Efficient generative models for imaging inverse problems

Jeffrey A. Fessler

EECS Department, BME Department, Dept. of Radiology
University of Michigan

http://web.eecs.umich.edu/~fessler

Wash. Univ. St. Louis ESE Department
2025-10-31

Acknowledgments:
Jason Hu, Bowen Song, Xiaojian Xu, Liyue Shen

arXiv 2406.02462 (NeurIPS 2024)
arXiv 2406.10211 (NeurIPS 2024)
arXiv 2410.11730 (IEEE T-CI, July 2025)

Outline

Under-determined inverse problems

- Applications: compressed sensing MRI, sparse-view CT, PET, inpainting, ...
  All have *linear* forward models for data:

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

  $\boldsymbol{y}$: sensor data (*e.g.*, sinogram)
  $\boldsymbol{A}$: wide system matrix (known)
  $\boldsymbol{x}$: latent image (or image series in dynamic problems)
  ε: noise with known distribution provides likelihood $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$
- Maximum-likelihood estimation (physics-based fitting) is usually non-unique:

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}) = \underbrace{\arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2}_{\text{(for gaussian noise)}}$$

- Minimum-norm least-squares solution is unique but usually impractical or useless:

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^{+}\boldsymbol{y} = \boldsymbol{y} \text{ for inpainting problem}$$

- hand-crafted regularizers:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} -\log \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}) + R(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}} \frac{1}{2\sigma_\varepsilon^2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + R(\boldsymbol{x})$$

- black-box data-driven supervised methods:

$$\boldsymbol{A}^+\boldsymbol{y} \to \boxed{\text{NN}} \to \hat{\boldsymbol{x}}$$

- unrolled deep learning methods (PNP, RED, MoDL, ...)
- Bayesian methods (*e.g.*, MAP) based on a prior $\mathrm{p}(\boldsymbol{x})$, lately (?) relabeled as generative models (or “genAI”)
-

Inverse problem solution methods

- hand-crafted regularizers:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} -\log \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}) + R(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}} \frac{1}{2\sigma_\varepsilon^2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + R(\boldsymbol{x})$$

- black-box data-driven supervised methods:

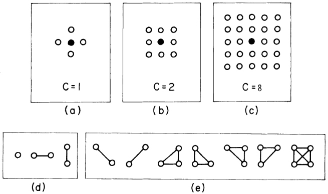
$$\boldsymbol{A}^+ \boldsymbol{y} \to \boxed{\text{NN}} \to \hat{\boldsymbol{x}}$$

- unrolled deep learning methods (PNP, RED, MoDL, ...)
- Bayesian methods (*e.g.*, MAP) based on a prior $\mathrm{p}(\boldsymbol{x})$, lately (?) relabeled as generative models (or "genAI")
- Appeal:
  - PNP-like training independent of $\boldsymbol{A}$ or $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$
  - Strong priors for complex systems with aggressive under-sampling
  - Posterior sampling from $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ for uncertainty quantification

Markov random field models

$$p(\boldsymbol{x}) \propto \prod_{c} \mathrm{e}^{-U_c(\boldsymbol{x}_c)}$$

(*e.g.*) Geman & Geman 1984 [1]



Mostly for inference?

GEMAN AND GEMAN: STOCHASTIC RELAXATION, GIBBS DISTRIBUTIONS, AND BAYESIAN RESTORATION 737

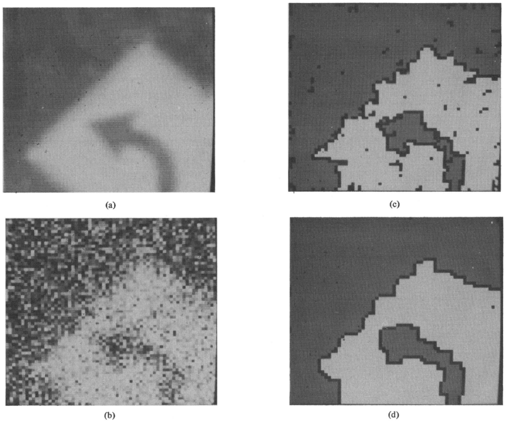


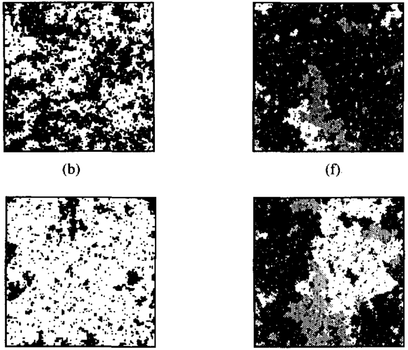
Fig. 7. (a) Blurred image (roadside scene). (b) Degraded image: Additive noise. (c) Restoration including line process; 100 iterations. (d) Restoration including line process; 1000 iterations.

MRF as generators? [2] T-PAMI 1994

An Empirical Study of the Simulation of Various Models Used for Images

A. J. Gray, J. W. Kay, and D. M. Titterington

***Abstract*— Markov random fields are typically used as priors in Bayesian image restoration methods to represent spatial information in the image. Commonly used Markov random fields are not in fact capable of representing the moderate-to-large scale clustering present in naturally occurring images and can also be time consuming to simulate,**



(b) (f) (c) (g)

Local vs global priors

Gray, Kay, Titterington [2] T-PAMI 1994

... the local properties of spatial Markov models are undoubtedly plausible descriptors of the local associations typical of many images, which is the way in which the models are often used. Nevertheless, it would be reassuring if models used as priors did in fact provide a realistic representation of our prior assumptions and if their (empirical) properties were more widely known.

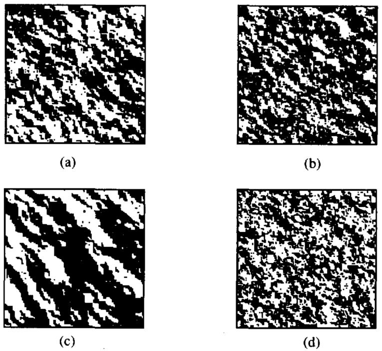


Fig. 4. Realizations of two-dimensional, one-parameter, autologistic Markov Mesh models: (a) binary, second-order model with $\beta = \log 5$; (b) three-color second-order model with $\beta = \log 5$; (c) binary second-order model with $\beta = \log 10$; (d) binary second-order model with $\beta = \log 3$.

Generative models are hot in imaging / inverse problems

Zhao, Ye, Bresler: Jan. 2023 IEEE SpMag survey paper [3]

- Generative adversarial network (GAN) models
- Variation auto-encoder (VAE) models [4]
- Normalizing flows [5, 6]
- Score-based diffusion models
  - Zaccharie Ramzi et al., NeurIPS Workshop 2020 [7]
  - Yang Song & Liyue Shen et al., NeurIPS Workshop 2021, ICLR 2022 [8, 9]
  - Ajil Jalal et al. ... Jon Tamir, NeurIPS 2021 [10]
  - Hyungjin Chung & Jong Chul Ye, MIA, Aug. 2022 [11]
  - Luo et al., MRM, 2023 [12]
  - ...
- Kazerouni et al. [13] have github catalog, including >20 (!) survey papers
- ... (hopelessly incomplete lists)

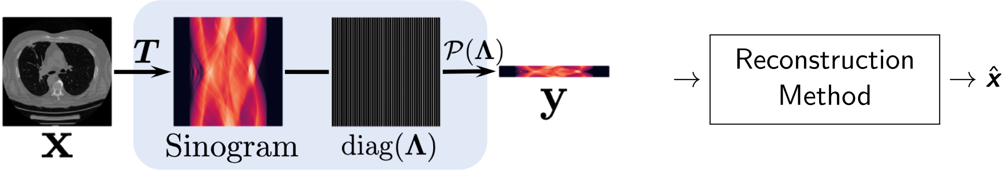
Common aim: model/learn prior $\mathrm{p}(\boldsymbol{x})$

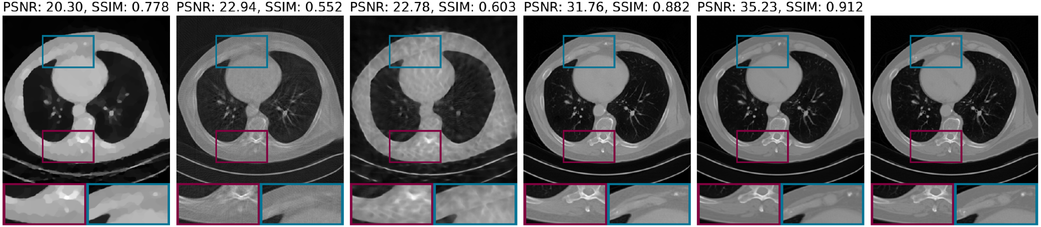
Medical example: Low-dose sparse-view X-ray CT imaging

From Song & Shen et al., ICLR 2022 [9].

Trained with 47K 2D CT images. Recon 23 projection views (≈ 17-fold dose reduction)

$$\boldsymbol{A} = \mathcal{P}(\boldsymbol{\Lambda})\boldsymbol{T}$$





(a) FISTA-TV (b) cGAN (c) Neumann (d) SIN-4c-PRN (e) Ours (f) Ground truth

Challenges with Bayesian models

1. Learning whole-image prior models requires many high-quality training images
   Some applications like dynamic MRI have few *if any* realistic training samples
   - Curse of dimensionality
   - Images live near manifolds (unsuitable for traditional density estimators)
   - Implicit bias of model is crucial
2.

Challenges with Bayesian models

1. Learning whole-image prior models requires many high-quality training images
   Some applications like dynamic MRI have few *if any* realistic training samples
   - Curse of dimensionality
   - Images live near manifolds (unsuitable for traditional density estimators)
   - Implicit bias of model is crucial
2. Existing models scale poorly to 3D or 3D+time
   - GPU memory
   - training data requirements
3.

Challenges with Bayesian models

1. Learning whole-image prior models requires many high-quality training images
   Some applications like dynamic MRI have few *if any* realistic training samples
   - Curse of dimensionality
   - Images live near manifolds (unsuitable for traditional density estimators)
   - Implicit bias of model is crucial
2. Existing models scale poorly to 3D or 3D+time
   - GPU memory
   - training data requirements
3. Training images should arise from relevant distribution $\mathrm{p}(\boldsymbol{x})$
   Imaging-system aspects like X-ray source spectrum may cause domain shift
4.

Challenges with Bayesian models

1. Learning whole-image prior models requires many high-quality training images
   Some applications like dynamic MRI have few *if any* realistic training samples
   - Curse of dimensionality
   - Images live near manifolds (unsuitable for traditional density estimators)
   - Implicit bias of model is crucial
2. Existing models scale poorly to 3D or 3D+time
   - GPU memory
   - training data requirements
3. Training images should arise from relevant distribution $\mathrm{p}(\boldsymbol{x})$
   Imaging-system aspects like X-ray source spectrum may cause domain shift
4. What does “uncertainty” mean if prior is misspecified?

Bayesian methods (generative models)

- Bayesian inference methods use the posterior:

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{y}) = \underbrace{\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})}_{\text{physics}} \; \underbrace{\mathrm{p}(\boldsymbol{x})}_{\text{prior}} \; / \; \mathrm{p}(\boldsymbol{y})$$

- Here the prior $\mathrm{p}(\boldsymbol{x})$ is for quantifying (prior) probability, not necessarily for generation.
- A model for the posterior $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ opens many doors:
  - Maximizing $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ is maximum a posteriori (MAP) estimation
  - The conditional mean $\mathsf{E}[\boldsymbol{x}|\boldsymbol{y}] = \int \boldsymbol{x}\, \mathrm{p}(\boldsymbol{x}|\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}$ is the MMSE estimator
  - Sampling from the posterior $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ facilitates uncertainty quantification in inference
- All these methods require the prior $\mathrm{p}(\boldsymbol{x})$, *i.e.*, a prior model $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$.
-

Bayesian methods (generative models)

- Bayesian inference methods use the posterior:

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{y}) = \underbrace{\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})}_{\text{physics}} \; \underbrace{\mathrm{p}(\boldsymbol{x})}_{\text{prior}} \; / \; \mathrm{p}(\boldsymbol{y})$$

- Here the prior $\mathrm{p}(\boldsymbol{x})$ is for quantifying (prior) probability, not necessarily for generation.
- A model for the posterior $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ opens many doors:
  - Maximizing $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ is maximum a posteriori (MAP) estimation
  - The conditional mean $\mathrm{E}[\boldsymbol{x}|\boldsymbol{y}] = \int \boldsymbol{x}\, \mathrm{p}(\boldsymbol{x}|\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}$ is the MMSE estimator
  - Sampling from the posterior $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ facilitates uncertainty quantification in inference
- All these methods require the prior $\mathrm{p}(\boldsymbol{x})$, *i.e.*, a prior model $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$.
- Or do they?

Sampling via Langevin dynamics

Sampling from a *prior* $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$ just needs its score function $\nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$, using Langevin dynamics, aka stochastic gradient ascent of log-prior:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \alpha_t \underbrace{\nabla \log \mathrm{p}(\boldsymbol{x}_{t-1}; \boldsymbol{\theta})}_{\text{score function}} + \beta_t \underbrace{\boldsymbol{\varepsilon}_t}_{\text{IID } \mathcal{N}(\mathbf{0}, \boldsymbol{I})}, \quad t = 1, \ldots, T.$$

- Draws samples from $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$ for suitable choices of $\{\alpha_t\}$, $\{\beta_t\}$, and (large) T [14].
- If $\alpha_t = 0$ and $\beta_t = \beta$, then akin to (isotropic) diffusion or Brownian motion

Distribution learning vs score learning

- Typical distribution models: $\mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\,\mathrm{e}^{-U(\boldsymbol{x};\boldsymbol{\theta})}$.
  Goal: learn $\boldsymbol{\theta}$ from training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$
- For IID samples $\{\boldsymbol{x}_t\}$, one could try to learn $\boldsymbol{\theta}$ by ML estimation:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T; \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{t=1}^{T} \log(\mathrm{p}(\boldsymbol{x}_t; \boldsymbol{\theta}))$$
$$= \arg\max_{\boldsymbol{\theta}} \left(-T\,Z(\boldsymbol{\theta}) + \sum_{t=1}^{T} -U(\boldsymbol{x}_t; \boldsymbol{\theta}) \right).$$

  Typically intractable due to the partition function $Z(\boldsymbol{\theta})$.
-

Distribution learning vs score learning

- Typical distribution models: $\mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\,\mathrm{e}^{-U(\boldsymbol{x};\boldsymbol{\theta})}$.
  Goal: learn $\boldsymbol{\theta}$ from training data $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T$
- For IID samples $\{\boldsymbol{x}_t\}$, one could try to learn $\boldsymbol{\theta}$ by ML estimation:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T;\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{t=1}^{T} \log(\mathrm{p}(\boldsymbol{x}_t;\boldsymbol{\theta}))$$
$$= \arg\max_{\boldsymbol{\theta}} \left(-T\,Z(\boldsymbol{\theta}) + \sum_{t=1}^{T} -U(\boldsymbol{x}_t;\boldsymbol{\theta})\right).$$

  Typically intractable due to the partition function $Z(\boldsymbol{\theta})$.
- In contrast, the score function is easier to handle:

$$\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) \triangleq \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{x}}\left(-\log Z(\boldsymbol{\theta}) - U(\boldsymbol{x};\theta)\right) = -\nabla_{\boldsymbol{x}} U(\boldsymbol{x};\theta).$$

Score matching

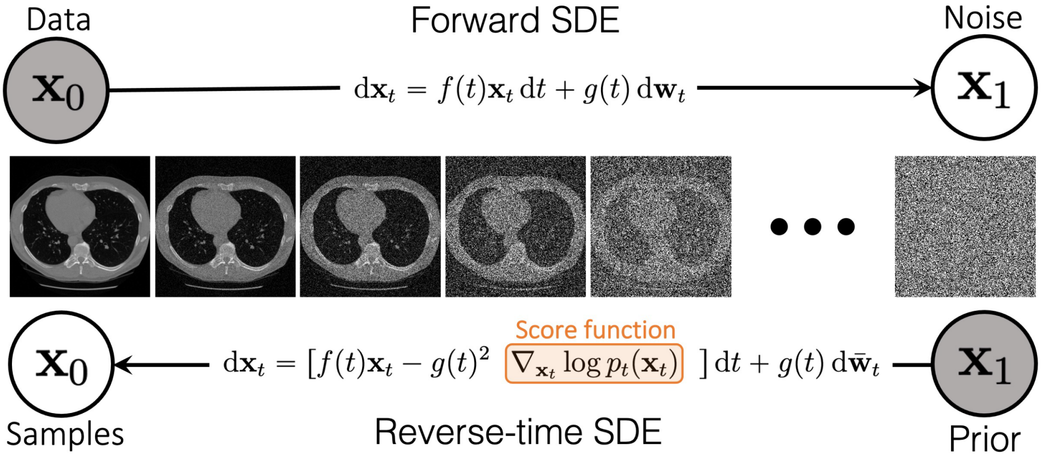
- Given training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, learn score function $\boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}) \stackrel{?}{=} \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$
-

- Given training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, learn score function $\boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}) \stackrel{?}{=} \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$
- Explicit score matching (ESM) (Hyvärinen, 2005 [15])
- Implicit score matching (ISM)
- Denoising score matching (DSM) (Vincent, 2011 [16])
- Noise-conditional score matching (NCSM) (Song, 2019 [17, eqn. (5)]):

$$\ell(\boldsymbol{\theta}; \sigma) \triangleq \frac{1}{2} \mathrm{E}_{\mathrm{q}_0(\boldsymbol{x})}\left[\mathrm{E}_{g_\sigma(\boldsymbol{z})}\left[\left\|\boldsymbol{s}(\boldsymbol{x}+\boldsymbol{z}; \boldsymbol{\theta}, \sigma) + \frac{\boldsymbol{z}}{\sigma^2}\right\|_2^2\right]\right], \quad \mathcal{L}(\boldsymbol{\theta}; \{\sigma_l\}) = \frac{1}{L}\sum_{l=1}^{L} \sigma_l^2\, \ell(\boldsymbol{\theta}; \sigma_l),$$

  where $\boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}, \sigma)$ denotes a *noise-conditional score network* (NCSN).
- $\boldsymbol{d}(\boldsymbol{x}; \boldsymbol{\theta}) \triangleq \boldsymbol{x} + \sigma^2 \boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}, \sigma)$: equivalent image denoiser by Tweedie's formula [18]
- Recommended choice [19]: $\boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}, \sigma) \triangleq \tilde{\boldsymbol{s}}(\boldsymbol{x}; \boldsymbol{\theta})/\sigma$, where $\tilde{\boldsymbol{s}}$ is unitless

Shen & Song et al., NeurIPS 2021 [8]



Score-based diffusion models: trade-offs

- No adversarial training needed
- High quality sample generation (if enough training data)
-

Score-based diffusion models: trade-offs

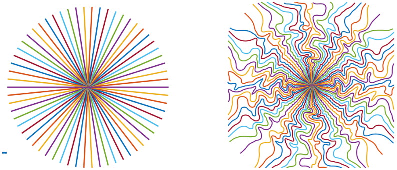
- No adversarial training needed
- High quality sample generation (if enough training data)
- Expensive sample generation (vs GAN models)
  - Distillation methods [20]
  - Consistency models [21]
  - Geometric decomposition [22]
  - Multi-scale [23, 24] and pyramidal [25] and coarse-to-fine [26] models
  - Faster ODE solvers [27]
  - Warm starts [28]
  - Latent diffusion models: use VAE and diffuse in latent space [29–31]. Used in Stable Diffusion by start-up Stability AI
  - 3D image reconstruction using 2D models [32, 33]
- Learning 3D (or 3D+T) whole-image generative models is challenging (training data, GPU memory, ...)

Challenge 1: Data availability

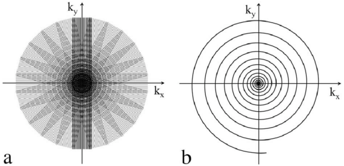
Whole images vs patches?

Jan. 2023 survey paper on generative models [3] does not mention “patch” once!?

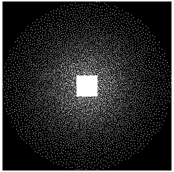
MRI k-space sampling:



[34]



[35]



[36]

Patch-based models have long history in inverse problems, *e.g.*,

- patch GAN [37–39]
- patch dictionary models [40, 41]
- non-local means, BM3D
- Wasserstein patch prior [42, 43] . . .

Motivating questions

- Can patch-based generative models be effective priors for inverse problems in applications with very limited training data?
  e.g., dynamic MRI

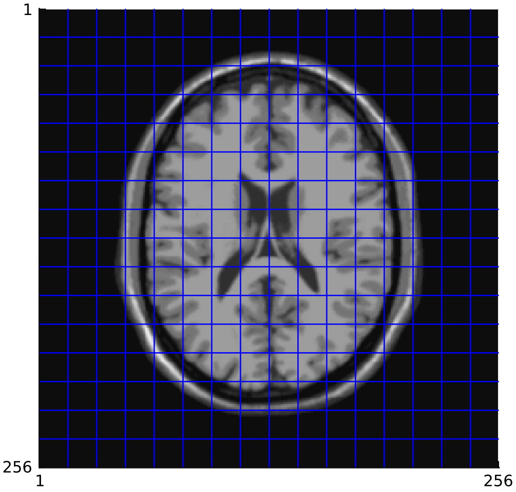
- Can patch-based generative models provide better robustness to distribution shifts, perhaps at the cost of reduced in-distribution performance?

- Can we use the "latest" generative models, *e.g.*, score-based models, for patches?

Warm up:
simple, but less effective, approach:

- Fixed patch size
- Fixed patch grid
- No position information

(Fessler, Hu, Xu, BASP 2023 [46])



Patch-based score modeling

- Start with MRF formulation, aka *fields of experts* model [51–53] for image $\boldsymbol{x}$:

$$\mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \mathrm{e}^{-\sum_c V_c(\boldsymbol{x};\boldsymbol{\theta})} = \frac{1}{Z(\boldsymbol{\theta})} \prod_c \mathrm{e}^{-V_c(\boldsymbol{x};\boldsymbol{\theta})}.$$

  - $\boldsymbol{\theta}$: parameter vector that describes the prior
  - V_c : *clique potential* for the cth image *patch*
  - $Z(\boldsymbol{\theta})$: (intractable) partition function
- Assume (temporarily) statistical spatial stationarity (image shift invariance):

$$V_c(\boldsymbol{x};\boldsymbol{\theta}) = V(\boldsymbol{G}_c\boldsymbol{x};\boldsymbol{\theta})$$

  - $\boldsymbol{G}_c$: wide binary matrix that grabs pixels of the cth patch from image $\boldsymbol{x}$
  - $V(\boldsymbol{v};\boldsymbol{\theta})$: common patch clique function

Patch-based score modeling (simple)

- Resulting log-prior:

$$\log \mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}) = -\log Z(\boldsymbol{\theta}) - \sum_c V(\boldsymbol{G}_c \boldsymbol{x};\boldsymbol{\theta})$$

- Corresponding overall *image score function* arises from *patch score function*:

$$\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) \triangleq \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}) = \sum_c \boldsymbol{G}_c' \boldsymbol{s}_V(\boldsymbol{G}_c \boldsymbol{x};\boldsymbol{\theta}), \qquad \boldsymbol{s}_V(\boldsymbol{v};\boldsymbol{\theta}) \triangleq -\nabla_{\boldsymbol{v}} V(\boldsymbol{v};\boldsymbol{\theta}).$$

- All we must learn is the patch score function $\boldsymbol{s}_V(\boldsymbol{v};\boldsymbol{\theta}) : \mathbb{R}^n \mapsto \mathbb{R}^n$, *e.g.*, a UNet.
- For non-overlapping patches:

$$\underbrace{\left\| \boldsymbol{s}(\boldsymbol{x}+\boldsymbol{z};\boldsymbol{\theta}) + \boldsymbol{z}/\sigma^2 \right\|_2^2}_{\text{image “denoise”}} = \left\| \sum_c \boldsymbol{G}_c' \boldsymbol{s}_V(\boldsymbol{G}_c(\boldsymbol{x}+\boldsymbol{z});\boldsymbol{\theta}) + \boldsymbol{z}/\sigma^2 \right\|_2^2$$

$$= \sum_c \underbrace{\left\| \boldsymbol{s}_V(\boldsymbol{x}_c+\boldsymbol{z}_c);\boldsymbol{\theta}) + \boldsymbol{z}_c/\sigma^2 \right\|_2^2}_{\text{patch “denoise”}}, \quad \boldsymbol{z}_c \triangleq \boldsymbol{G}_c \boldsymbol{z}$$

Patch-based score learning (simple)

- For training image patches $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_T\}$, apply *denoising score matching* (DSM) of Vincent, 2011 [16], typically for a range of noise variances σ^2 [14]:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} \mathsf{E}_{\sigma \sim p(\sigma)} \left[\sigma^2 \, \mathsf{E}_{\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_n)} \left[\frac{1}{2} \left\| \boldsymbol{s}_V(\boldsymbol{v}_t + \boldsymbol{z}; \boldsymbol{\theta}, \sigma) + \frac{\boldsymbol{z}}{\sigma^2} \right\|_2^2 \right] \right].$$

- Final patch score model is $\boldsymbol{s}_V(\boldsymbol{v}; \hat{\boldsymbol{\theta}}, \sigma_{\min})$.
-

- For training image patches $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_T\}$, apply *denoising score matching* (DSM) of Vincent, 2011 [16], typically for a range of noise variances σ^2 [14]:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\arg\min} \frac{1}{T} \sum_{t=1}^{T} \mathrm{E}_{\sigma \sim p(\sigma)} \left[\sigma^2 \, \mathrm{E}_{\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_n)} \left[\frac{1}{2} \left\| \boldsymbol{s}_V(\boldsymbol{v}_t + \boldsymbol{z}; \boldsymbol{\theta}, \sigma) + \frac{\boldsymbol{z}}{\sigma^2} \right\|_2^2 \right] \right].$$

- Final patch score model is $\boldsymbol{s}_V(\boldsymbol{v}; \hat{\boldsymbol{\theta}}, \sigma_{\text{min}})$.
- Network input is just image patches, never the entire image
  $\Longrightarrow$ scales to large 2D images, 3D, 4D, etc.
-

Patch-based score learning (simple)

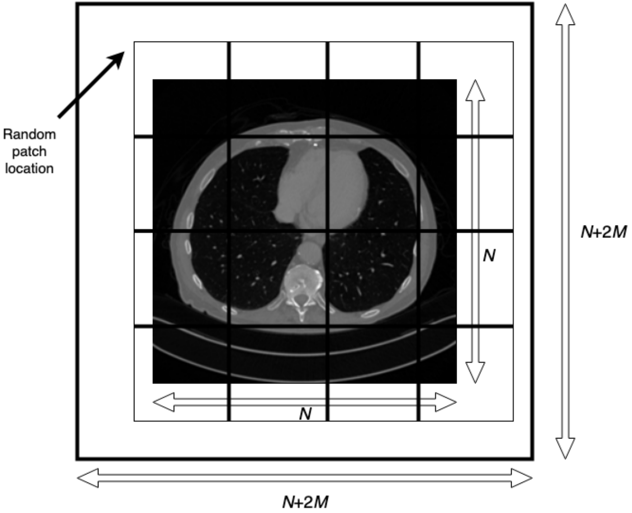
- For training image patches $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_T\}$, apply *denoising score matching* (DSM) of Vincent, 2011 [16], typically for a range of noise variances σ^2 [14]:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\arg\min} \frac{1}{T} \sum_{t=1}^{T} \mathsf{E}_{\sigma \sim p(\sigma)} \left[\sigma^2 \, \mathsf{E}_{\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_n)} \left[\frac{1}{2} \left\| \boldsymbol{s}_V(\boldsymbol{v}_t + \boldsymbol{z}; \boldsymbol{\theta}, \sigma) + \frac{\boldsymbol{z}}{\sigma^2} \right\|_2^2 \right] \right].$$

- Final patch score model is $\boldsymbol{s}_V(\boldsymbol{v}; \hat{\boldsymbol{\theta}}, \sigma_{\min})$.
- Network input is just image patches, never the entire image
  $\Longrightarrow$ scales to large 2D images, 3D, 4D, etc.
- Drawbacks:
  - Visible patch boundaries
  - Fixed patch size slows learning
  - Suboptimal stationarity assumption (*cf.* vertebrae)

Improved patch modeling

- zero-pad image $\boldsymbol{x}$
- use multiple grid locations



Inspirations:

- Wavelet "cycle spinning" [47, 54–57]
- Wang, NeurIPS 2023 [58]

Probability model with padding & grids & positions

- $N_1 \times N_2$: original image size
- $P_1 \times P_2$: patch size
- $K_i \triangleq 1 + \lfloor N_i / P_i \rfloor$, $i = 1, 2$: # non-overlapping patches for original image
- $(N_1 + 2M_1) \times (N_2 + 2M_2)$: padded image size; $M_i \triangleq K_i P_i - N_i$
- Product probability model:

$$\mathrm{p}(\boldsymbol{x}) \triangleq \frac{1}{Z} \underbrace{\prod_{m=1}^{M_1 M_2}}_{\substack{\text{grid}\\\text{shifts}}} \left(\underbrace{\mathrm{p}_{m,B}(\boldsymbol{x}_{m,B})}_{\substack{\text{border}\\\text{region}}} \underbrace{\prod_{k=1}^{K_1 K_2} \mathrm{p}_{m,k}(\boldsymbol{x}_{m,k})}_{\text{patches}} \right) = \frac{1}{Z} \prod_{m=1}^{M_1 M_2} \prod_{k=1}^{K_1 K_2} \underbrace{\mathrm{e}^{-V(\boldsymbol{x}_{m,k}; m, k)}}_{\substack{\text{position}\\\text{encoding}}}$$

  - $\boldsymbol{x}_{m,B}$: border pixels for mth shift (all zero)
  - $\boldsymbol{x}_{m,k}$: kth patch for mth shift

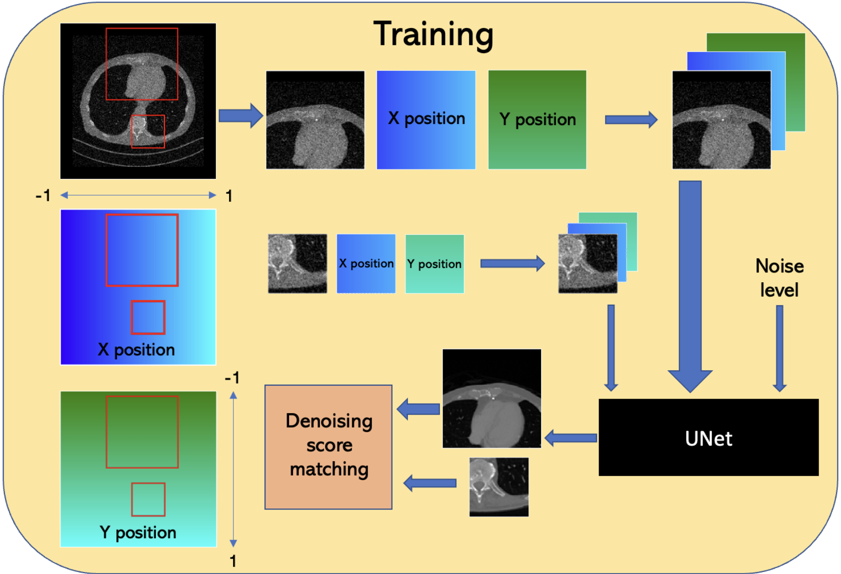
- $N_1 \times N_2$: original image size
- $P_1 \times P_2$: patch size
- $K_i \triangleq 1 + \lfloor N_i / P_i \rfloor,\ i = 1, 2$: # non-overlapping patches for original image
- $(N_1 + 2M_1) \times (N_2 + 2M_2)$: padded image size; $M_i \triangleq K_i P_i - N_i$
- Product probability model:

$$\mathrm{p}(\boldsymbol{x}) \triangleq \frac{1}{Z} \underbrace{\prod_{m=1}^{M_1 M_2}}_{\text{grid shifts}} \left(\underbrace{\mathrm{p}_{m,B}(\boldsymbol{x}_{m,B})}_{\text{border region}} \underbrace{\prod_{k=1}^{K_1 K_2} \mathrm{p}_{m,k}(\boldsymbol{x}_{m,k})}_{\text{patches}} \right) = \frac{1}{Z} \prod_{m=1}^{M_1 M_2} \prod_{k=1}^{K_1 K_2} \underbrace{\mathrm{e}^{-V(\boldsymbol{x}_{m,k}; m, k)}}_{\text{position encoding}}$$

  - $\boldsymbol{x}_{m,B}$: border pixels for mth shift (all zero)
  - $\boldsymbol{x}_{m,k}$: kth patch for mth shift
- Learn position-dependent patch score function $\boldsymbol{s}(\boldsymbol{v}; \boldsymbol{\theta}, m, k) = -\nabla_{\boldsymbol{v}} V(\boldsymbol{v}; m, k)$

Patch Diffusion Inverse Solver (PaDIS): Training

NeurIPS 2024 [60]
arXiv 2406.02462



Training images (CT)

AAPM 2016 CT challenge data [61];
10 3D volumes,
rescaled to 256^3

Example slices:

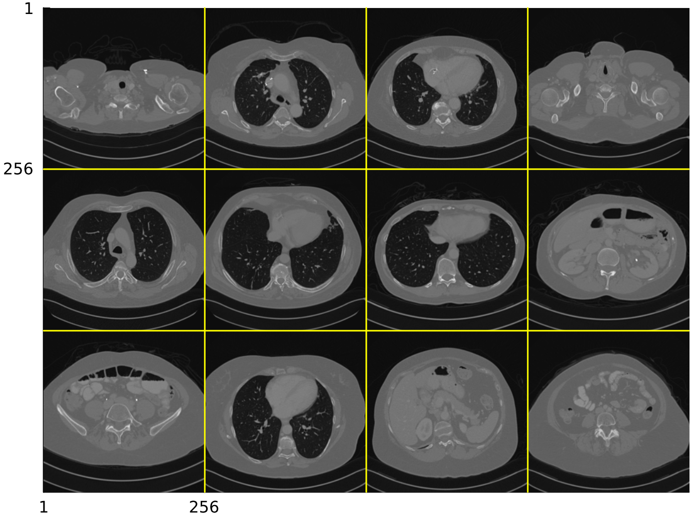
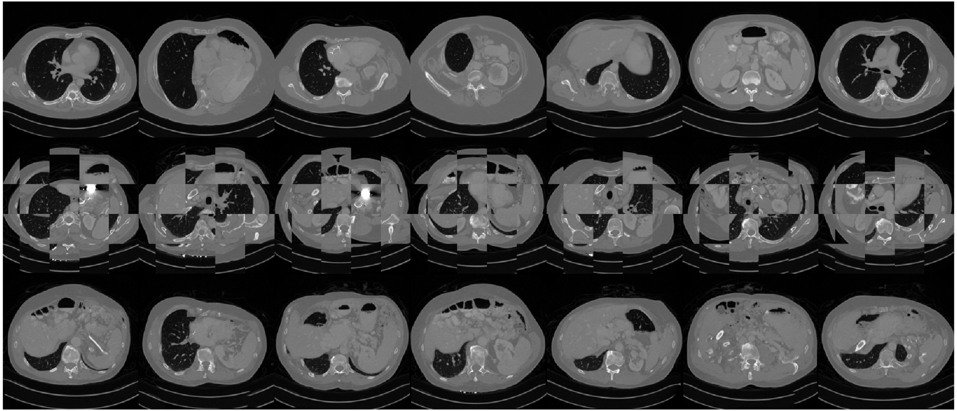


Image generation (unconditional sampling from prior)

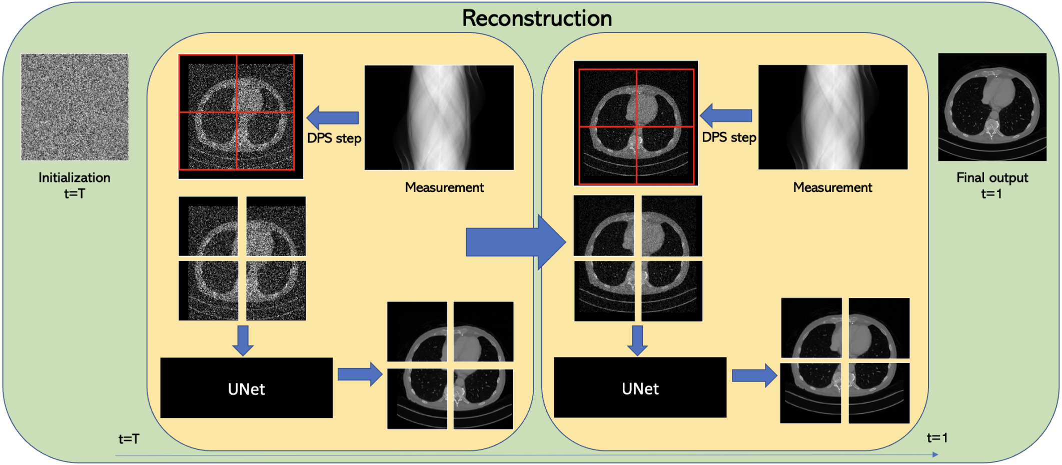


- Top: generation with a network trained on whole images (2D...)
- Middle: patch-only version of [58] (non-overlapping patches).
- Bottom: generation with proposed PaDIS prior.

Training time results

2 A40 GPUs using PyTorch and ADAM

- whole image model: 24 – 36 hours
- patch-based model: $\approx$ 12 hours



Diffusion posterior sampling (DPS) (Chung et al., ICLR 2023 [62]) with Langevin dynamics, modified to use patch score with random grid shifts.

PaDIS algorithm (modified from DPS)

Input: $\boldsymbol{y}$, $\boldsymbol{A}$, T, $\sigma_1 < \sigma_2 < \ldots < \sigma_T$, $\epsilon > 0$, $\{\zeta_t > 0\}$, P_1, P_2, M_1, M_2, trained noise-conditional, position-encoded patch denoiser $\boldsymbol{d}(\cdot; \boldsymbol{\theta}_*, m, k, \sigma)$

Initialize random image $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \sigma_T^2 \boldsymbol{I})$

for $t = T : 1$ **do**

Randomly select grid integer $m \in \{1, \ldots, M_1 M_2\}$

for $k = 1 : (K_1 K_2)$ **do** (parallelizable)

Extract patch $\boldsymbol{x}_{m,k}$

Denoise patch: $\boldsymbol{d}_{m,k} \triangleq \boldsymbol{d}(\boldsymbol{x}_{m,k}; \boldsymbol{\theta}_*, m, k, \sigma_t)$

end for

Combine denoised patches to get denoised image $\boldsymbol{d}$

Compute image score function: $\boldsymbol{s} = (\boldsymbol{d} - \boldsymbol{x}) / \sigma_t^2$

Data term: $\boldsymbol{x} := \boldsymbol{x} - \zeta_t \nabla_{\boldsymbol{x}} \| \boldsymbol{A}\, \boldsymbol{d}(\boldsymbol{x}) - \boldsymbol{y} \|_2^2$

Sample $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \sigma_t^2 \boldsymbol{I})$

Step size $\alpha_t \triangleq \epsilon\, \sigma_t^2$

Langevin update: $\boldsymbol{x} := \boldsymbol{x} + \frac{\alpha_t}{2} \boldsymbol{s} + \sqrt{\alpha_t} \boldsymbol{z}$

end for

CT Experiments

Default setup:

- 9 of 10 volumes for training $\Longrightarrow$ 2304 slices
- 25 slices of 10th volume for testing
- 512 element parallel-beam CT detector
- $\boldsymbol{A}$ from Operator Discretization Library (ODL)
- 56×56 patch size
- U-Net of Karras 2022 [59]
- Step size $\zeta_t = \zeta / \|\boldsymbol{A}\boldsymbol{d}(\boldsymbol{x}_t) - \boldsymbol{y}\|_2$
- 1000 neural function evaluations (NFEs) [59]

Quantitative results on three different inverse problems

Method	CT, 20 Views		CT, 8 Views		Deblurring		Superresolution	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	24.93	0.595	21.39	0.415	24.54	0.688	25.86	0.739
ADMM-TV	26.82	0.724	23.09	0.555	28.22	0.792	25.66	0.745
PnP-ADMM [63]	26.86	0.607	22.39	0.489	28.82	0.818	26.61	0.785
PnP-RED [64]	27.99	0.622	23.08	0.441	29.91	0.867	26.36	0.766
Whole image diffusion	32.84	0.835	25.74	0.706	30.19	0.853	29.17	0.827
Langevin dynamics [17]	33.03	0.846	27.03	0.689	30.60	0.867	26.83	0.744
Predictor-corrector [11]	32.35	0.820	23.65	0.546	28.42	0.724	26.97	0.685
VE-DDNM [65]	31.98	0.861	27.71	0.759	-	-	26.01	0.727
Patch Averaging [50]	33.35	0.850	28.43	0.765	29.41	0.847	27.67	0.802
Patch Stitching	32.87	0.837	26.71	0.710	29.69	0.849	27.50	0.780
PaDIS (Ours)	**33.57**	**0.854**	**29.48**	**0.767**	**30.80**	**0.870**	**29.47**	**0.846**

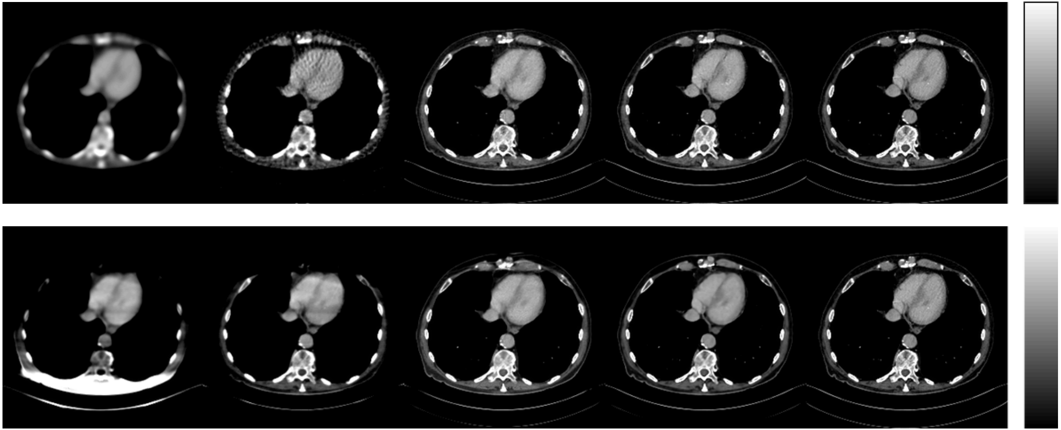
(Averages across all test images.)

More inverse problem experiments

Method	CT, 60 Views		CT, Fan Beam		Heavy Deblurring	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	25.89	0.746	20.07	0.521	21.14	0.569
ADMM-TV	30.93	0.833	25.78	0.719	26.03	0.724
Whole image diffusion	35.83	0.894	26.89	0.835	28.35	0.808
PaDIS (Ours)	**39.28**	**0.941**	**29.91**	**0.932**	**28.91**	**0.818**

Example images

baseline FBP	ADMM-TV	whole image diffusion	PaDIS	ground truth



Top: 60 view CT
Bottom: fan-beam CT

$\approx$ 400 HU window width

Patchsize

P	PSNR↑	SSIM↑
8	32.57	0.844
16	32.57	0.829
32	32.72	0.853
56	**33.57**	**0.854**
96	33.36	**0.854**
256	32.84	0.835

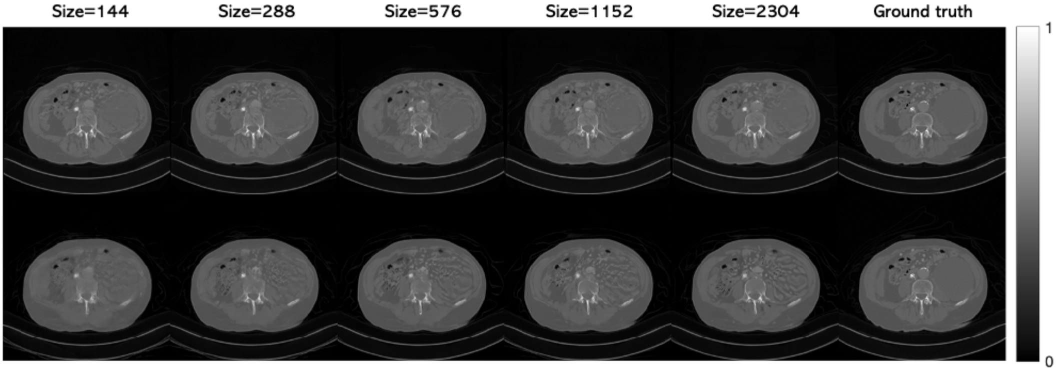
Positional encoding

	PSNR↑	SSIM↑
no position enc.	23.25	0.459
no position+init	24.51	0.518
with position enc.	**33.57**	**0.854**

Effect of training dataset size on CT reconstruction

Dataset size	Patches 56×56		Whole image 256×256	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
144	32.28	0.841	29.12	0.804
288	32.43	0.837	31.09	0.829
576	33.03	0.846	31.81	0.835
1152	33.01	0.849	31.36	0.834
2304	**33.57**	0.854	32.84	0.835

20 view CT reconstruction: training dataset sizes

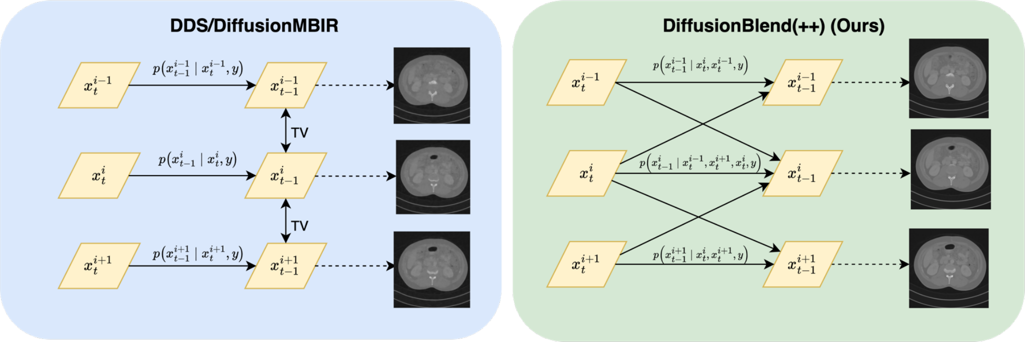


Top : PaDIS
Bottom : whole image diffusion model

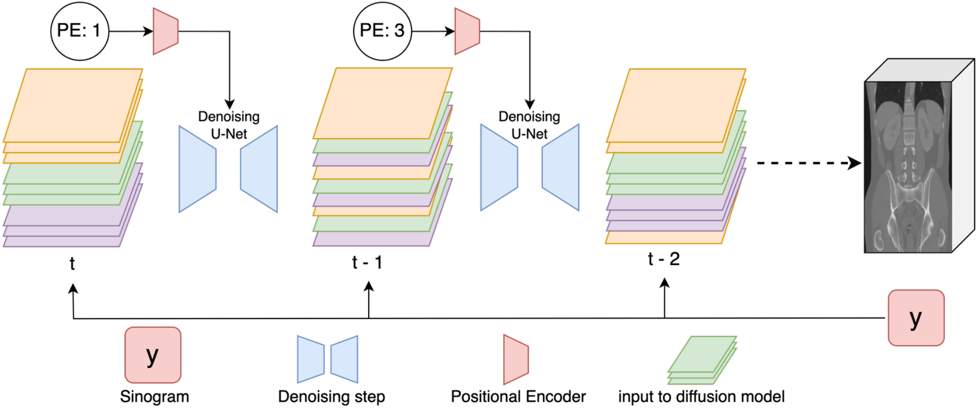
Challenge 2: Data dimensions & scaling to 3D (and 4D)

3D “patches” with DiffusionBlend

arXiv 2406.10211 (NeurIPS 2024) [66]



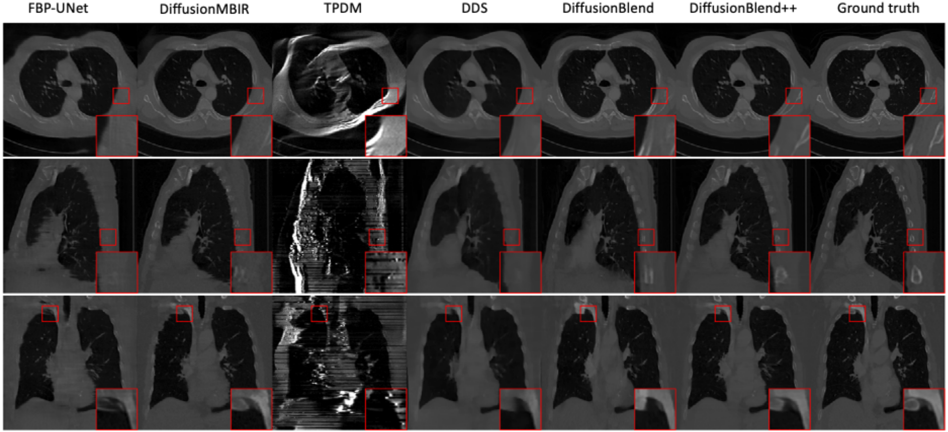
DiffusionBlend models groups of slices



Method	Distribution Model
DiffusionMBIR (2D) [32]	$\frac{1}{Z}\prod_{i=1}^{H} p(\boldsymbol{x}[:,:,i])$
TPDM ($\perp$ 2D) [33]	$\frac{1}{Z}\left(\prod_{i=1}^{N} q_\theta(\boldsymbol{x}[:,:,i])^\alpha\right)\left(\prod_{j=1}^{N} q_\phi(\boldsymbol{x}[j,:,:])^\beta\right)$
DiffusionBlend	$\frac{1}{Z}\prod_{i=1}^{H} p(\boldsymbol{x}[:,:,i] \mid \boldsymbol{x}[:,:,i-j:i-1], \boldsymbol{x}[:,:,i+1:i+j])$
DiffusionBlend++	$\frac{1}{Z}\prod_{i=1}^{r} p(\boldsymbol{x}[:,:,\mathcal{S}_i])$

DiffusionBlend for 3D limited-angle CT

90° angular range



Improved quality both qualitatively and quantitively with strong prior.

3D limited-angle CT results

Method	AAPM Dataset						LIDC Dataset					
	Axial		Sagittal		Coronal		Axial		Sagittal		Coronal	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
FBP	16.36	0.643	16.36	0.524	15.62	0.531	18.79	0.672	19.84	0.675	20.01	0.676
FBP-UNet	27.38	0.910	27.81	0.918	28.44	0.930	29.42	0.885	29.50	0.884	29.54	0.887
DiffusionMBIR	25.98	0.872	27.14	0.877	27.74	0.903	<u>30.52</u>	0.906	30.57	0.906	30.68	0.907
TPDM	-	-	-	-	-	-	14.44	0.141	14.06	0.141	14.54	0.313
DDS 2D	28.05	0.916	27.99	0.916	28.82	0.922	27.92	0.843	27.89	0.835	27.96	0.842
DDS	28.20	0.918	28.17	0.926	29.03	0.934	28.12	0.865	28.06	0.869	28.13	0.879
DiffusionBlend (Ours)	<u>35.38</u>	<u>0.971</u>	<u>35.85</u>	<u>0.972</u>	**37.62**	<u>0.972</u>	30.43	<u>0.917</u>	<u>31.24</u>	<u>0.920</u>	<u>31.02</u>	<u>0.924</u>
DiffusionBlend++ (Ours)	**35.86**	**0.975**	**36.03**	**0.976**	<u>37.45</u>	**0.976**	**34.33**	**0.957**	**34.48**	**0.957**	**34.64**	**0.956**

[66, Table 4]

Challenge 3: Distribution shifts

Test-time latent $\boldsymbol{x}$ far from training distribution:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \varepsilon, \quad \boldsymbol{x} \sim \tilde{\mathrm{p}}(\cdot) \neq \mathrm{p}(\cdot)$$

- Non-Bayes approach
  Abandon training via self-supervision, *e.g.*, deep image prior (DIP) [67]:

$$\hat{\boldsymbol{x}} = f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{z}), \qquad \hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A} f_{\boldsymbol{\theta}}(\boldsymbol{z})\|_2^2, \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

  Neural network $f_{\boldsymbol{\theta}}(\cdot)$ acts as implicit regularizer.
  DIP is prone to overfitting of noisy measurements [67];
  remedies such as early stopping, regularization, network initialization [68–70].

- Self-supervised (whole-image) diffusion models [71, 72]
  "Deep diffusion image prior" (DDIP) or "steerable conditional diffusion:"

$$L(\boldsymbol{\theta}) = \|\boldsymbol{y} - \boldsymbol{A}\,\mathrm{CG}(\hat{\boldsymbol{x}}_{0|t}(\boldsymbol{x}_t; \boldsymbol{\theta}))\|_2^2$$
$$\mathrm{CG}(\hat{\boldsymbol{x}}_{0|t}) \triangleq \arg\min_{\boldsymbol{x}} \frac{\gamma}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \frac{1}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{0|t}\|_2^2$$

  Conjugate gradient (CG) descent is used to enforce data fidelity.
  Still requires early stopping to avoid over-fitting.

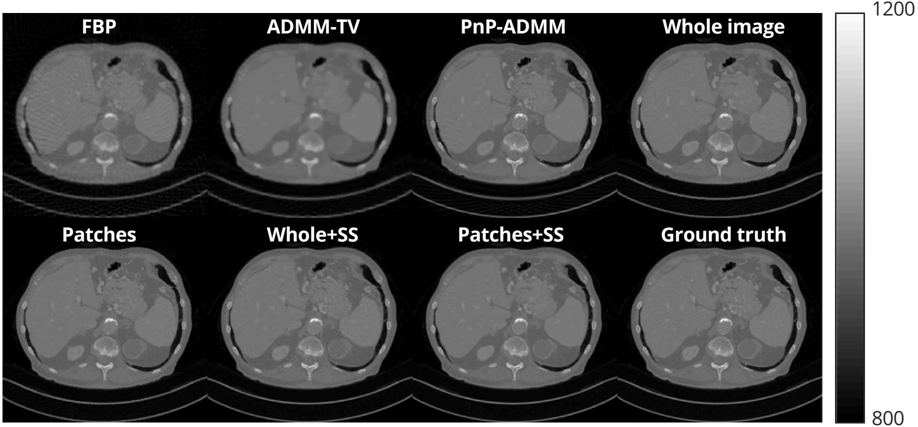
Patch-based test-time adaptation I

- Patch-based test-time adaptation [73, 74] arXiv 2410.11730 (IEEE T-CI, in-press)
  Test-time loss for diffusion model adaptation:

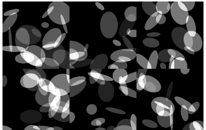
$$L(\boldsymbol{\theta}) = \left\| \boldsymbol{y} - \boldsymbol{A} \sum\nolimits_c \boldsymbol{G}_c' D_{\boldsymbol{\theta}}(\boldsymbol{G}_c \boldsymbol{x}_t, c | \boldsymbol{y}) \right\|_2^2$$

  Patch-based denoiser for diffusion model

$$D_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_c \boldsymbol{G}_c' D_{\boldsymbol{\theta}}(\boldsymbol{G}_c \boldsymbol{x}, c),$$



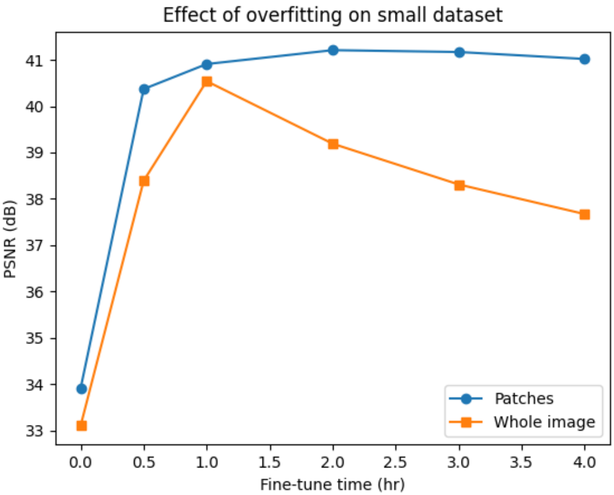
No in-distribution training data. Pre-trained with random ellipses.
Results of 60-view CT reconstruction using self supervised (SS) loss.



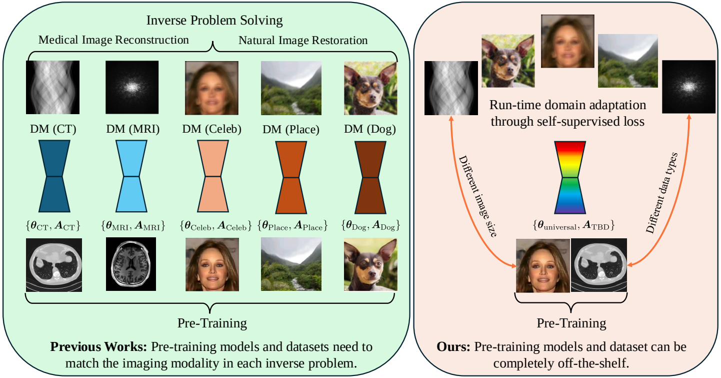
Method	CT, 20 Views		CT, 60 Views		Deblurring		Superresolution	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	24.93	0.613	30.15	0.784	23.93	0.666	25.42	0.724
ADMM-TV	26.81	0.750	31.14	0.862	27.58	0.773	25.22	0.729
PnP-ADMM [63]	30.20	0.838	36.75	0.932	28.98	0.815	27.29	0.796
PnP-RED [64]	27.12	0.682	32.68	0.876	28.37	0.793	27.73	0.809
Whole image	28.11	0.800	33.10	0.911	25.85	0.742	25.65	0.742
Patches [60]	27.44	0.719	33.97	0.934	26.77	0.782	26.12	0.759
Whole+SS [72]	33.19	0.861	40.47	0.957	29.50	0.831	27.07	0.701
Patches+SS (Ours)	**33.77**	**0.874**	**41.45**	**0.969**	**30.34**	**0.860**	**28.10**	**0.827**

"SS" = self-supervision, aka test-time adaptation

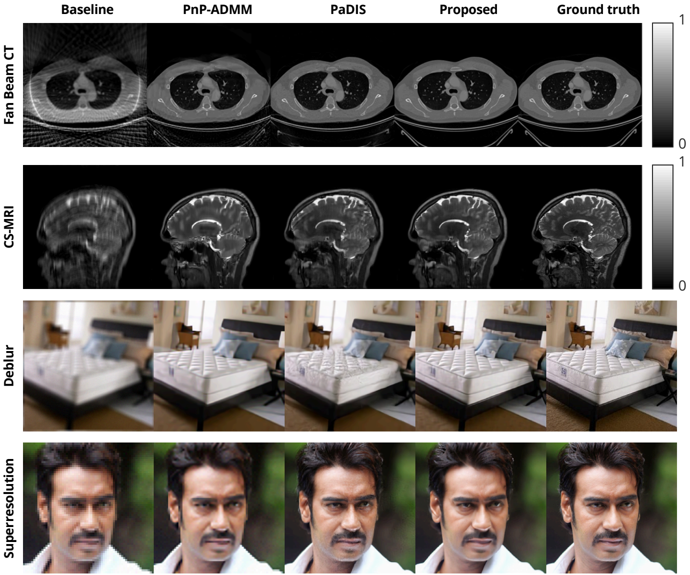
Patch-based test-time adaptation IV



- Extension to cases where # of channels at test time differs from training data, *e.g.*, MR reconstruction (real/imag) from patch-based diffusion model pre-trained on color (RGB) natural images and grayscale CT images [75]



SPAR results



Comparison of quantitative results on four different medical imaging inverse problems.

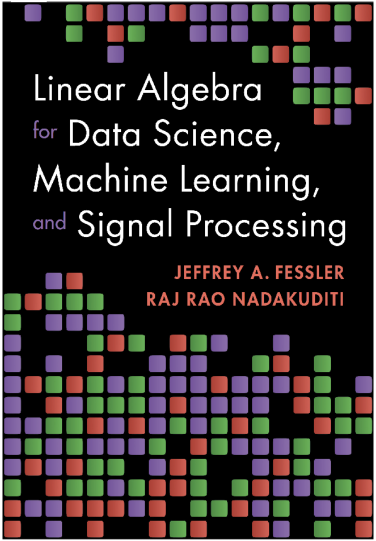
Method	PBCT, 60 Views		FBCT, 40 Views		512 × 512 CT		CS-MRI, 7×	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	30.15	0.784	17.86	0.381	28.33	0.700	33.94	0.894
ADMM-TV	31.14	0.862	24.20	0.628	29.36	0.788	36.74	0.924
PnP-ADMM [63]	36.75	0.932	28.86	0.747	37.48	0.910	35.77	0.907
PaDIS+FC [60]	39.16	0.942	27.91	0.796	33.11	0.831	35.17	0.904
SCD [72]	41.16	0.962	21.28	0.463	–	–	–	–
Ours (SPAR)	**42.72**	**0.972**	**36.11**	**0.918**	**38.81**	**0.929**	**39.15**	**0.949**
Ideal*	42.82	0.973	36.34	0.923	38.94	0.930	39.42	0.953

*not available in practice with a single diffusion model

Summary / future directions

- Challenges
  - Dearth of data
  - Dimensionality
  - Distribution shifts
- Promise
  - Generative models are promising for under-determined inverse problems
  - Learning patch score models is feasible with denoising score matching
  - For limited training data, patch-models can outperform whole-image models
- Future steps
  - Integrate invariances: amplitude scale / rotation / flip / DC offset ...
  - Explore trade-offs between generalizability and in-distribution performance
  - Extend to 3D, 3D+Time, 3D+Multicontrast

Tutorial Julia code: `https://github.com/JeffFessler/ScoreMatching.jl`



- Online demos:
  `https://github.com/JeffFessler/book-la-demo`
- Topics include: low-rank matrix approximation, robust PCA, photometric stereo, video foreground/background separation, spectral clustering, matrix completion, ...
- Cambridge Univ. Press, 2024

Talk and code available online at
`http://web.eecs.umich.edu/~fessler`

[1] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images." In: *IEEE Trans. Patt. Anal. Mach. Int.* 6.6 (Nov. 1984), 721–41.

[2] A. J. Gray, J. W. Kay, and D. M. Titterington. "An empirical study of the simulation of various models used for images." In: *IEEE Trans. Patt. Anal. Mach. Int.* 16.5 (May 1994), 507–12.

[3] Z. Zhao, J. C. Ye, and Y. Bresler. "Generative models for inverse imaging problems: from mathematical foundations to physics-driven applications." In: *IEEE Sig. Proc. Mag.* 40.1 (Jan. 2023), 148–63.

[4] E. D. Zhong, T. Bepler, B. Berger, and J. H. Davis. "CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks." In: *Nature Meth.* 18.2 (2021), 176–85.

[5] D. Rezende and S. Mohamed. "Variational inference with normalizing flows." In: *Proc. Intl. Conf. Mach. Learn.* 2015, 1530–8.

[6] F. Altekruger, A. Denker, P. Hagemann, J. Hertrich, P. Maass, and G. Steidl. "PatchNR: learning from very few images by patch normalizing flow regularization." In: *Inverse Prob.* 39.6 (May 2023), p. 064006.

[7] Z. Ramzi, B. Remy, F. Lanusse, J-L. Starck, and P. Ciuciu. "Denoising score-matching for uncertainty quantification in inverse problems." In: *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*. 2020.

[8] Y. Song, L. Shen, L. Xing, and S. Ermon. "Solving inverse problems in medical imaging with score-based generative models." In: *NeurIPS Deep Inv. Work.* 2021.

[9] Y. Song, L. Shen, L. Xing, and S. Ermon. "Solving inverse problems in medical imaging with score-based generative models." In: *Proc. Intl. Conf. on Learning Representations*. 2022.

[10] A. Jalal, M. Arvinte, G. Daras, E. Price, A. Dimakis, and J. Tamir. "Robust compressed sensing MR imaging with deep generative priors." In: *NeurIPS Workshop Deep Inverse*. 2021.

[11] H. Chung and J. C. Ye. "Score-based diffusion models for accelerated MRI." In: *Med. Im. Anal.* 80 (Aug. 2022), p. 102479.

[12] G. Luo, M. Blumenthal, M. Heide, and M. Uecker. "Bayesian MRI reconstruction with joint uncertainty estimation using diffusion models." In: *Mag. Res. Med.* 90.1 (July 2023), 295–311.

[13] A. Kazerouni, E. K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, and D. Merhof. "Diffusion models in medical imaging: A comprehensive survey." In: *Med. Im. Anal.* 88 (Aug. 2023), p. 102846.

[14] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. "Score-based generative modeling through stochastic differential equations." In: *Proc. Intl. Conf. on Learning Representations*. 2021.

[15] A. Hyvärinen. "Estimation of non-normalized statistical models by score matching." In: *J. Mach. Learning Res.* 6.24 (2005), 695–709.

[16] P. Vincent. "A connection between score matching and denoising autoencoders." In: *Neural Comput.* 23.7 (July 2011), 1661–74.

[17] Y. Song and S. Ermon. "Generative modeling by estimating gradients of the data distribution." In: *NeurIPS*. 2019.

[18] B. Efron. "Tweedie's formula and selection bias." In: *J. Am. Stat. Assoc.* 106.496 (2011), 1602–14.

[19] Y. Song and S. Ermon. "Improved techniques for training score-based generative models." In: *NeurIPS*. Vol. 33. 2020, 12438–48.

[20] T. Salimans and J. Ho. "Progressive distillation for fast sampling of diffusion models." In: *Proc. Intl. Conf. on Learning Representations*. 2022.

[21] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. *Consistency models*. 2023.

[22] H. Chung, S. Lee, and J. C. Ye. "Decomposed diffusion sampler for accelerating large-scale inverse problems." In: *Proc. Intl. Conf. on Learning Representations*. 2024.

[23] F. Guth, S. Coste, V. D. Bortoli, and Stéphane Mallat. "Wavelet score-based generative modeling." In: *NeurIPS*. 2022.

[24] Z. Kadkhodaie, F. Guth, Stephane Mallat, and E. P. Simoncelli. "Learning multi-scale local conditional probability models of images." In: *Proc. Intl. Conf. on Learning Representations*. 2023.

[25] D. Ryu and J. C. Ye. *Pyramidal denoising diffusion probabilistic models*. 2022.

[26] S. Lee, H. Chung, J. Kim, and J. C. Ye. "Progressive deblurring of diffusion models for coarse-to-fine image synthesis." In: *NeurIPS Workshop SBM*. 2022.

[27] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. "DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps." In: *NeurIPS*. 2022.

[28] H. Chung, B. Sim, and J. C. Ye. "Come-closer-diffuse-faster: accelerating conditional diffusion models for inverse problems through stochastic contraction." In: *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*. 2022, 12403–12.

[29] A. Vahdat, K. Kreis, and J. Kautz. "Score-based generative modeling in latent space." In: *NeurIPS*. 2021.

[30] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and Bjorn Ommer. "High-resolution image synthesis with latent diffusion models." In: *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*. 2022, 10674–85.

[31] K. C. Tezcan, N. Karani, C. F. Baumgartner, and E. Konukoglu. "Sampling possible reconstructions of undersampled acquisitions in MR imaging with a deep learned prior." In: *IEEE Trans. Med. Imag.* 41.7 (July 2022), 1885–96.

[32] H. Chung, D. Ryu, M. T. McCann, M. L. Klasky, and J. C. Ye. "Solving 3D inverse problems using pre-trained 2D diffusion models." In: *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*. 2023, 22542–51.

[33] S. Lee, H. Chung, M. Park, J. Park, W-S. Ryu, and J. C. Ye. "Improving 3D imaging with pre-trained perpendicular 2D diffusion models." In: *Proc. Intl. Conf. Comp. Vision*. 2023.

[34] G. Wang, T. Luo, J-F. Nielsen, D. C. Noll, and J. A. Fessler. "B-spline parameterized joint optimization of reconstruction and k-space trajectories (BJORK) for accelerated 2D MRI." In: *IEEE Trans. Med. Imag.* 41.9 (Sept. 2022), 2318–30.

[35] W. Wu and K. L. Miller. "Image formation in diffusion MRI: A review of recent technical developments." In: *J. Mag. Res. Im.* 46.3 (Sept. 2017), 646–62.

[36] S. Bhadra, W. Zhou, and M. A. Anastasio. "Medical image reconstruction with image-adaptive priors learned by use of generative adversarial networks." In: *Proc. SPIE 11312 Medical Imaging: Phys. Med. Im.* 2020, p. 113120V.

[37] C. Li and M. Wand. "Precomputed real-time texture synthesis with Markovian generative adversarial networks." In: *Proc. European Comp. Vision Conf.* 2016, 702–16.

[38] P. Isola, J-Y. Zhu, T. Zhou, and A. A. Efros. "Image-to-image translation with conditional adversarial networks." In: *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*. 2017, 5967–76.

[39] A. Elnekave and Y. Weiss. "Generating natural images with direct patch distributions matching." In: *Proc. European Comp. Vision Conf.* Vol. 13677. 2022.

[40] M. Aharon, M. Elad, and A. Bruckstein. "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation." In: *IEEE Trans. Sig. Proc.* 54.11 (Nov. 2006), 4311–22.

[41] S. Ravishankar and Y. Bresler. "MR image reconstruction from highly undersampled k-space data by dictionary learning." In: *IEEE Trans. Med. Imag.* 30.5 (May 2011), 1028–41.

[42] J. Hertrich, A. Houdard, and C. Redenbach. "Wasserstein patch prior for image superresolution." In: *IEEE Trans. Computational Imaging* 8 (2022), 693–704.

[43] F. Altekruger and J. Hertrich. "WPPNets and WPPFlows: The power of Wasserstein patch priors for superresolution." In: *SIAM J. Imaging Sci.* 16.3 (2023), 1033–67.

[44] G. Vaksman, M. Zibulevsky, and M. Elad. "Patch ordering as a regularization for inverse problems in image processing." In: *SIAM J. Imaging Sci.* 9.1 (2016), 287–319.

[45] M. Piening, F. Altekruger, J. Hertrich, P. Hagemann, A. Walther, and G. Steidl. *Learning from small data sets: Patch-based regularizers in inverse problems for image reconstruction*. 2023.

[46] J. A. Fessler, J. Hu, and X. Xu. "Generalizability (or not?) of patch-based image models." In: *BASP*. Invited presentation. 2023.

[47] U. S. Kamilov, E. Bostan, and M. Unser. “Variational justification of cycle spinning for wavelet-based solutions of inverse problems.” In: *IEEE Signal Proc. Letters* 21.11 (Nov. 2014), 1326–30.

[48] A. Saucedo, S. Lefkimmiatis, N. Rangwala, and K. Sung. “Improved computational efficiency of locally low rank MRI reconstruction using iterative random patch adjustments.” In: *IEEE Trans. Med. Imag.* 36.6 (2017), 1209–20.

[49] J. L. Rumberger, X. Yu, P. Hirsch, M. Dohmen, V. E. Guarino, A. Mokarian, L. Mais, J. Funke, and D. Kainmueller. “How shift equivariance impacts metric learning for instance segmentation.” In: *Proc. Intl. Conf. Comp. Vision*. 2021, 7108–16.

[50] O. Ozdenizci and R. Legenstein. “Restoring vision in adverse weather conditions with patch-based denoising diffusion models.” In: *IEEE Trans. Patt. Anal. Mach. Int.* 45.8 (Jan. 2023), 10346–57.

[51] G. E. Hinton. “Training products of experts by minimizing contrastive divergence.” In: *Neural Computation* 14.8 (Aug. 2002), 1771–800.

[52] S. Roth and M. J. Black. “Fields of experts.” In: *Intl. J. Comp. Vision* 82.2 (Jan. 2009), 205–29.

[53] D. P. Kingma and Y. LeCun. “Regularized estimation of image statistics by score matching.” In: *NeurIPS*. 2010, 1126–34.

[54] R. R. Coifman and D. L. Donoho. *Translation-invariant denoising*. 1995.

[55] M. A. T. Figueiredo and R. D. Nowak. “An EM algorithm for wavelet-based image restoration.” In: *IEEE Trans. Im. Proc.* 12.8 (Aug. 2003), 906–16.

[56] U. Kamilov, E. Bostan, and M. Unser. “Wavelet shrinkage with consistent cycle spinning generalizes total variation denoising.” In: *IEEE Signal Proc. Letters* 19.4 (Apr. 2012), 187–90.

[57] F. Ong and M. Lustig. “Beyond low rank + sparse: multiscale low rank matrix decomposition.” In: *IEEE J. Sel. Top. Sig. Proc.* 10.4 (June 2016), 672–87.

[58] Z. Wang, Y. Jiang, H. Zheng, P. Wang, P. He, Z. Wang, W. Chen, and M. Zhou. “Patch diffusion: faster and more data-efficient training of diffusion models.” In: *NeurIPS*. Vol. 36. 2023, 72137–54.

Bibliography VI

[59] T. Karras, M. Aittala, T. Aila, and S. Laine. "Elucidating the design space of diffusion-based generative models." In: *NeurIPS*. 2022.

[60] J. Hu, B. Song, X. Xu, L. Shen, and J. A. Fessler. "Learning image priors through patch-based diffusion models for solving inverse problems." In: *NeurIPS*. 2024.

[61] C. H. McCollough, A. C. Bartley, R. E. Carter, B. Chen, T. A. Drees, P. Edwards, D. R. Holmes, A. E. Huang, F. Khan, S. Leng, K. L. McMillan, G. J. Michalak, K. M. Nunez, L. Yu, and J. G. Fletcher. "Low-dose CT for the detection and classification of metastatic liver lesions: Results of the 2016 Low Dose CT Grand Challenge." In: *Med. Phys.* 44.10 (Oct. 2017), e339–52.

[62] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. "Diffusion posterior sampling for general noisy inverse problems." In: *Proc. Intl. Conf. on Learning Representations*. 2023.

[63] X. Xu, J. Liu, Y. Sun, B. Wohlberg, and U. S. Kamilov. "Boosting the performance of plug-and-play priors via denoiser scaling." In: *Proc., IEEE Asilomar Conf. on Signals, Systems, and Comp.* 2020, 1305–12.

[64] Y. Hu, J. Liu, X. Xu, and U. S. Kamilov. "Monotonically convergent regularization by denoising." In: *Proc. IEEE Intl. Conf. on Image Processing*. 2022, 426–30.

[65] Y. Wang, J. Yu, and J. Zhang. "Zero-shot image restoration using denoising diffusion null-space model." In: *Proc. Intl. Conf. Mach. Learn.* 2023.

[66] B. Song, J. Hu, Z. Luo, J. A. Fessler, and L. Shen. "DiffusionBlend: learning 3D image prior through position-aware diffusion score blending for 3D computed tomography reconstruction." In: *NeurIPS*. 2024.

[67] D. Ulyanov, A. Vedaldi, and V. Lempitsky. "Deep image prior." In: *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*. 2018, 9446–54.

[68] J. Liu, Y. Sun, X. Xu, and U. S. Kamilov. "Image restoration using total variation regularized deep image prior." In: *Proc. IEEE Conf. Acoust. Speech Sig. Proc.* 2019, 7715–9.

[69] Y. Jo, S. Y. Chun, and J. Choi. "Rethinking deep image prior for denoising." In: *Proc. Intl. Conf. Comp. Vision*. 2021, 5067–76.

[70] R. Barbano, J. Leuschner, M. Schmidt, A. Denker, A. Hauptmann, P. Maass, and B. Jin. "An educated warm start for deep image prior-based micro CT reconstruction." In: *IEEE Trans. Computational Imaging* 8 (2022), 1210–22.

[71] H. Chung and J. C. Ye. "Deep diffusion image prior for Efficient OOD adaptation in 3D inverse problems." In: *Proc. European Comp. Vision Conf.* 2024, 432–55.

[72] R. Barbano, A. Denker, H. Chung, T. H. Roh, S. Arrdige, P. Maass, B. Jin, and J. C. Ye. "Steerable conditional diffusion for out-of-distribution adaptation in imaging inverse problems." In: *IEEE Trans. Med. Imag.* 44.5 (May 2025), 2093–104.

[73] J. Hu, B. Song, J. A. Fessler, and L. Shen. *Patch-based diffusion models beat whole-image models for mismatched distribution inverse problems*. 2024.

[74] J. Hu, B. Song, J. A. Fessler, and L. Shen. "Test-time adaptation improves inverse problem solving with patch-based diffusion models." In: *IEEE Trans. Computational Imaging* 11 (July 2025), 980–91.

[75] J. Hu, Z. Li, B. Song, L. Shen, and J. A. Fessler. "SPAR: refining a single pretrained diffusion model to solve inverse problems in many modalities." In: *NeurIPS*. Submitted. 2025.