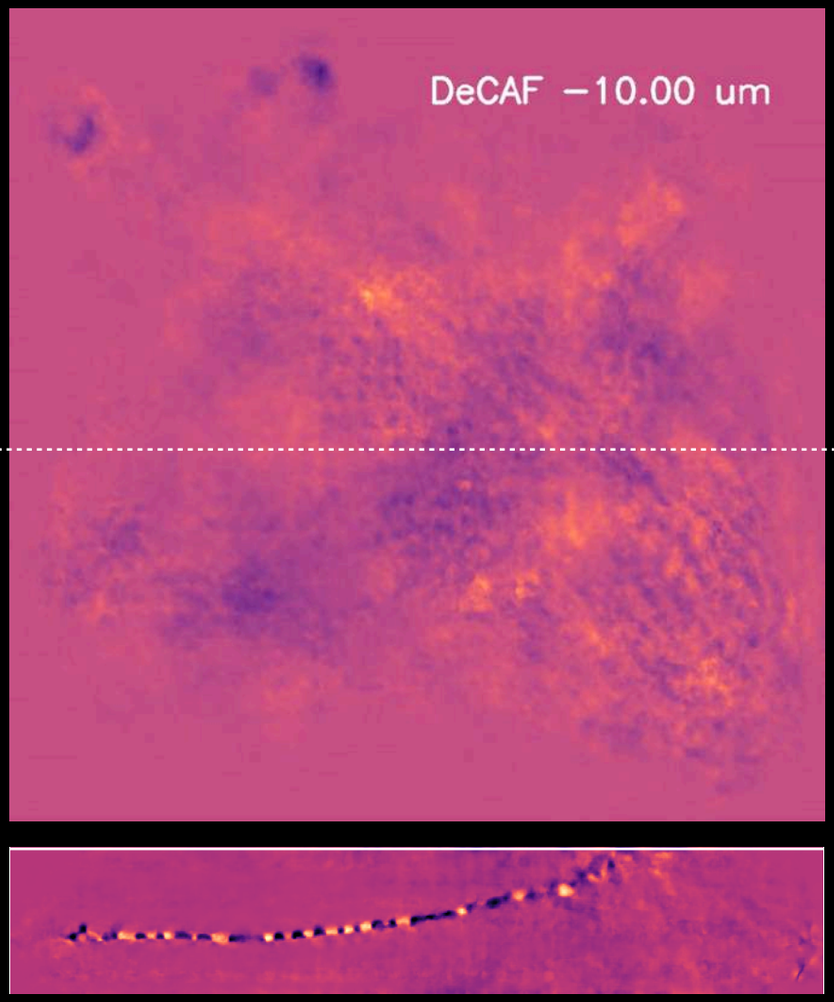
DeCAF can recover high-quality 3D RI volumes from experimentally collected 2D IDT images

DeCAF can recover high-quality 3D RI volumes from experimentally collected 2D IDT images

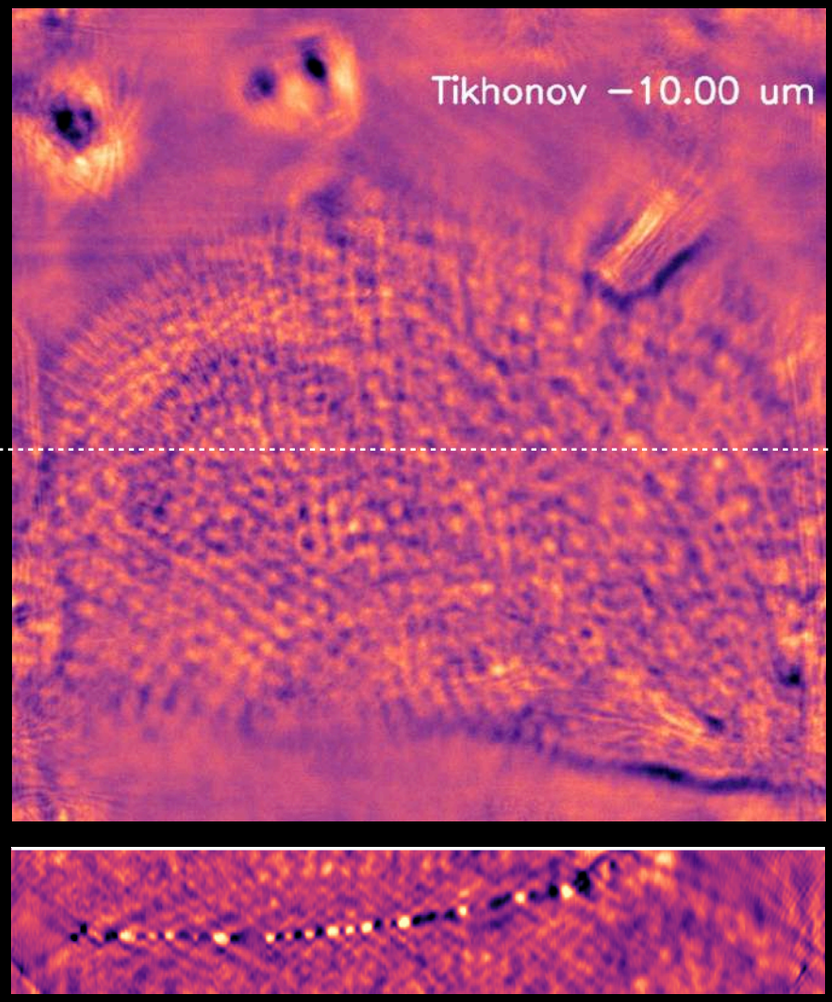


Note how DeCAF outperforms SIMBA by about $1.5\times$ dB on simulated data; SIMBA is a traditional (discrete) online PnP method!

DeCAF can recover high-quality 3D RI volumes from experimentally collected 2D IDT images



DeCAF (ours)



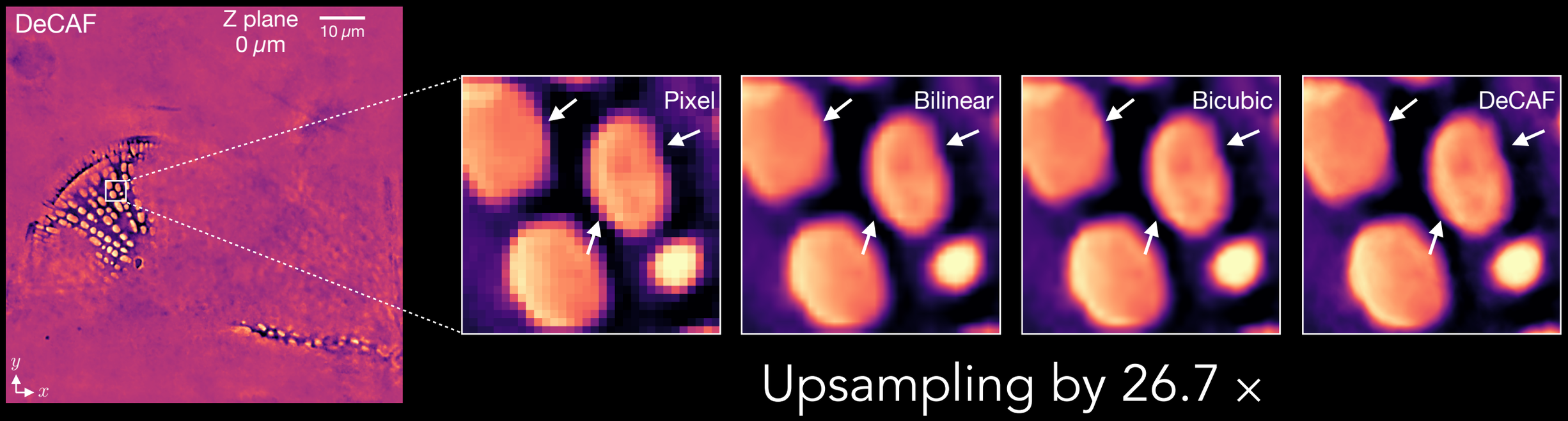
[Ling et al 2019]

Lateral view

Lateral view

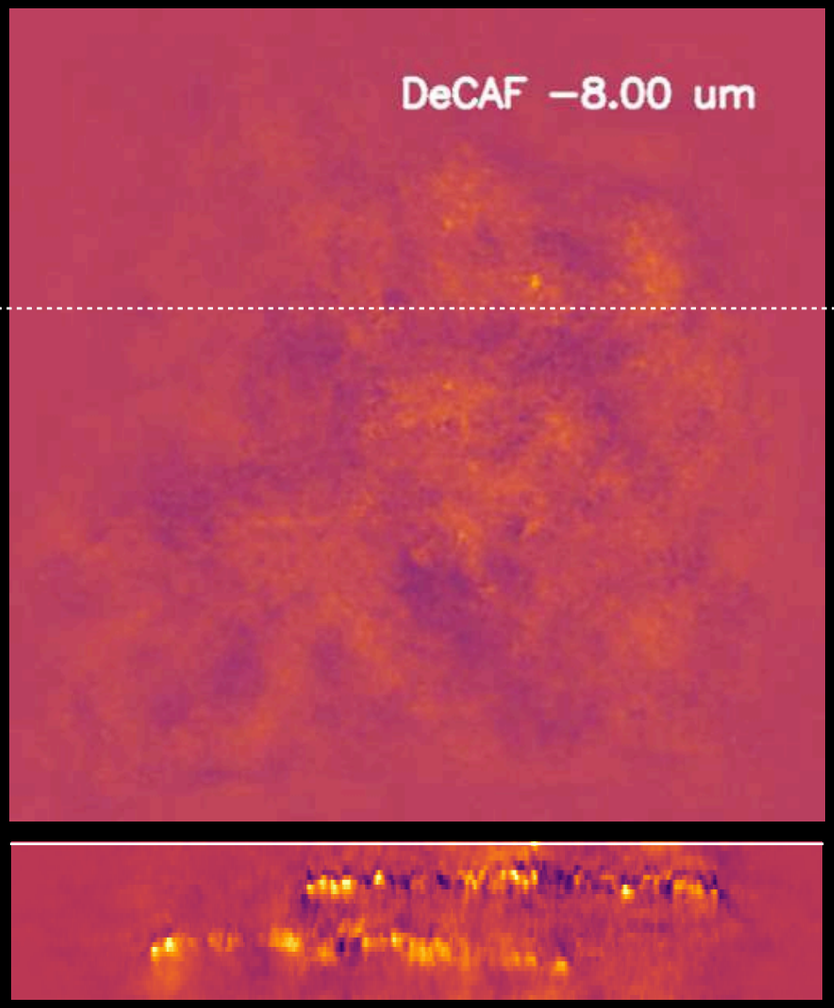
Note the optical sectioning ability of DeCAF on this Diatom Algae sample!

DeCAF can recover high-quality 3D RI volumes from experimentally collected 2D IDT images

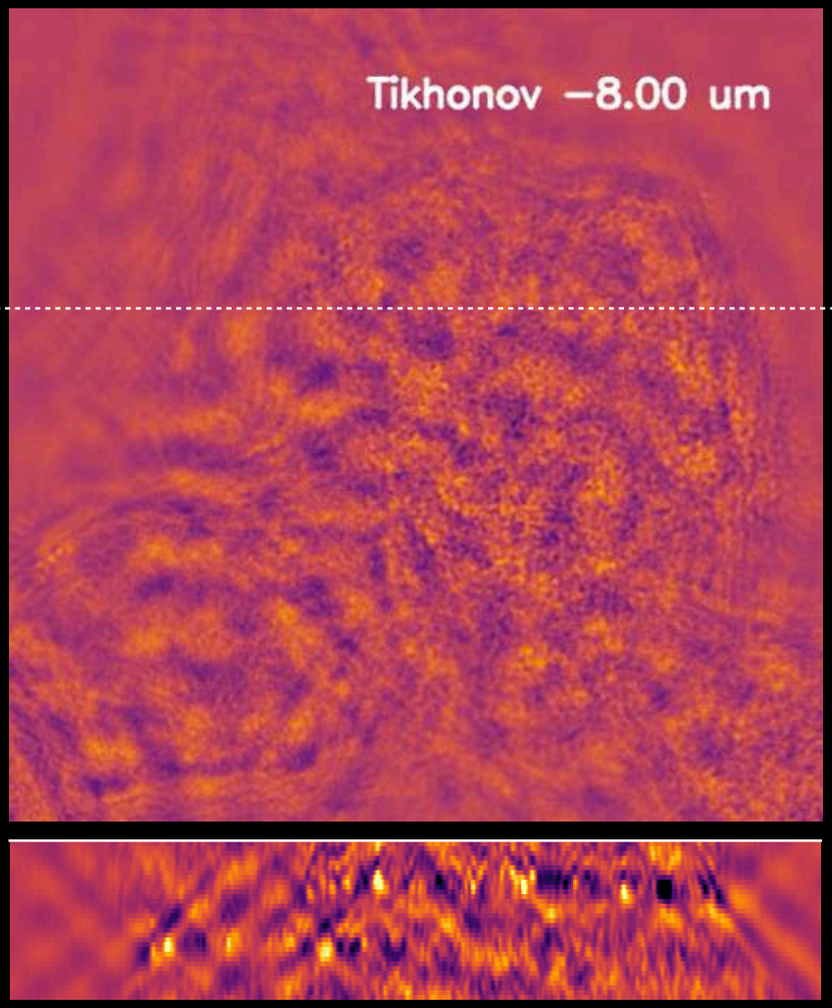


Note how DeCAF can synthesize different parts of this Diatom Algae sample on any grid!

DeCAF can recover high-quality 3D RI volumes from experimentally collected 2D IDT images



DeCAF (ours)

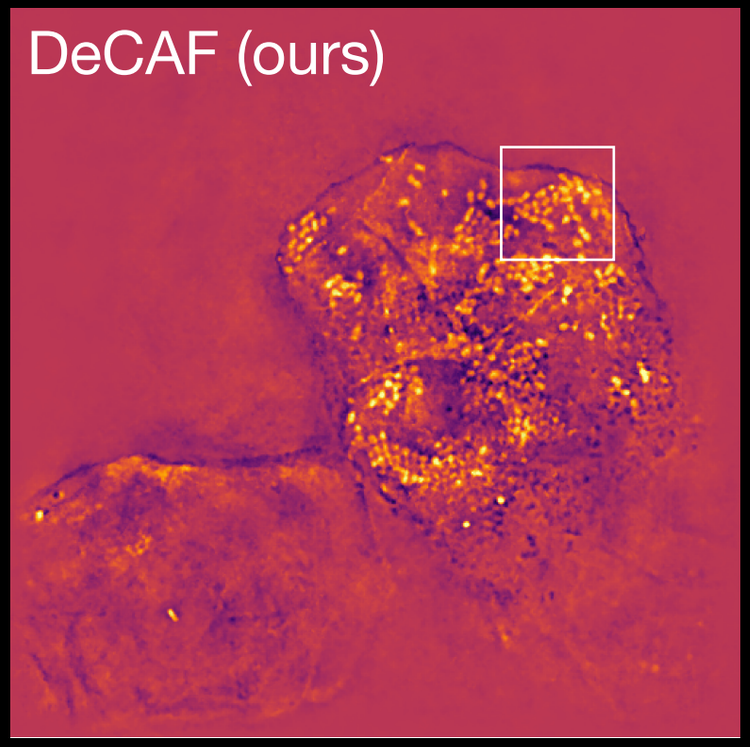


[Ling et al 2019]

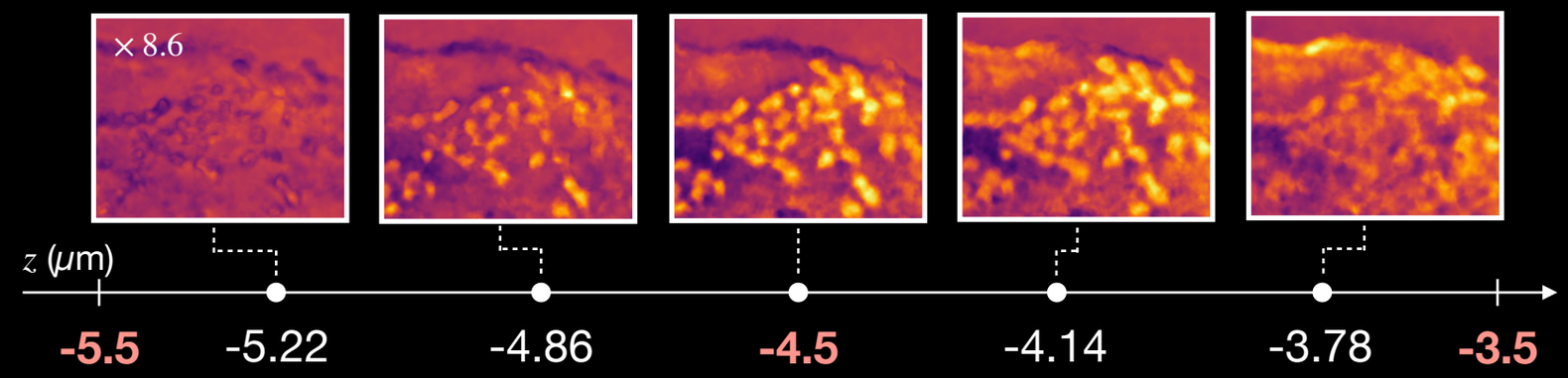
Lateral view

Note the optical sectioning ability of DeCAF on this Human Buccal Epithelial Cells sample!

DeCAF can recover high-quality 3D RI volumes from experimentally collected 2D IDT images



Continuous interpolation along z dimension



Note how DeCAF can be used to synthesize any slice of this Human Buccal Epithelial Cells sample along the z dimension!

To conclude

- **PnP** is one of the most influential class of methods across computational imaging and inverse problems

- We presented three recent extensions of the PnP models for solving large-scale inverse problems in IDT, CT, and MRI

- **ODER** is a framework for training implicit online neural nets with theoretical error bounds on the accuracy

- **DeCAF** is a PnP framework for directly reconstructing an image continuously represented by a neural field

- **DOLCE** is a conditional diffusion model for sampling realistic-looking images from few limited-angle CT data

Washington University in St. Louis

Big **Thank You** to my team at the WashU **Computational Imaging Group (CIG)**

My Twitter: **@ukmlv**
Web: **cigroup.wustl.edu**
Group Twitter: **@wustlcig**
GitHub: **github.com/wustl-cig**

NSF · NIH National Institutes of Health · CZ · Department of Energy · Research Corporation for Science Advancement · leap