An introduction to score-based generative models

Jeffrey A. Fessler

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

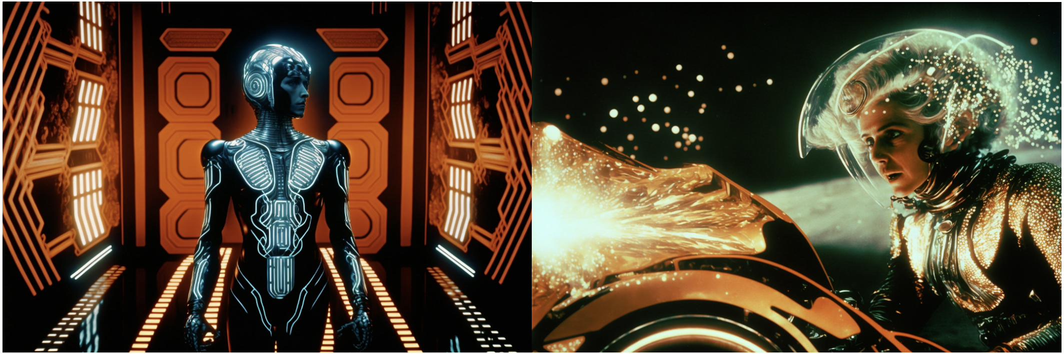
http://web.eecs.umich.edu/~fessler

MIDAS AI in Science and Engineering Day
2023-04-03

Acknowledgments:
Jason Hu, Xiaojian Xu, Mike McCann (LANL)

Outline

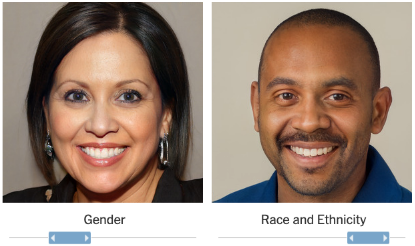
Generative models are hot in graphics



Computer ("AI") generated stills from hypothetical movie: Chilean director Alejandro Jodorowsky's 1976 version of "Tron" using midjourney.com as reported in 2023-01-13 NY Times article "This film does not exist" by director Frank Pavich.

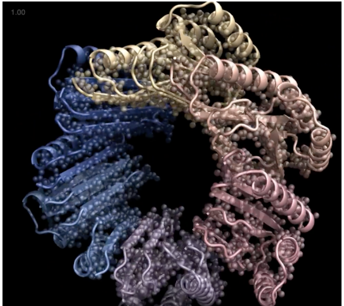
Generative models are hot in the news

- 2020-11-21 NY Times “Designed to Deceive: Do These People Look Real to You?”
  Article about generated (aka fake) faces.



- 2022-10-21 NY Times “A Coming-Out Party for Generative A.I., Silicon Valley’s New Craze”
  (about “Stable Diffusion” image generator)
  https://nyti.ms/3SjsN0k

- 2023-01-09 NY Times “A.I. Turns Its Artistry to Creating New Human Proteins”
  https://nyti.ms/3IzY66m

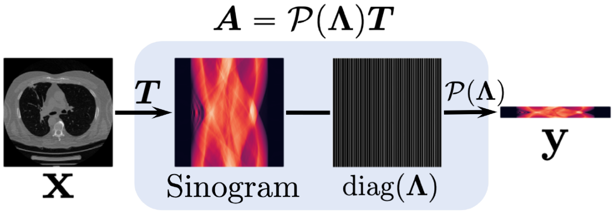


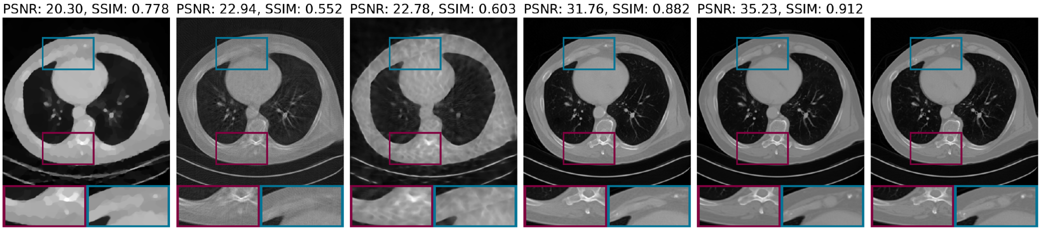
Generative models are hot in imaging / inverse problems

Zhao, Ye, Bresler: Jan. 2023 IEEE SpMag survey paper [1]

- Generative adversarial network (GAN) models
- Variation auto-encoder (VAE) models [2]
- Normalizing flows [3]
- Score-based diffusion models
  - Ramzi et al., NeurIPS 2020 [4]
  - Yang Song et al., NeurIPS 2021, ICLR 2022 [5, 6]
  - Jalal et al., NeurIPS 2021 [7]
  - Chung & Ye, MIA, Aug. 2022 [8]
  - Luo et al., 2022 arXiv 2202.01479 [9]
  - ...
- Kazerouni et al. [10] have github catalog, including 5 survey papers
- ... (hopelessly incomplete lists)

From Song et al., ICLR 2022, trained with 47K 2D images, 23 projection views [6]





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

Generative models: What

- A *generative model* is a model for some probability distribution $\mathrm{p}(\boldsymbol{x})$,
-

Generative models: What

- A *generative model* is a model for some probability distribution $\mathrm{p}(\boldsymbol{x})$,
- and usually a method for drawing samples from that distribution.
  "generation" (think: random number generator)
-

Generative models: What

- A *generative model* is a model for some probability distribution $\mathrm{p}(\boldsymbol{x})$,
- and usually a method for drawing samples from that distribution.
  "generation" (think: random number generator)
- Usually the model depends on some (or many) parameters, $\boldsymbol{\theta}$.
  i.e., we work with a parametric model $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$.
-

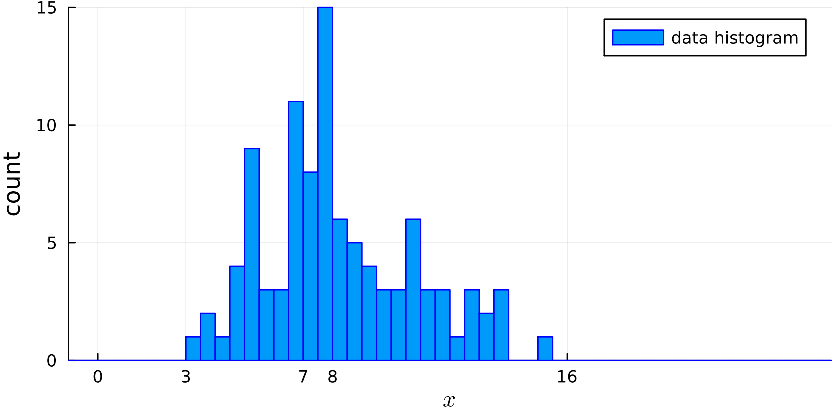
Generative models: What

- A *generative model* is a model for some probability distribution $\mathrm{p}(\boldsymbol{x})$,
- and usually a method for drawing samples from that distribution.
  "generation" (think: random number generator)
- Usually the model depends on some (or many) parameters, $\boldsymbol{\theta}$.
  i.e., we work with a parametric model $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$.
- Challenge 1: Learning the parameters $\boldsymbol{\theta}$ from some training data.
  Hopefully that data is representative of the population of interest.
  (skin color, brain lesions...)
-

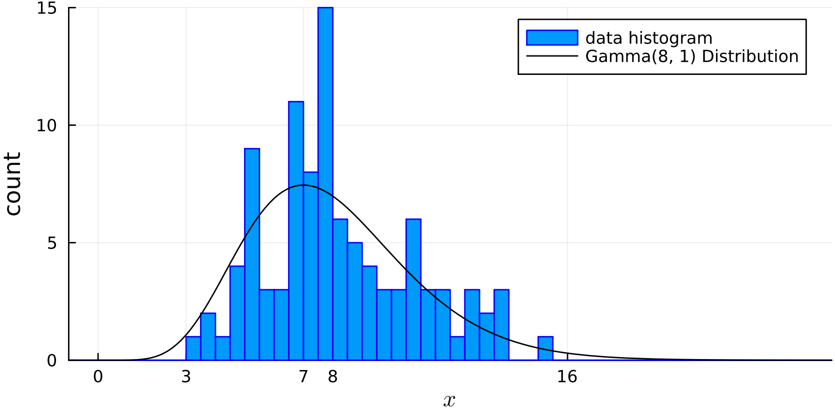
Generative models: What

- A *generative model* is a model for some probability distribution $\mathrm{p}(\boldsymbol{x})$,
- and usually a method for drawing samples from that distribution.
  "generation" (think: random number generator)
- Usually the model depends on some (or many) parameters, $\boldsymbol{\theta}$.
  i.e., we work with a parametric model $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$.
- Challenge 1: Learning the parameters $\boldsymbol{\theta}$ from some training data.
  Hopefully that data is representative of the population of interest.
  (skin color, brain lesions...)
- Challenge 2: Drawing samples efficiently from $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$.

Toy example



Toy example



- Maximum-likelihood (ML) estimation to fit the two Gamma distribution parameters requires an iterative method.
- After fitting, drawing samples (aka generation) is not trivial, involving an acceptance-rejection method.

- Generating samples, *e.g.*, computer-generated graphics, music, poetry . . .
  -

Generative models: Why

- Generating samples, *e.g.*, computer-generated graphics, music, poetry . . .
  - Often those examples involve conditional distributions $p_{\theta}(\boldsymbol{x}|\boldsymbol{z})$
  - "generate a jazz melody in the style of Miles Davis"
-

Generative models: Why

- Generating samples, *e.g.*, computer-generated graphics, music, poetry . . .
  - Often those examples involve conditional distributions $\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$
  - "generate a jazz melody in the style of Miles Davis"
- ChatGPT: *Write a short joke about AI*
  -

Generative models: Why

- Generating samples, *e.g.*, computer-generated graphics, music, poetry ...
  - Often those examples involve conditional distributions $\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x} | \boldsymbol{z})$
  - "generate a jazz melody in the style of Miles Davis"
- ChatGPT: *Write a short joke about AI*
  - `Why did the AI cross the road? To get to the other side of the algorithm!`
  - `Why did the AI cross the road? To get to the other cache.`
  - `Why did the AI cross the road? To get to the other dataset!`
-

Generative models: Why

- Generating samples, *e.g.*, computer-generated graphics, music, poetry ...
  - Often those examples involve conditional distributions $\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$
  - "generate a jazz melody in the style of Miles Davis"
- ChatGPT: *Write a short joke about AI*
  - `Why did the AI cross the road? To get to the other side of the algorithm!`
  - `Why did the AI cross the road? To get to the other cache.`
  - `Why did the AI cross the road? To get to the other dataset!`
- *Is ChatGPT based on a generative model?*

  `Yes, ChatGPT is based on a generative language model. Specifically, it is built using the GPT (Generative Pre-trained Transformer) architecture, which is a type of deep neural network designed for generating natural language text.`

Generative models: Why

- Bayesian inference
  - ◦

Generative models: Why

- Bayesian inference
  - Given (test) data $\boldsymbol{y}$ related to a latent variable $\boldsymbol{x}$
  - Known likelihood function $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$, e.g., for human-made sensors
  - Want to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$
  -

Generative models: Why

- Bayesian inference
  - Given (test) data $\boldsymbol{y}$ related to a latent variable $\boldsymbol{x}$
  - Known likelihood function $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$, e.g., for human-made sensors
  - Want to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$
  - Maximum-likelihood estimation is insufficient for under-determined problems

$$\arg\max_{\boldsymbol{x}} \mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$$

    (sparse-view X-ray CT, accelerated MRI scans, ...)
  - Bayesian methods use the posterior

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{y}) = \frac{\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{p}(\boldsymbol{x})}{\mathrm{p}(\boldsymbol{y})}$$

  -

- Bayesian inference
  - Given (test) data $\boldsymbol{y}$ related to a latent variable $\boldsymbol{x}$
  - Known likelihood function $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$, e.g., for human-made sensors
  - Want to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$
  - Maximum-likelihood estimation is insufficient for under-determined problems

$$\arg\max_{\boldsymbol{x}} \mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$$

    (sparse-view X-ray CT, accelerated MRI scans, ...)
  - Bayesian methods use the posterior

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{y}) = \frac{\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})\,\mathrm{p}(\boldsymbol{x})}{\mathrm{p}(\boldsymbol{y})}$$

  - Here the prior $\mathrm{p}(\boldsymbol{x})$ is for quantifying probability, not generation.

Benefits of Bayesian methods

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 facilitates uncertainty quantification in inference

All of these require the prior $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$.

Benefits of Bayesian methods

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 facilitates uncertainty quantification in inference

All of these require the prior $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$. Or do they?

Sampling and MAP estimation just need its score function $\nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$, using Langevin dynamics, aka stochastic gradient ascent of log-likelihood:

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

- ◦ Draws samples from $\mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$ for suitable choices of $\{\alpha_t\}$, $\{\beta_t\}$, and (large) T [11].
- ◦ So how do we learn a score function?

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

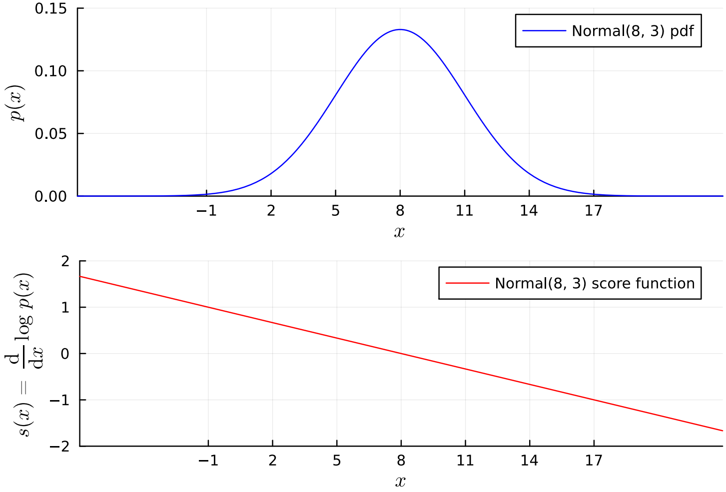
- Typical distribution models: $\mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\,\mathrm{e}^{-U(\boldsymbol{x};\boldsymbol{\theta})}$.
  Goal: learn $\boldsymbol{\theta}$ from training data $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T$
- For IID samples $\{\boldsymbol{x}_t\}$,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

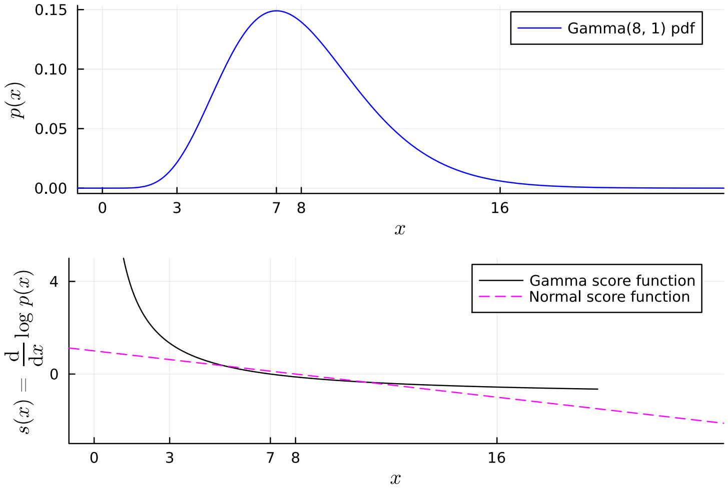
$$\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) \triangleq \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{x}} \left(-\log Z(\boldsymbol{\theta}) - U(\boldsymbol{x};\theta)\right) = -\nabla_{\boldsymbol{x}} U(\boldsymbol{x};\theta).$$

Score function example: 1D Gaussian



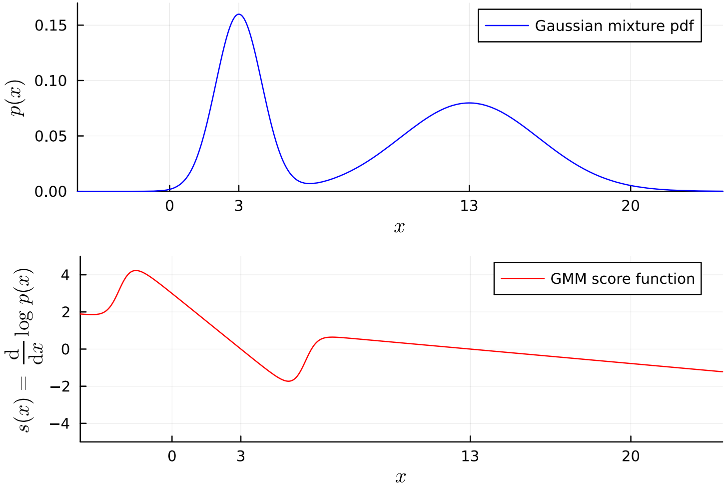
$$\mathsf{p}(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, \mathrm{e}^{-(x-\mu)^2/2\sigma^2} \implies s(x) = \frac{\mathrm{d}}{\mathrm{d}x} \log \mathsf{p}(x) = \frac{1}{\sigma^2}(x - \mu)$$

Score function example: 1D Gamma

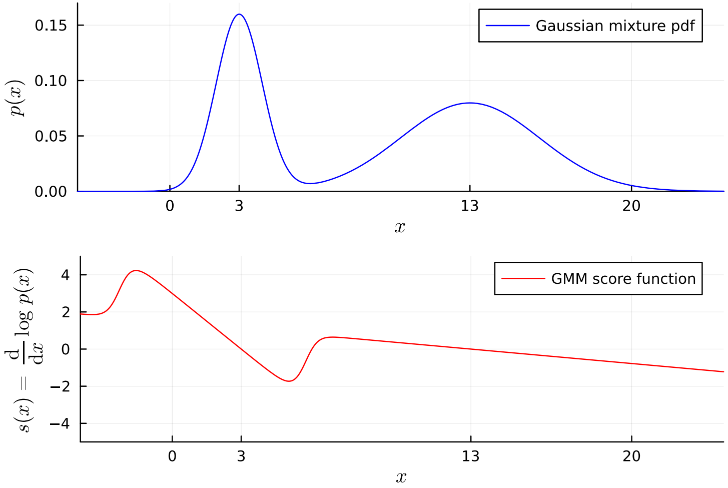


Note sign of score function to left and right of mode.

Score function example: 1D GMM



Score function example: 1D GMM

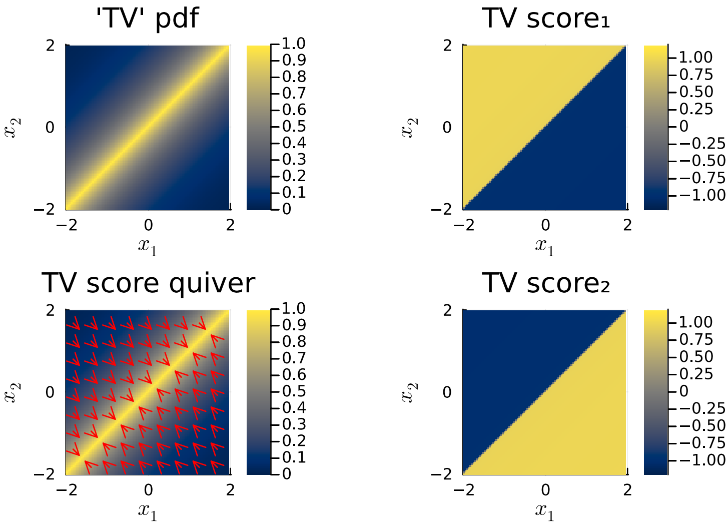


- Could you recover the pdf $\mathrm{p}(x)$ from its score function $s(x)$ in 1D?

Score function example: 2D

Total variation (TV) prior for 2×1 patch:

$$p(\boldsymbol{x}) \propto e^{-\beta |x_2 - x_1|}$$



Score matching

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

Score matching

- Given training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, learn score function $\boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}) = \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$
- Explicit score matching (ESM)
  - Estimate data distribution (kernel density estimation): $\mathrm{q}_\sigma(\boldsymbol{x}) = \sum_{t=1}^{T} g_\sigma(\boldsymbol{x} - \boldsymbol{x}_i)$
  - Match model score to data score (Hyvärinen, 2005 [12]):

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J_{\mathrm{ESM,q}}(\boldsymbol{\theta}), \quad J_{\mathrm{ESM,q}}(\boldsymbol{\theta}) \triangleq \frac{1}{2} \mathrm{E}_{\mathrm{q}(\boldsymbol{x})}\left[\left\| \boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}) - \nabla \log \mathrm{q}(\boldsymbol{x}) \right\|_2^2 \right]. \tag{1}$$

-

- Given training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, learn score function $\boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}) = \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta})$
- Explicit score matching (ESM)
  - Estimate data distribution (kernel density estimation): $\mathrm{q}_\sigma(\boldsymbol{x}) = \sum_{t=1}^T g_\sigma(\boldsymbol{x} - \boldsymbol{x}_i)$
  - Match model score to data score (Hyvärinen, 2005 [12]):

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J_{\mathrm{ESM,q}}(\boldsymbol{\theta}), \quad J_{\mathrm{ESM,q}}(\boldsymbol{\theta}) \triangleq \frac{1}{2} \mathrm{E}_{\mathrm{q}(\boldsymbol{x})}\left[\| \boldsymbol{s}(\boldsymbol{x}; \boldsymbol{\theta}) - \nabla \log \mathrm{q}(\boldsymbol{x}) \|_2^2 \right]. \tag{1}$$

- Implicit score matching (ISM):

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J_{\mathrm{ISM,q_0}}(\boldsymbol{\theta}), \quad J_{\mathrm{ISM,q_0}}(\boldsymbol{\theta}) = \frac{1}{T} \sum_{t=1}^T \sum_i \left(\partial_i s_i(\boldsymbol{x}_t; \boldsymbol{\theta}) + \frac{1}{2} \left| s_i(\boldsymbol{x}_t; \boldsymbol{\theta}) \right|^2 \right) \tag{2}$$

$$\partial_i s_i(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{\partial}{\partial x_i} s_i(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{\partial^2}{\partial x_i^2} \log \mathrm{p}(\boldsymbol{x}; \boldsymbol{\theta}). \tag{3}$$

-

- Given training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, learn score function $\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x};\boldsymbol{\theta})$
- Explicit score matching (ESM)
  - Estimate data distribution (kernel density estimation): $\mathrm{q}_\sigma(\boldsymbol{x}) = \sum_{t=1}^T g_\sigma(\boldsymbol{x} - \boldsymbol{x}_i)$
  - Match model score to data score (Hyvärinen, 2005 [12]):

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J_{\mathrm{ESM,q}}(\boldsymbol{\theta}), \quad J_{\mathrm{ESM,q}}(\boldsymbol{\theta}) \triangleq \frac{1}{2} \mathrm{E}_{\mathrm{q}(\boldsymbol{x})}\left[\| \boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) - \nabla \log \mathrm{q}(\boldsymbol{x}) \|_2^2 \right]. \quad (1)$$

- Implicit score matching (ISM):

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J_{\mathrm{ISM,q_0}}(\boldsymbol{\theta}), \quad J_{\mathrm{ISM,q_0}}(\boldsymbol{\theta}) = \frac{1}{T} \sum_{t=1}^T \sum_i \left(\partial_i s_i(\boldsymbol{x}_t;\boldsymbol{\theta}) + \frac{1}{2} |s_i(\boldsymbol{x}_t;\boldsymbol{\theta})|^2 \right) \quad (2)$$

$$\partial_i s_i(\boldsymbol{x};\boldsymbol{\theta}) = \frac{\partial}{\partial x_i} s_i(\boldsymbol{x};\boldsymbol{\theta}) = \frac{\partial^2}{\partial x_i^2} \log \mathrm{p}(\boldsymbol{x};\boldsymbol{\theta}). \quad (3)$$

- Denoising score matching (DSM)

- Vincent, 2011 [13] showed this remarkable equivalence between ESM and DSM:

$$
\begin{aligned}
J_{\mathrm{ESM},\mathsf{q}_\sigma}(\boldsymbol{\theta}) &= \mathsf{E}_{\mathsf{q}_\sigma(\boldsymbol{x})}\left[\frac{1}{2}\left\|\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) - \nabla \log \mathsf{q}_\sigma(\boldsymbol{x})\right\|_2^2\right] \\
&= \frac{1}{T}\sum_{t=1}^{T}\int \frac{1}{2}\left\|\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) - \nabla \log \mathsf{q}_\sigma(\boldsymbol{x})\right\|_2^2 \, g_\sigma(\boldsymbol{x} - \boldsymbol{x}_t)\,\mathrm{d}\boldsymbol{x} \\
\overset{!}{=} J_{\mathrm{DSM},\mathsf{q}_\sigma}(\boldsymbol{\theta}) &\triangleq \frac{1}{T}\sum_{t=1}^{T}\mathsf{E}_{g_\sigma(\boldsymbol{z})}\left[\frac{1}{2}\left\|\boldsymbol{s}(\boldsymbol{x}_t + \boldsymbol{z};\boldsymbol{\theta}) + \frac{\boldsymbol{z}}{\sigma^2}\right\|_2^2\right] \\
&\approx \frac{1}{T}\sum_{t=1}^{T}\frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\left\|\boldsymbol{s}(\boldsymbol{x}_t + \boldsymbol{z}_{t,m};\boldsymbol{\theta}) + \frac{\boldsymbol{z}_{t,m}}{\sigma^2}\right\|_2^2
\end{aligned}
$$

-

- Vincent, 2011 [13] showed this remarkable equivalence between ESM and DSM:

$$
\begin{aligned}
J_{\mathrm{ESM},\mathsf{q}_\sigma}(\boldsymbol{\theta}) &= \mathsf{E}_{\mathsf{q}_\sigma(\boldsymbol{x})}\left[\frac{1}{2}\left\|\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) - \nabla \log \mathsf{q}_\sigma(\boldsymbol{x})\right\|_2^2\right] \\
&= \frac{1}{T}\sum_{t=1}^{T}\int \frac{1}{2}\left\|\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}) - \nabla \log \mathsf{q}_\sigma(\boldsymbol{x})\right\|_2^2 g_\sigma(\boldsymbol{x}-\boldsymbol{x}_t)\,\mathrm{d}\boldsymbol{x} \\
\stackrel{!}{=} J_{\mathrm{DSM},\mathsf{q}_\sigma}(\boldsymbol{\theta}) &\triangleq \frac{1}{T}\sum_{t=1}^{T}\mathsf{E}_{g_\sigma(\boldsymbol{z})}\left[\frac{1}{2}\left\|\boldsymbol{s}(\boldsymbol{x}_t+\boldsymbol{z};\boldsymbol{\theta}) + \frac{\boldsymbol{z}}{\sigma^2}\right\|_2^2\right] \\
&\approx \frac{1}{T}\sum_{t=1}^{T}\frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\left\|\boldsymbol{s}(\boldsymbol{x}_t+\boldsymbol{z}_{t,m};\boldsymbol{\theta}) + \frac{\boldsymbol{z}_{t,m}}{\sigma^2}\right\|_2^2
\end{aligned}
$$

- The last term is a kind of denoising operation.

Noise-conditional score matching

- Q: How much noise (what σ) to use in DSM?
-

Noise-conditional score matching

- Q: How much noise (what σ) to use in DSM?
- A: many!
-

Noise-conditional score matching

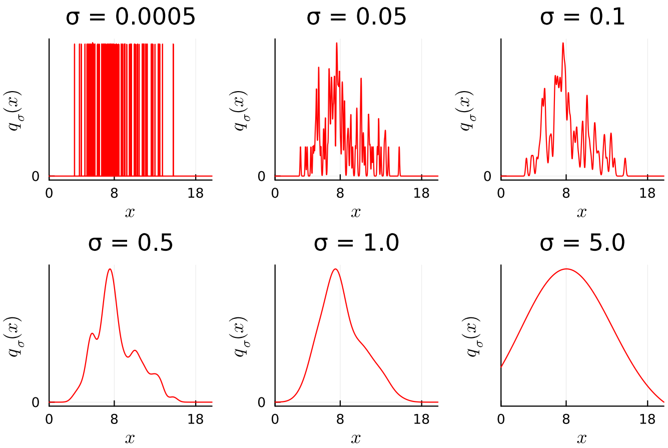
- Q: How much noise (what σ) to use in DSM?
- A: many!
- Noise-conditional score matching (NCSM) [14, eqn. (5)]:

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

  where $\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta},\sigma)$ denotes a *noise-conditional score network* (NCSN).
- Recommended choice [15]: $\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta},\sigma) \triangleq \tilde{\boldsymbol{s}}(\boldsymbol{x};\boldsymbol{\theta})/\sigma$, where $\tilde{\boldsymbol{s}}$ is unitless

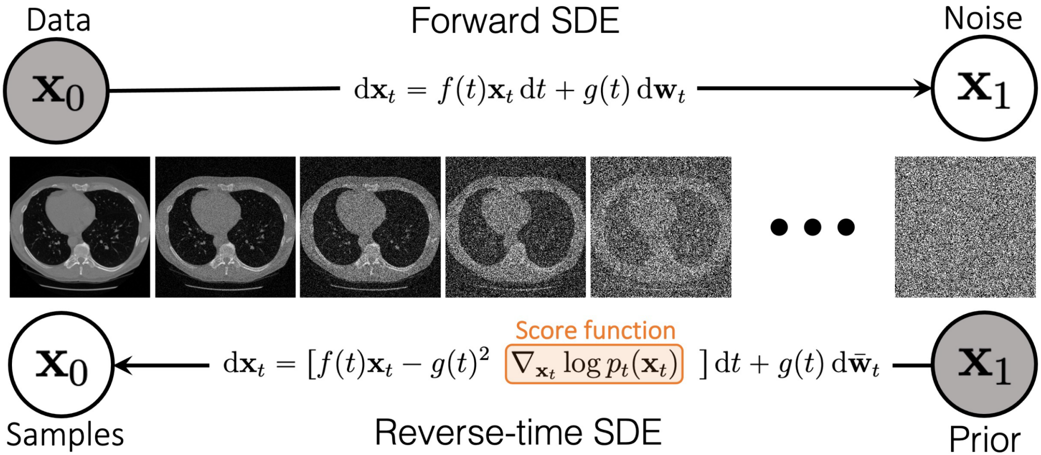
DSM 1D illustration

$T = 100$
training samples



Noise-conditional score network training / sampling

Shen & Song et al., NeurIPS 2021 [5]



Score-based diffusion models: trade-offs

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

Score-based diffusion models: trade-offs

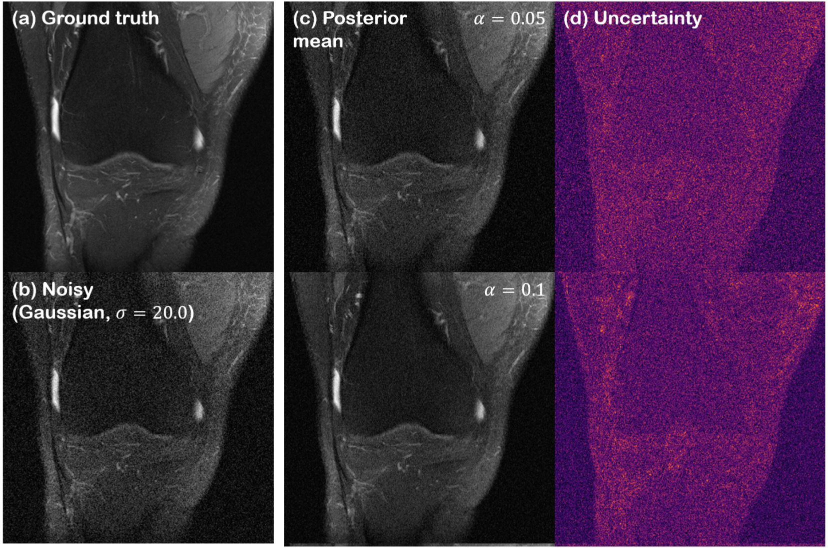
- No adversarial training needed
- High quality sample generation (if enough training data)
- Expensive sample generation (vs GAN models)
  - Distillation methods [16]
  - Consistency models [17]
  - Geometric decomposition [18]
  - Multi-scale [19, 20] and pyramidal [21] and coarse-to-fine [22] models
  - Faster ODE solvers [23]
  - Warm starts [24]
  - Latent diffusion models: use VAE and diffuse in latent space [25, 26]. Used in Stable Diffusion by start-up Stability AI
  - 3D image reconstruction using 2D models [27, 28]

Medical imaging applications

- Segmentation [29]
- Sparse-view CT reconstruction [30]
- Motion correction in MRI [31]
- Image analysis [10]
- Denoising / super-resolution [32]
- . . .

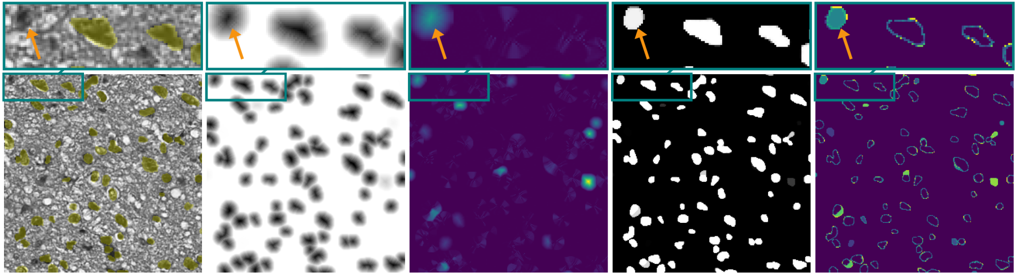
Example: MR image denoising

Chung et al., IEEE T-MI 2023 [32]



Example: Microscopy image segmentation

- Bogensperger et al., 2023 [29]; signed distance function instead of binary mask
  MoNuSeg microscopy images, Haematoxylin and Eosin (H&E) stained
  30 train, 14 test, size 1000 × 1000, >21,000 annotated nuclei

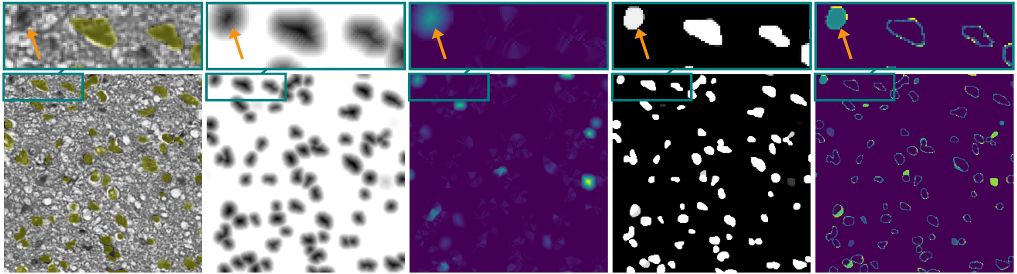


$x \otimes m_{gt}$ mean $\widetilde{m}(0)$ std $\widetilde{m}(0)$ mean m std m

-

Example: Microscopy image segmentation

- Bogensperger et al., 2023 [29]; signed distance function instead of binary mask
  MoNuSeg microscopy images, Haematoxylin and Eosin (H&E) stained
  30 train, 14 test, size 1000 × 1000, >21,000 annotated nuclei



$x \otimes m_{gt}$ | mean $\widetilde{m}(0)$ | std $\widetilde{m}(0)$ | mean m | std m

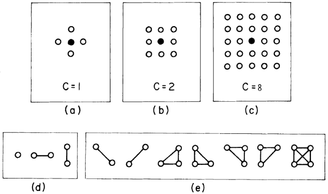
- Uncertainty map indicates possible segmentation errors

NY Times article
about fake faces

See it?

Markov random field models

(*e.g.*) Geman & Geman 1984 [33]



Mostly for inference?

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

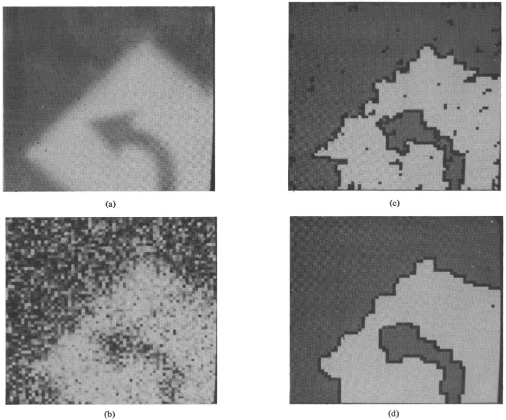


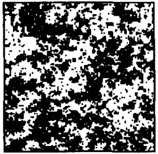
Fig. 7. (a) Blurred image (roadside scene). (b) Degraded image: Additive noise. (c) Restoration including line process; 100 iterations. (d) Restoration including line process; 1000 iterations.

MRF as generators? [34] T-PAMI 1994

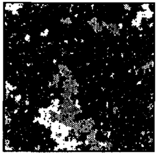
An Empirical Study of the Simulation of Various Models Used for Images

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

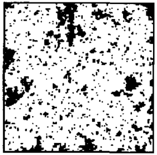
***Abstract*— Markov random fields are typically used as priors in Bayesian image restoration methods to represent spatial information in the image. Commonly used Markov random fields are not in fact capable of representing the moderate-to-large scale clustering present in naturally occurring images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



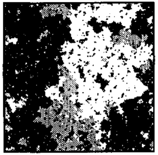
(b)



(f)



(c)

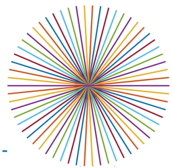


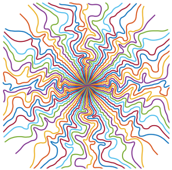
(g)

Whole images vs patches?

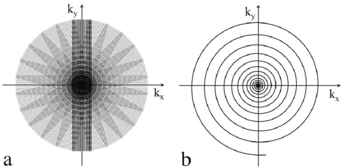
Jan. 2023 survey paper on generative models [1] does not mention “patch” once!?

MRI k-space sampling:

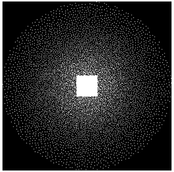




[35]



[36]



[37]

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

- patch GAN [38–40]
- patch dictionary models [41, 42]
- non-local means, BM3D . . .

Motivating questions

- Could patch-based generative models provide better robustness to distribution shifts, perhaps at the cost of reduced in-distribution performance?
- Especially in applications with very limited training data?
  e.g., dynamic MRI
- Can we use the "latest" generative models, namely score-based models, for patches?

Patch-based score modeling

- Start with MRF formulation, aka *field of experts* model [43]:

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

  - $\boldsymbol{\theta}$: parameter vector that describes the prior
  - V_c : *clique potential* for the cth image *patch*
  - $Z(\boldsymbol{\theta})$: intractable partition function
- Assume 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 $\boldsymbol{x}$
  - $V(\boldsymbol{z};\boldsymbol{\theta})$: common parent clique function

Patch-based score modeling

- 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*:

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

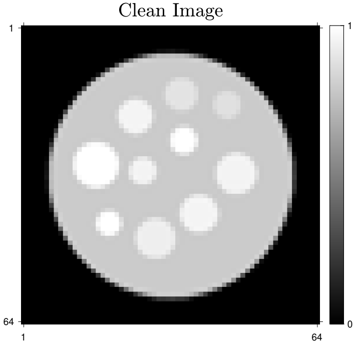
- All we must learn is the patch score function $\boldsymbol{s}_V(\boldsymbol{v}; \boldsymbol{\theta}) : \mathbb{R}^n \mapsto \mathbb{R}^n$, *e.g.*, a MLP.
- For training image patches $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_T\}$, apply *denoising score matching* (DSM) of Vincent, 2011 [13], typically for a range of noise variances σ^2 [11]:

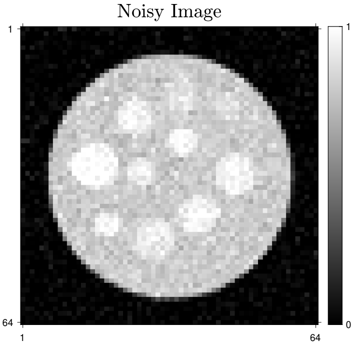
$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} \mathrm{E}_{\sigma \sim p(\sigma)} \left[\sigma^2 \, \mathrm{E}_{\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_n)} \left[\frac{1}{2} \left\| \boldsymbol{s}_V(\boldsymbol{v}_t + \boldsymbol{z}; \boldsymbol{\theta},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

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

Simple exploration with *anecdotal* results

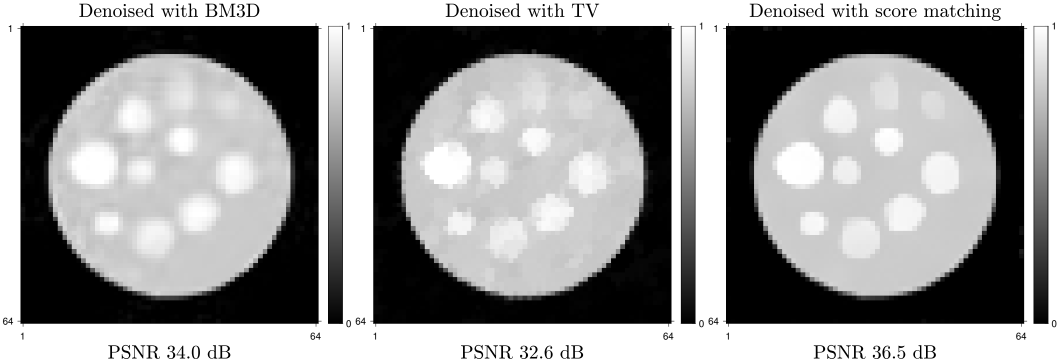
- 3×3 patches
- MLP patch score model (9, 40, 80, 160, 320, 320, 160, 80, 40, 9)
  first 5 with leaky ReLU, last 3 with tanh
- 1000 similar training examples





PSNR 26.2 dB

Denoising results



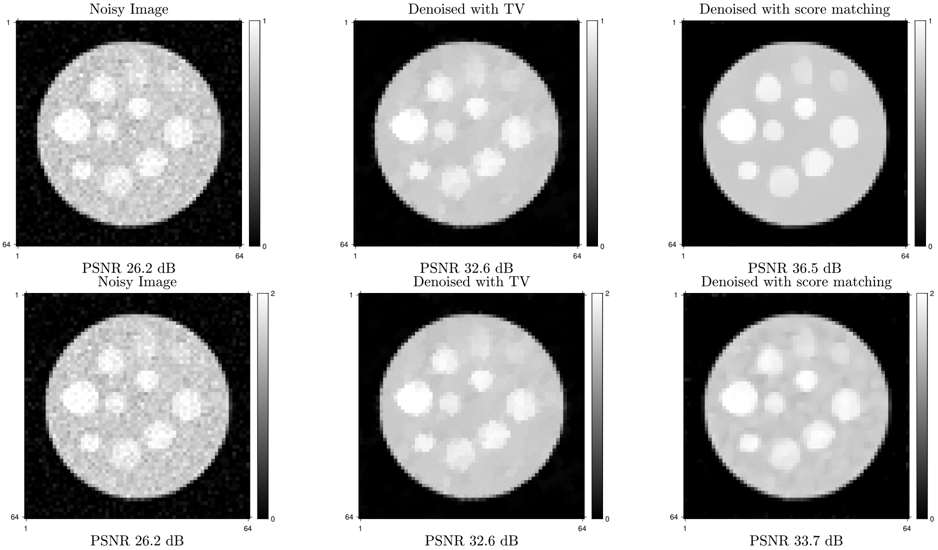
- BM3D from `https://webpages.tuni.fi/foi/GCF-BM3D`
- TV regularization parameter optimized by oracle for best PSNR.
- MAP estimate by greedy gradient ascent of log posterior: (no β!)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \nabla_{\boldsymbol{x}} \log \mathrm{p}\left(\boldsymbol{x}_k | \boldsymbol{y}; \hat{\boldsymbol{\theta}}\right) = \boldsymbol{x}_k + \alpha_k \left(\nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}_k) - \sum_c \boldsymbol{G}_c' \boldsymbol{s}_V(\boldsymbol{G}_c \boldsymbol{x}_k; \hat{\boldsymbol{\theta}})\right).$$

Generalizability to distribution shift? (pitfalls...)

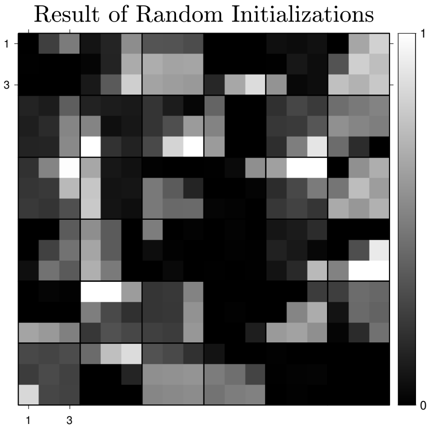
Old:

New:

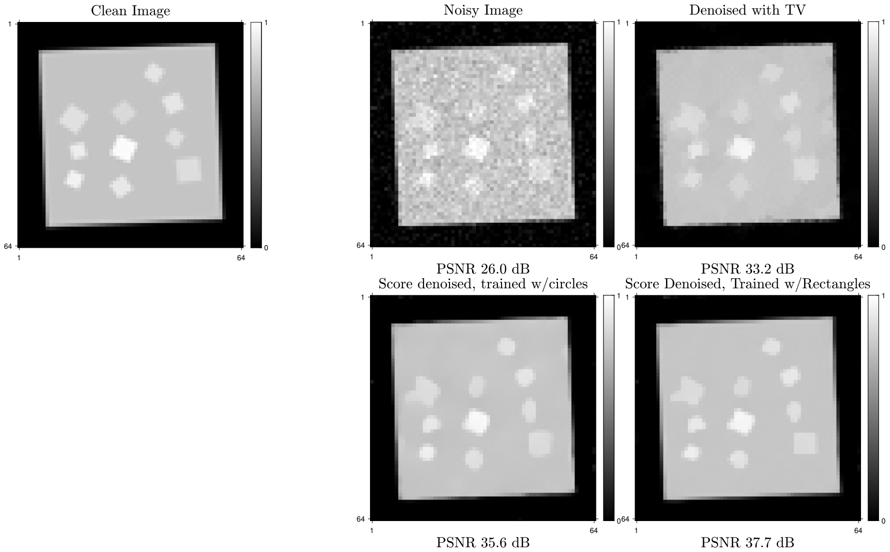


What changed?

MAP from random noise

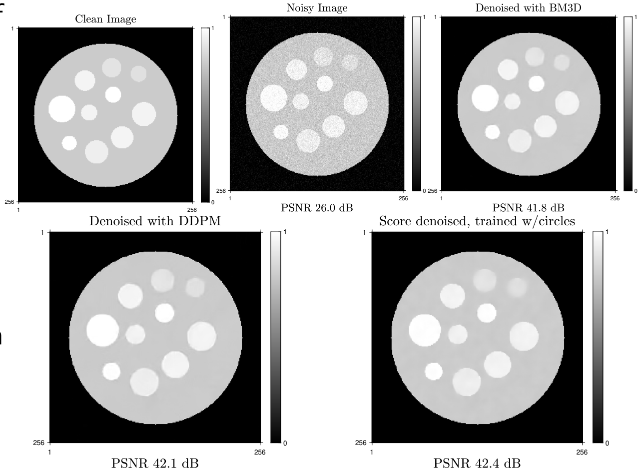


Distribution shift: rectangle test image



Whole-image vs patch models

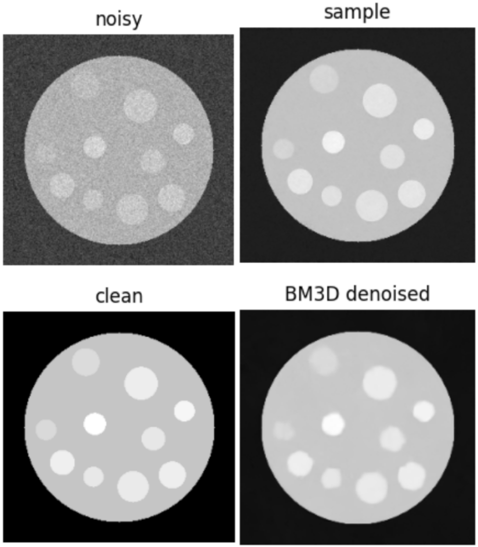
- Whole-image diffusion model of Hu et al. (SPIE, 2022) [44]
- `https://github.com/DeweiHu/OCT_DDPM`
- Based on Ho et al. (NeurIPS, 2020) [45] denoising diffusion prob. model (DDPM)
- Trained with 1000 disk images.
- Tested with noisy disk phantom
- One sample from posterior



Whole-image models and generalizability?

- Diffusion model of Hu et al. (SPIE, 2022) [44] trained with 3600 flower images.

- Tested with noisy disk phantom (PSNR 20.3 dB)
- One sample from posterior
`https://github.com/DeweiHu/OCT_DDPM`



37.0 dB

35.3 dB

Summary / future directions

- Learning patch score models is feasible with denoising score matching
- Amplitude scale invariance is not inherent to score-based models
  Easily (?) fixed by patch normalization, but what other more subtle pitfalls exist?
- Integrate invariances: amplitude scale / rotation / flip / DC offset
- Compare with whole-image models:
  - "pure" CNN score models with small receptive fields
  - multi-scale score models [19, 20]
  - ...
- Explore trade-offs between generalizability and in-distribution performance
- Is the "optimal" patch size the whole image? (Even for 3D+T?)

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

Resources

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

Bibliography I

[1] 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 (2023), 148–63.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] G. Luo, M. Heide, and M. Uecker. *MRI reconstruction via data driven Markov chain with joint uncertainty estimation*. 2022.

[10] A. Kazerouni, E. K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, and D. Merhof. *Diffusion models for medical image analysis: A comprehensive survey*. 2022.

[11] 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.

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

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

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

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

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

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

[18] H. Chung, S. Lee, and J. C. Ye. *Fast diffusion sampler for inverse problems by geometric decomposition*. 2023.

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

[20] 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.

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

[22] S. Lee, H. Chung, J. Kim, and J. C. Ye. *Progressive deblurring of diffusion models for coarse-to-fine image synthesis*. 2022.

[23] 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.

[24] 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.

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

[26] 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.

[27] H. Chung, D. Ryu, M. T. McCann, M. L. Klasky, and J. C. Ye. *Solving 3D inverse problems using pre-trained 2D diffusion models*. 2022.

[28] 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*. 2023.

[29] L. Bogensperger, D. Narnhofer, F. Ilic, and T. Pock. *Score-based generative models for medical image segmentation using signed distance functions*. 2023.

[30] W. Xia, W. Cong, and G. Wang. *Patch-based denoising diffusion probabilistic model for sparse-view CT reconstruction*. 2022.

[31] B. Levac, A. Jalal, and J. I. Tamir. *Accelerated motion correction for MRI using score-based generative models*. 2022.

[32] H. Chung, E. S. Lee, and J. C. Ye. "MR image denoising and super-resolution using regularized reverse diffusion." In: *IEEE Trans. Med. Imag.* (2023).

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

Bibliography IV

[39] 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.

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

[41] 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.

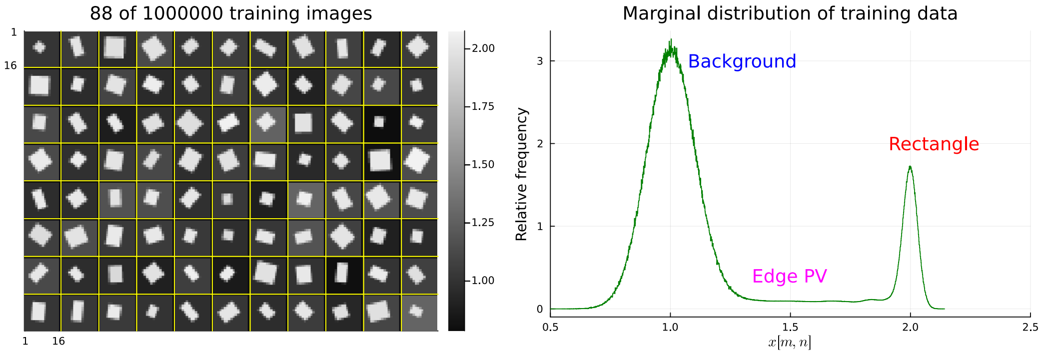
[42] S. Ravishankar and Y. Bresler. "MR image reconstruction from highly undersampled k-space data by dictionary learning." In: *IEEE Trans. Med. Imag.* 30.5 (May 2011), 1028–41.

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

[44] D. Hu, Y. K. Tao, and I. Oguz. "Unsupervised denoising of retinal OCT with diffusion probabilistic model." In: *spie-12032*. 2022, p. 1203206.

[45] J. Ho, A. Jain, and P. Abbeel. "Denoising diffusion probabilistic models." In: *NeurIPS*. Vol. 33. 2020, 6840–51.

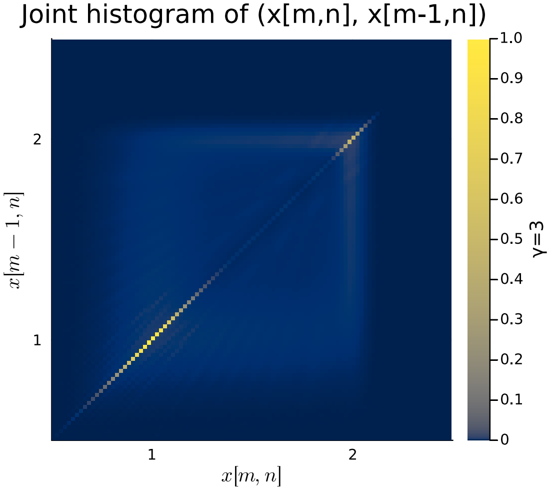
A simple exploration



- Stochastic image model with random: center, width, orientation, background $\mathcal{N}(1, 0.1^2)$, rectangle foreground $\mathcal{N}(1, 0.03^2)$
- 10^6 training images of size 16×16 with partial volume effects.
- Data lies on 7-dimensional manifold.

Patch statistics: joint distribution

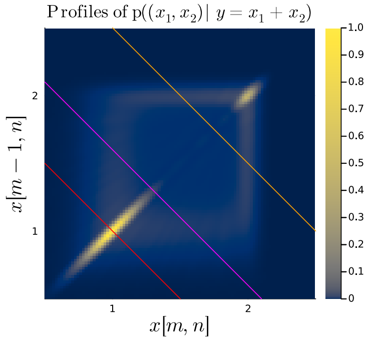
2×1 patches (cf TV)

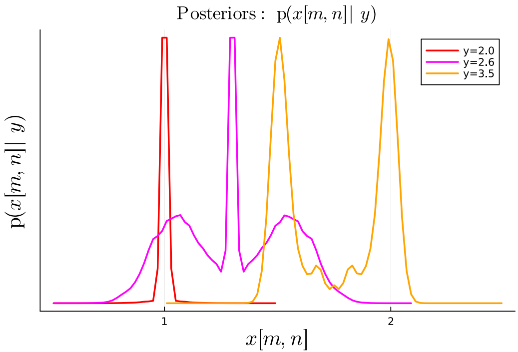


Patch statistics: posterior distributions

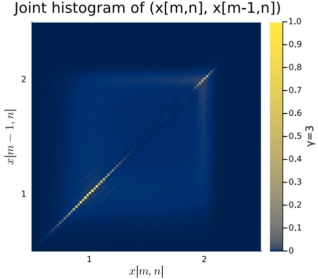
$$p((x[m,n], x[m,n-1]) \mid y = x[m,n] + x[m,n-1])$$

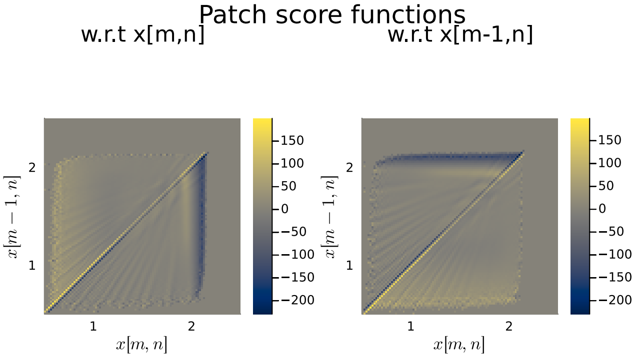
- MRI "center of k-space"
- MRI "2× acceleration





Patch statistics: score functions



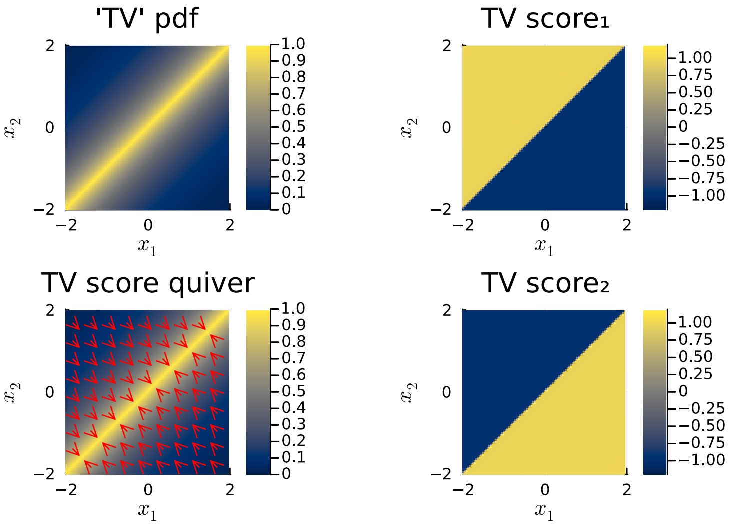


(Manifold data $\Longrightarrow$ score function $\boldsymbol{s}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log \mathrm{p}(\boldsymbol{x})$ is not well-defined.)

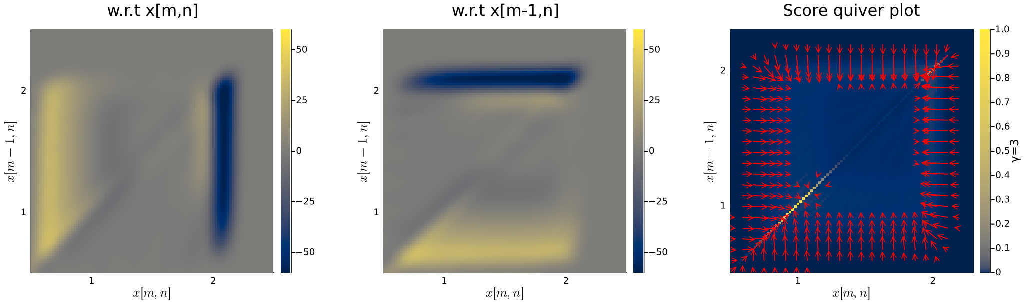
“TV” score function

Total variation (TV) prior for 2×1 patch:

$$p(\boldsymbol{x}) \propto e^{-\beta |x_2 - x_1|}$$

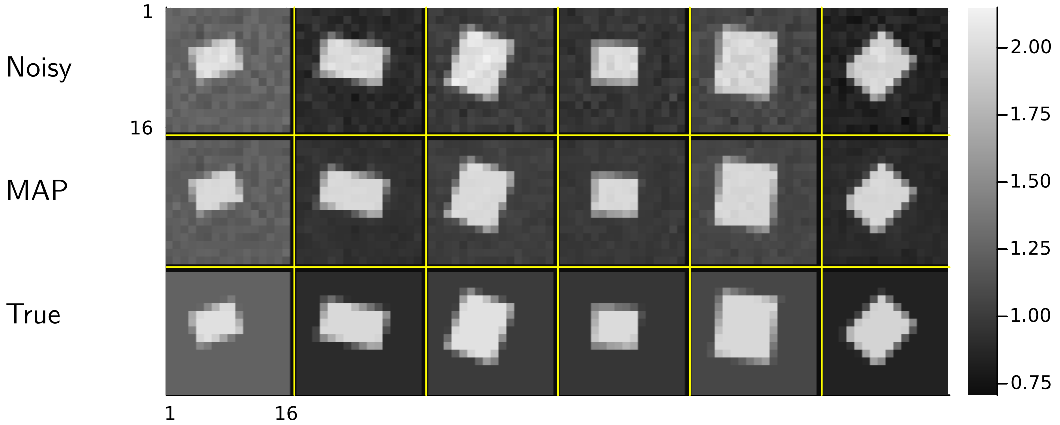


Following trends in score matching [11, 13]
Adding gaussian noise to training data $\equiv$ smoothing score function



MAP denoising via gradient ascent (test images)

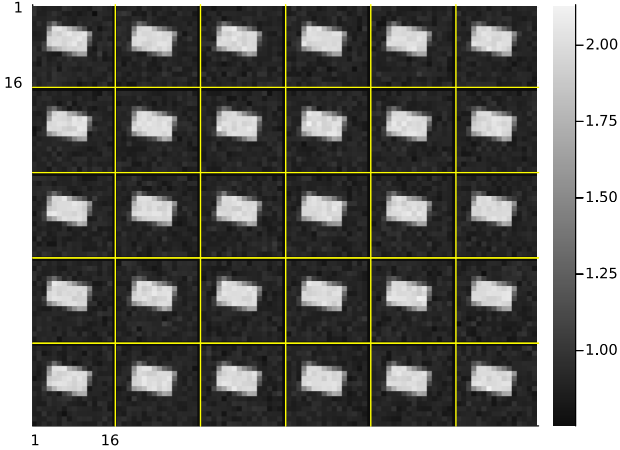
Noisy 29.5dB, MAP 29.9dB, True



Uncertainty?

- Sample from $p(\boldsymbol{x}|\boldsymbol{y})$
- Perform multiple realizations

30 noise realizations



Multiple realizations