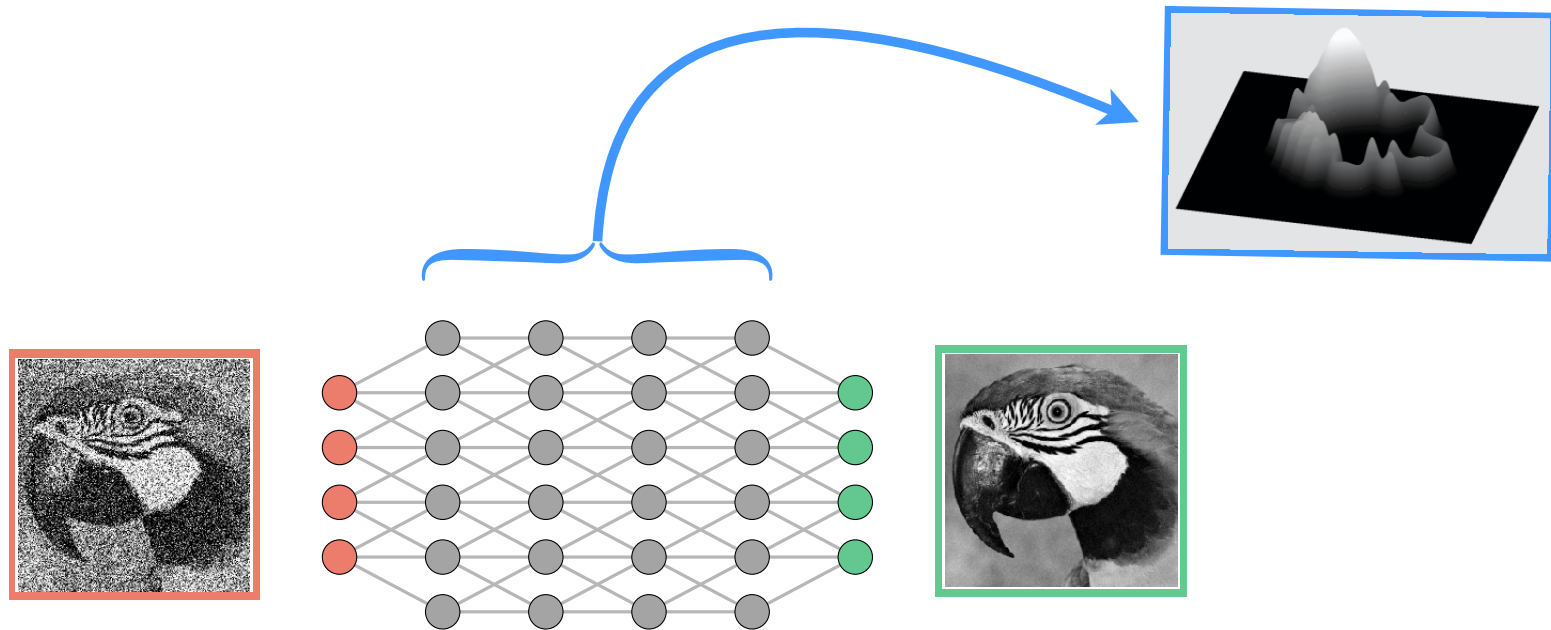
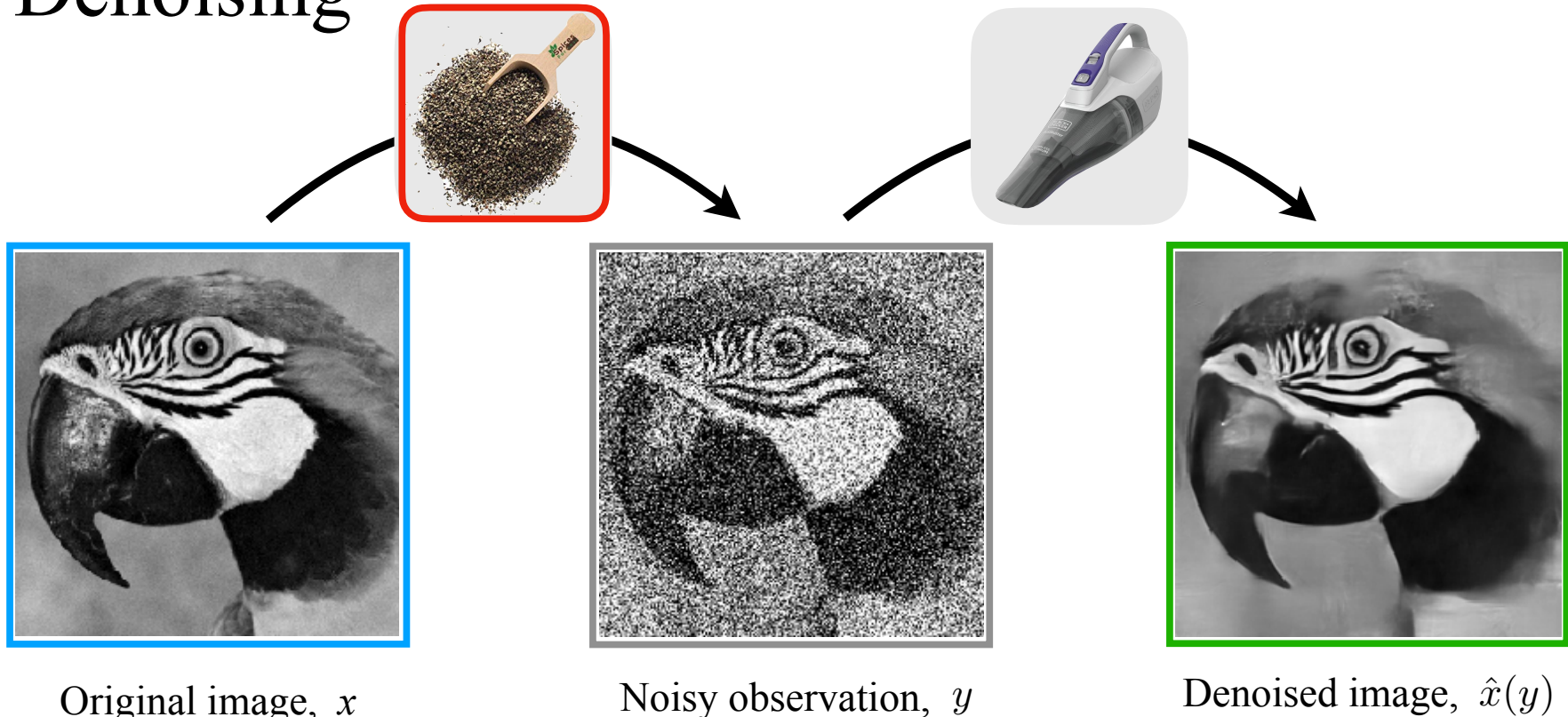


Extracting and using the prior implicitly embedded in a denoiser



Eero Simoncelli,
with Zahra Kadkhodaie, Sreyas Mohan,
and Carlos Fernandez-Granda

Denoising



Traditional Bayesian view:
the least squares estimate is
the posterior mean

$$\begin{aligned}
 \hat{x}(y) &= \min_{\hat{x}} \int \|\hat{x} - x\|^2 p(x|y) dx \\
 &= \int x p(x|y) dx \\
 &= \int x \frac{p(y|x)p(x)}{p(y)} dx
 \end{aligned}$$

Noise model (points to $p(y|x)$)
Signal (prior) model (points to $p(x)$)

Simplest case: additive white Gaussian noise (AWGN)

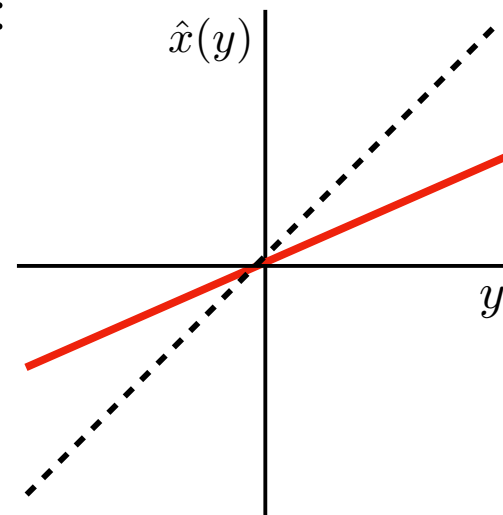
$$y = x + z, \quad z \sim N(0, \sigma_z^2)$$

Note: Prior predictive density is a **blurred** version of the prior:

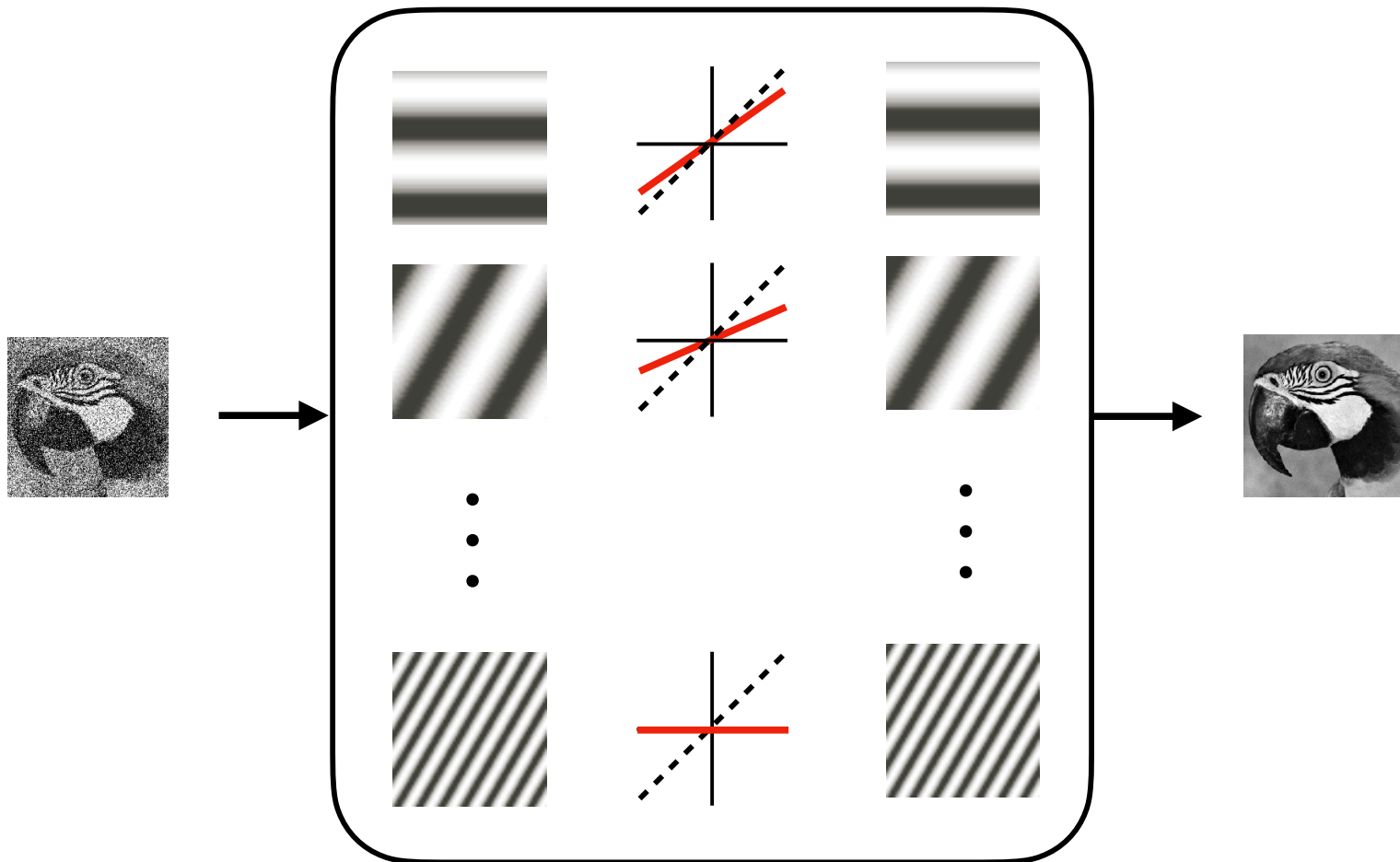
$$p(y) = \int p(y|x) p(x) dx \propto \int e^{-\|y-x\|^2/2\sigma_z^2} p(x) dx$$

Gaussian Prior => Least-squares estimator is linear:

$$\hat{x}(y) = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_z^2} y$$



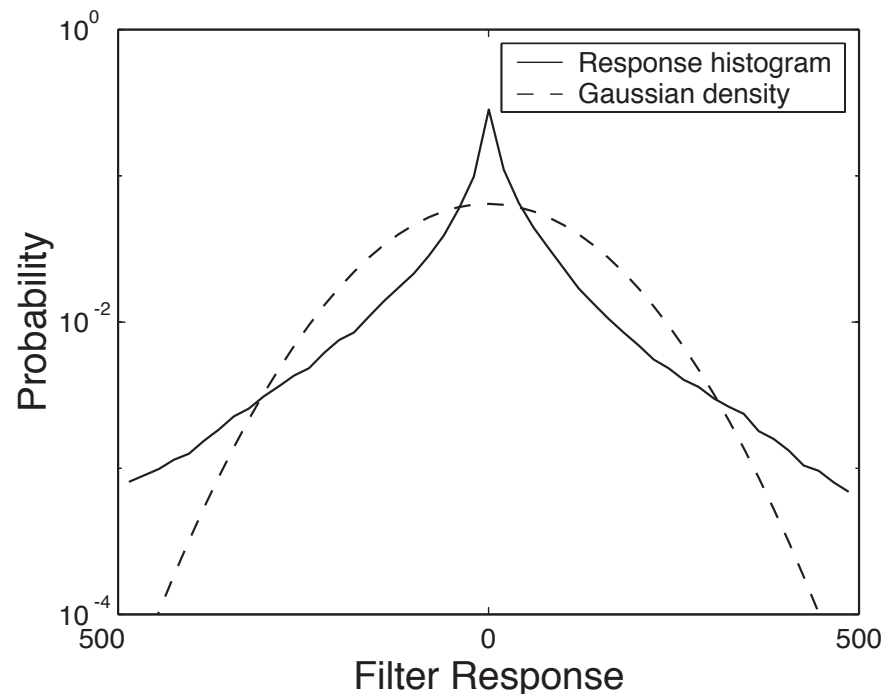
“Wiener filter” (1942)



Lowpass filter $\Rightarrow$ denoised pixels are weighted sum of neighbors

Approximately: projection onto low-frequency subspace

For photographic images (and many other natural signals),
bandpass filter responses are *not* Gaussian



[Burt&Adelson 82; Field 87; Mallat 89; etc]

Simplest case: additive white Gaussian noise (AWGN)

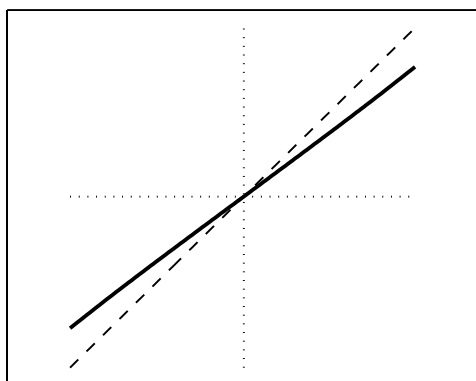
$$y = x + z, \quad z \sim N(0, \sigma_z^2)$$

Prior predictive density is a **blurred** version of the prior:

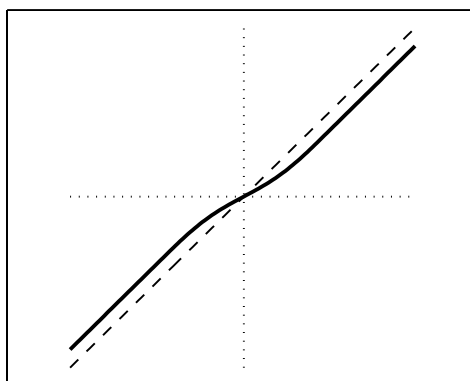
$$p(y) = \int p(y|x) p(x) dx \propto \int e^{-\|y-x\|^2/2\sigma_z^2} p(x) dx$$

Sparse/heavy-tailed prior $\Rightarrow$ least-squares estimator is **nonlinear**:

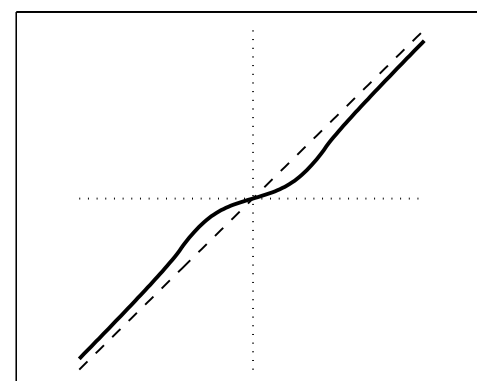
Example: $p(x) \propto \exp(-|x/s|^\beta)$



$$\beta = 2.0$$



$$\beta = 1.0$$



$$\beta = 0.5$$

[Simoncelli & Adelson 96]

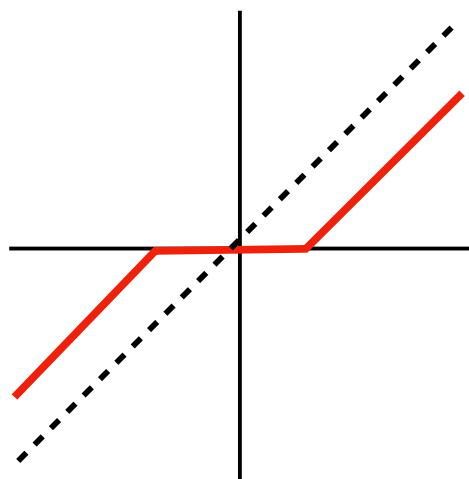
Simplest case: additive white Gaussian noise (AWGN)

$$y = x + z, \quad z \sim N(0, \sigma_z^2)$$

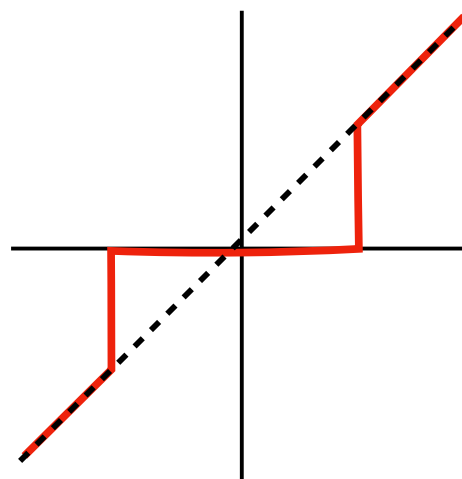
Prior predictive density is a **blurred** version of the prior:

$$p(y) = \int p(y|x) p(x) dx \propto \int e^{-\|y-x\|^2/2\sigma_z^2} p(x) dx$$

Sparse/heavy-tailed prior $\Rightarrow$ least-squares estimator is **nonlinear**:



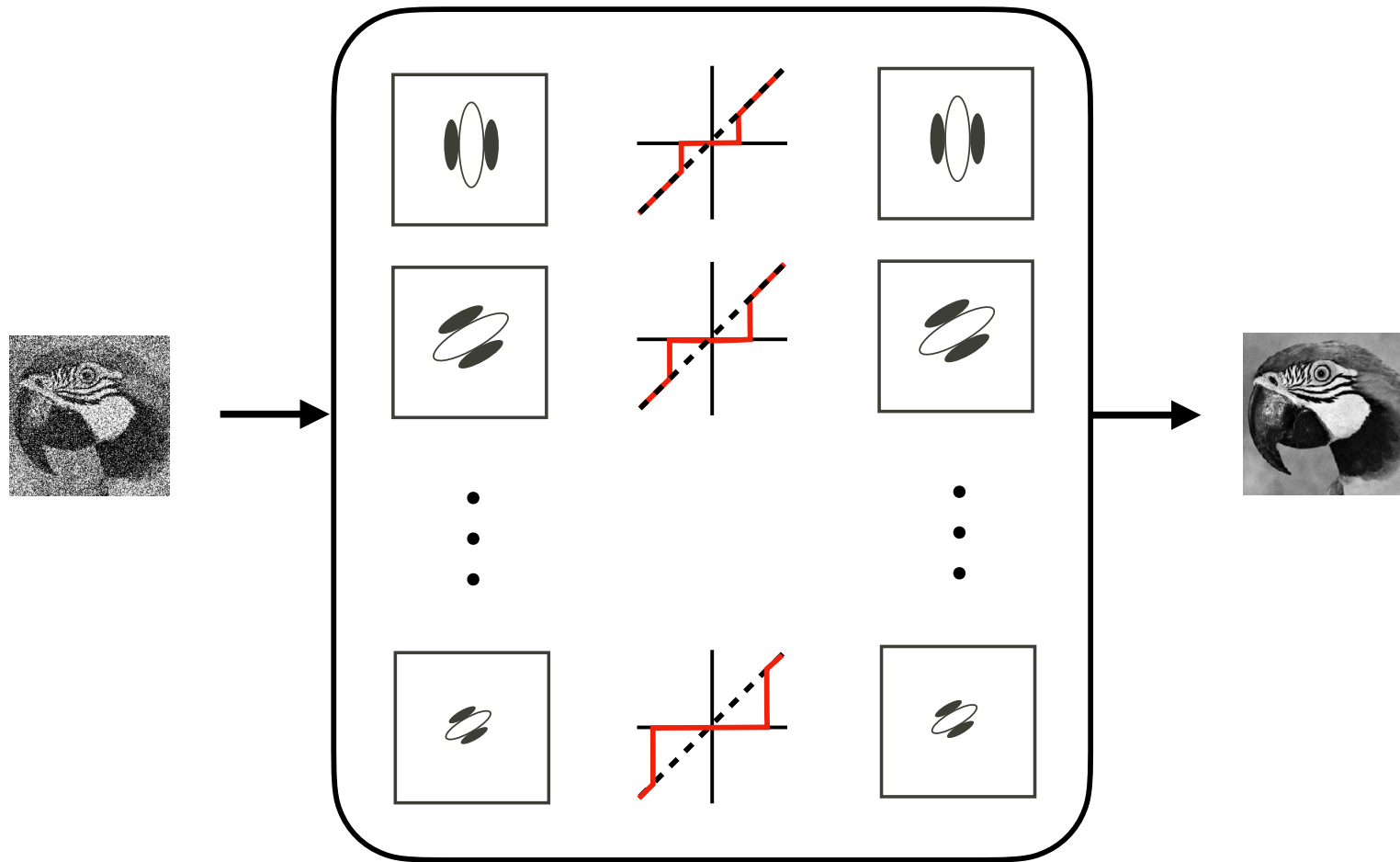
soft threshold



hard threshold

[Donoho & Johnstone 94]

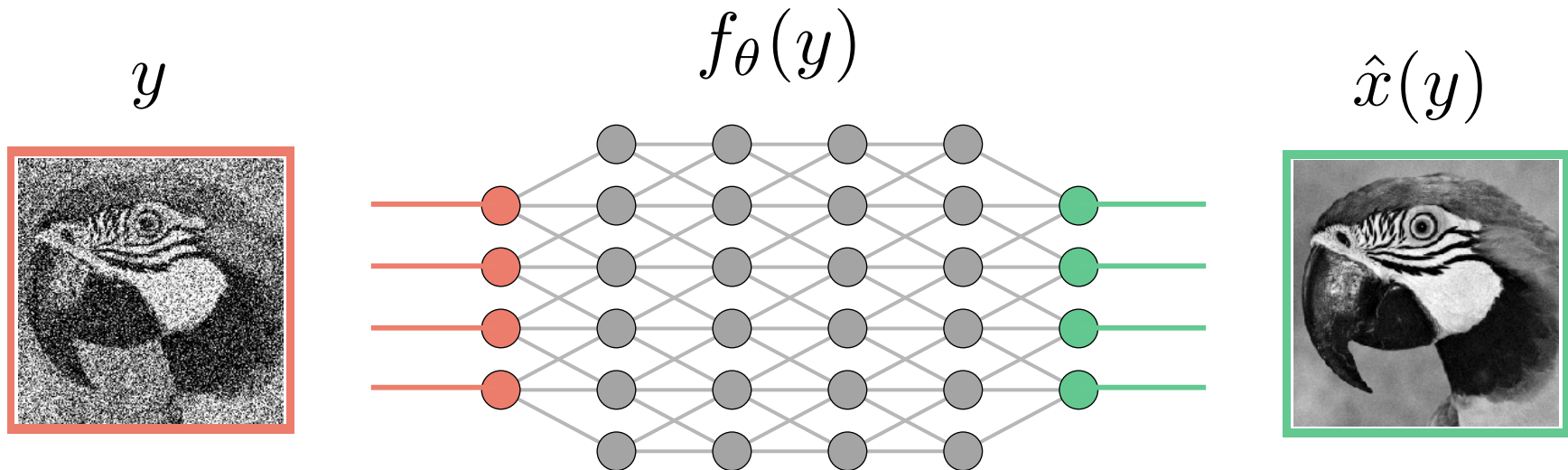
“Wavelet shrinkage”



Projection onto an *adaptive* subspace

[Donoho & Johnstone 94]

Denoising with CNNs



Example: DnCNN (Zhang et al., 2017)

20 layers, with 64 channels, each using a 3x3 filter, batchNorm, ReLU

~700,000 parameters (convolution kernels + additive constants)

Trained on $\{y, x\}$ pairs. Images x from Berkeley image dataset.

State-of-the-art performance (2017).

Given trained denoising CNN, $f_{\hat{\theta}}(y)$

Analysis: properties of learned mapping

[Mohan et al., ICLR 2020]

Interpretation: what is the implicit “prior”?

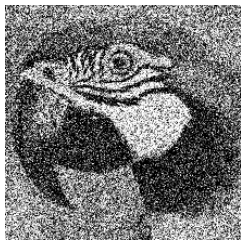
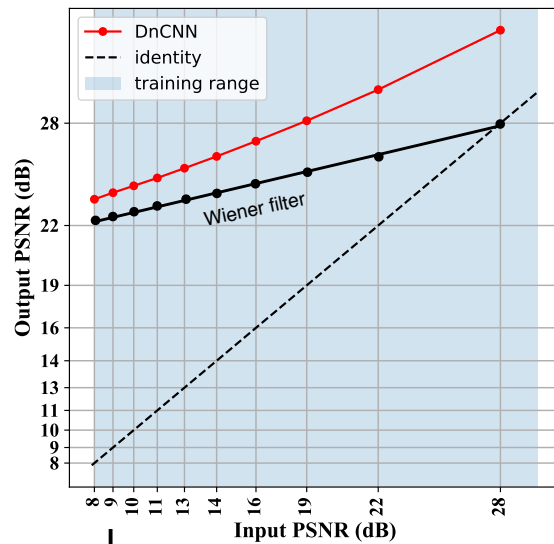
Generalization: Can we use this prior to solve other problems?

[Kadkhodaie & Simoncelli, ArXiv 2020]

Analysis: Generalization to other noise levels

Universal blind denoising ...

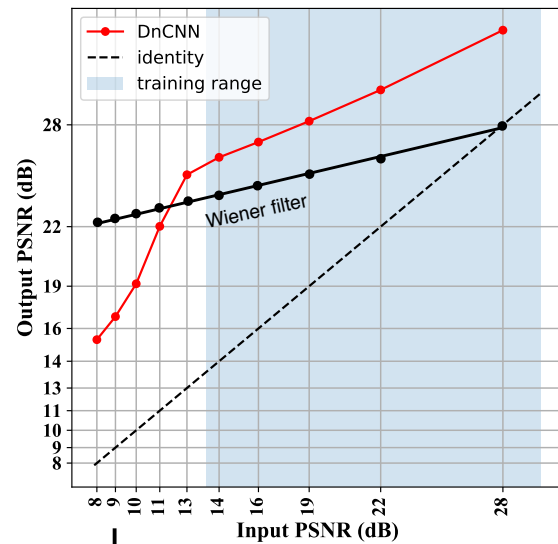
... but, doesn't generalize beyond training range



Noisy test image



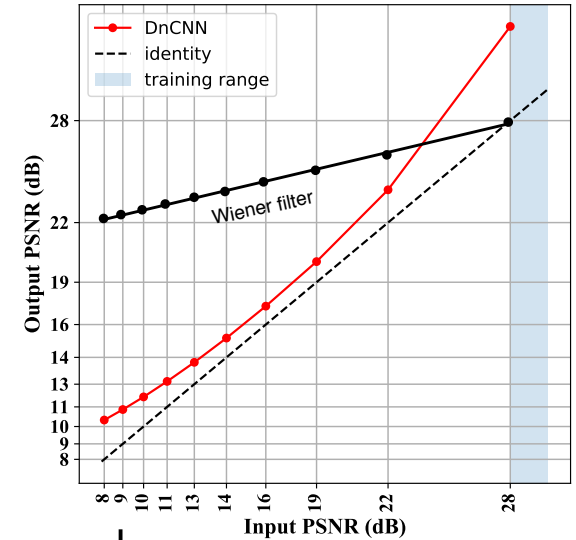
Denoised



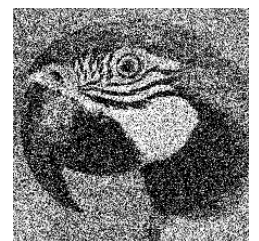
Noisy test image



Denoised



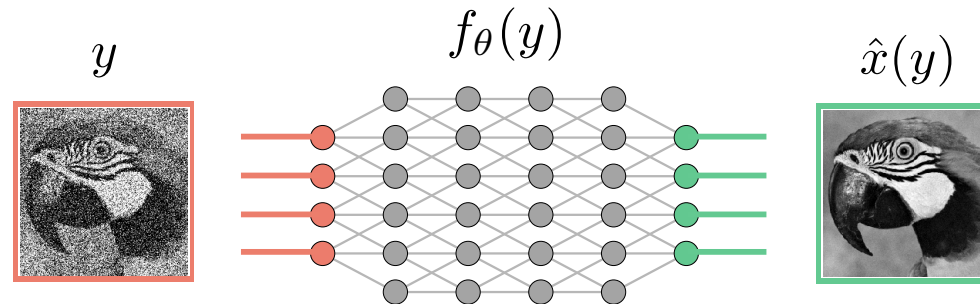
Noisy test image



Denoised

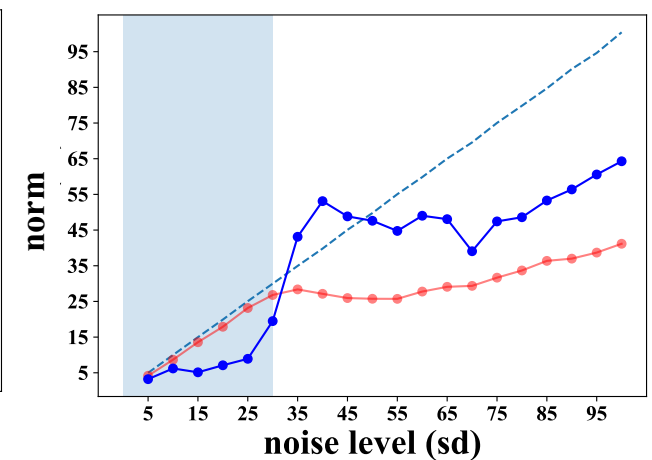
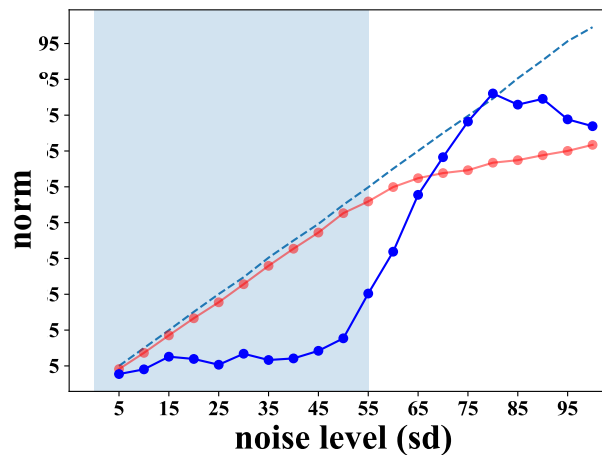
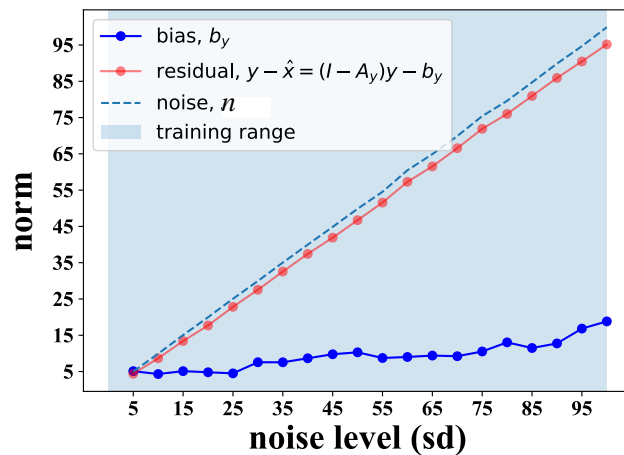
[Mohan et al., ICLR 2020]

Analysis: Adaptive affine transformation



Denoiser is a cascade of (adaptive) affine transformations:

$$f(y) = R(W_1 y + b_1)$$



Network bias impedes generalization

Removing bias enables generalization



Noisy training image,
 $\sigma = 10$ (max level)



Noisy test image,
 $\sigma = 90$

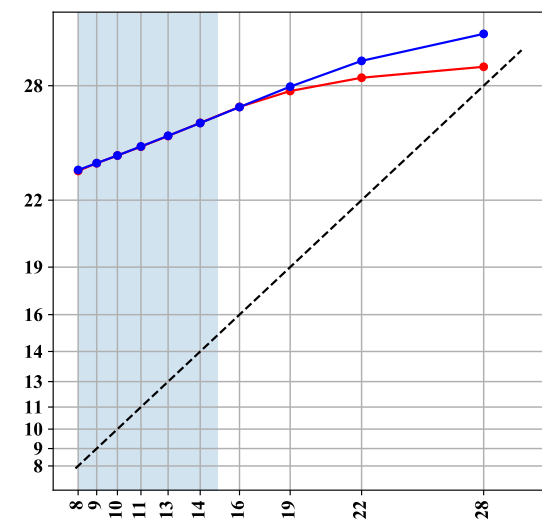
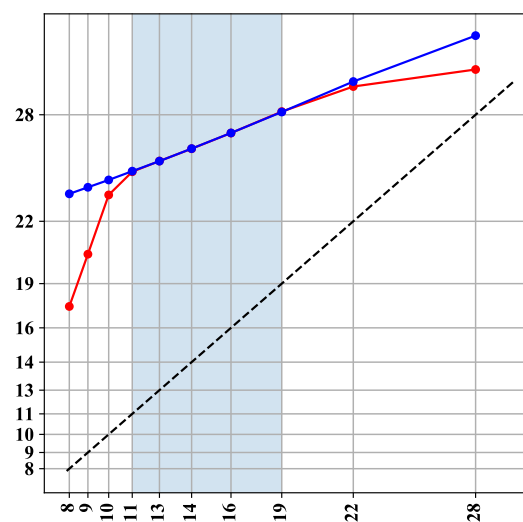
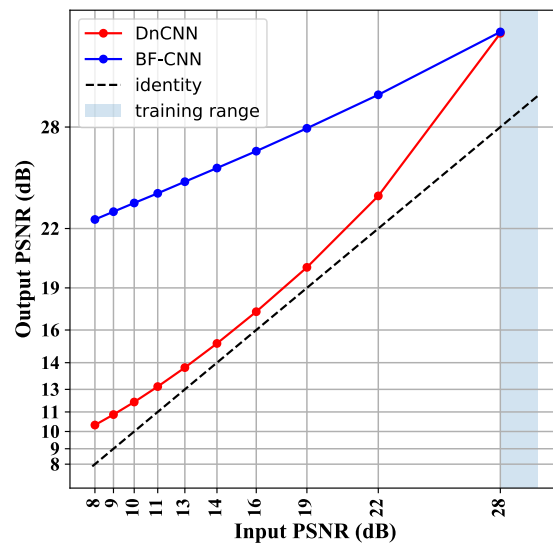


Test image, denoised
by DnCNN



Test image, denoised
by BF-CNN

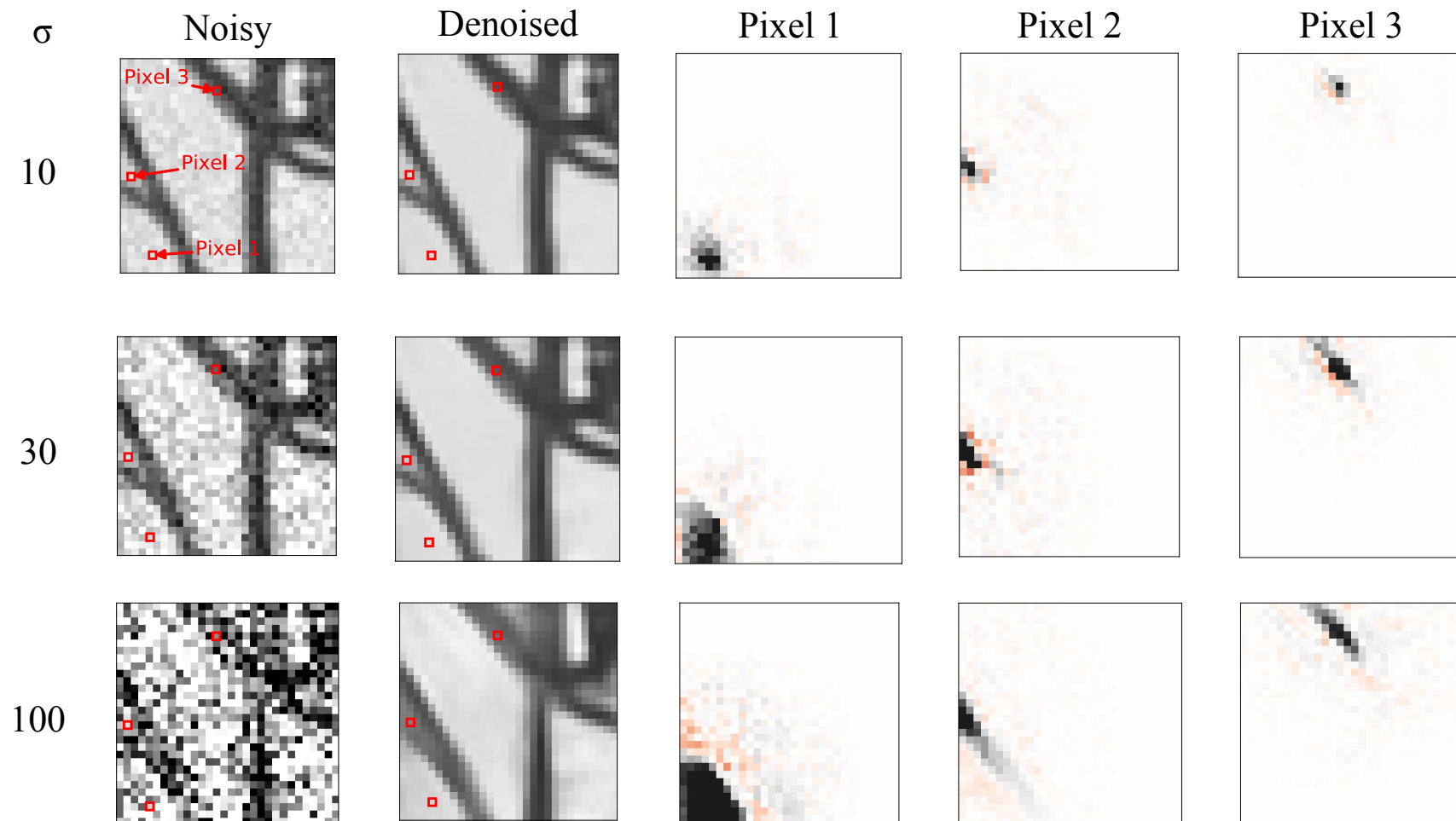
Bias- free CNN (BF-CNN): $f_{\text{BF}}(y) = W_L R(W_{L-1} \dots R(W_1 y)) = A_y y$,



Removing bias also allows interpretation: Adaptive filtering

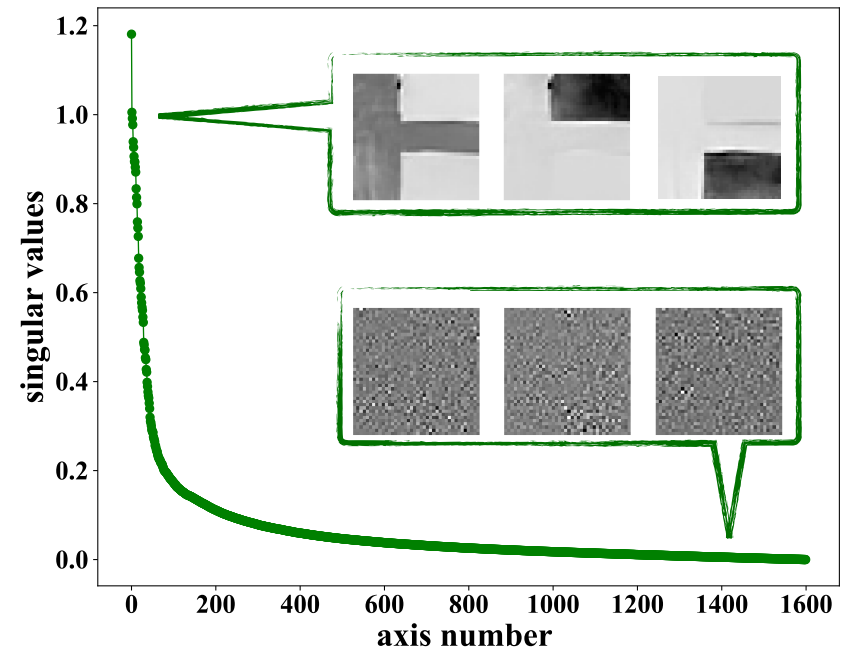
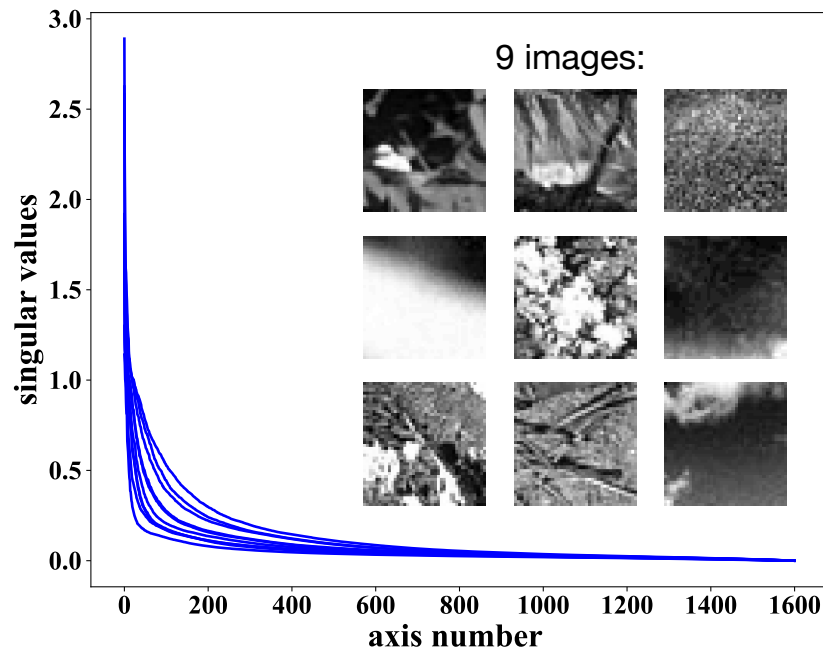
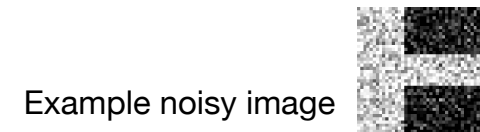
$$f_{BF}(y) = A_y y$$

Weighting functions:



Interpretation: Projection onto adaptive signal subspace

$$f_{\text{BF}}(y) = A_y y = U S V^T y$$



ToDo

Denoising is achieved with cascades of convolution and rectification.
How do this achieve such highly adaptive shrinkage?

Can we improve denoising (and embedded prior) with a modified architecture?

How can we incorporate desired properties of invariance or symmetry?

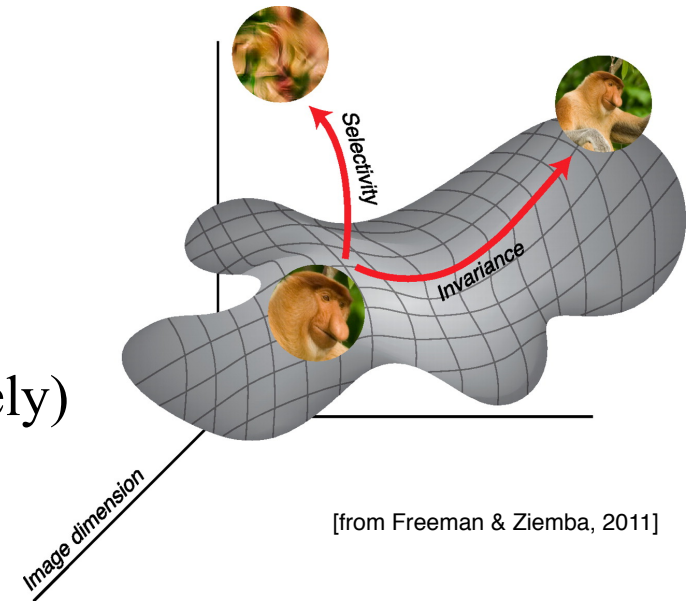
Robust and interpretable blind image denoising via bias-free convolutional neural networks.

S Mohan, Z Kadkhodaie, EP Simoncelli, C Fernandez-Granda.
Int'l Conf Learning Representations (ICLR) 2020.

Software implementation, including 4 different BF CNNs:
https://github.com/LabForComputationalVision/bias_free_denoising

Manifold prior hypothesis

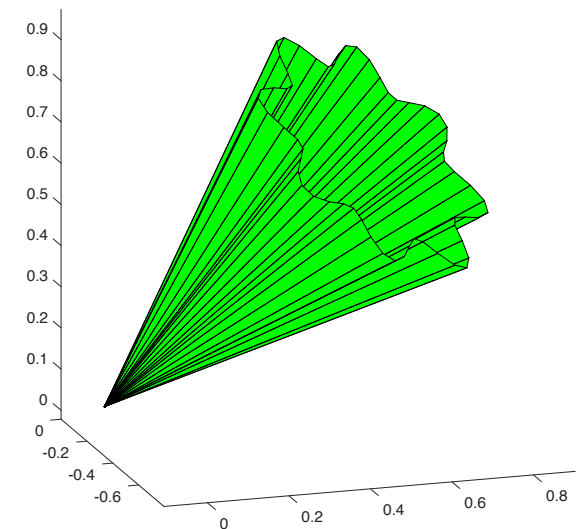
- Visual images lie on a low-dimensional surface (manifold), spanned by various natural deformations
- Images on this manifold are (approximately) equally probable - at least locally

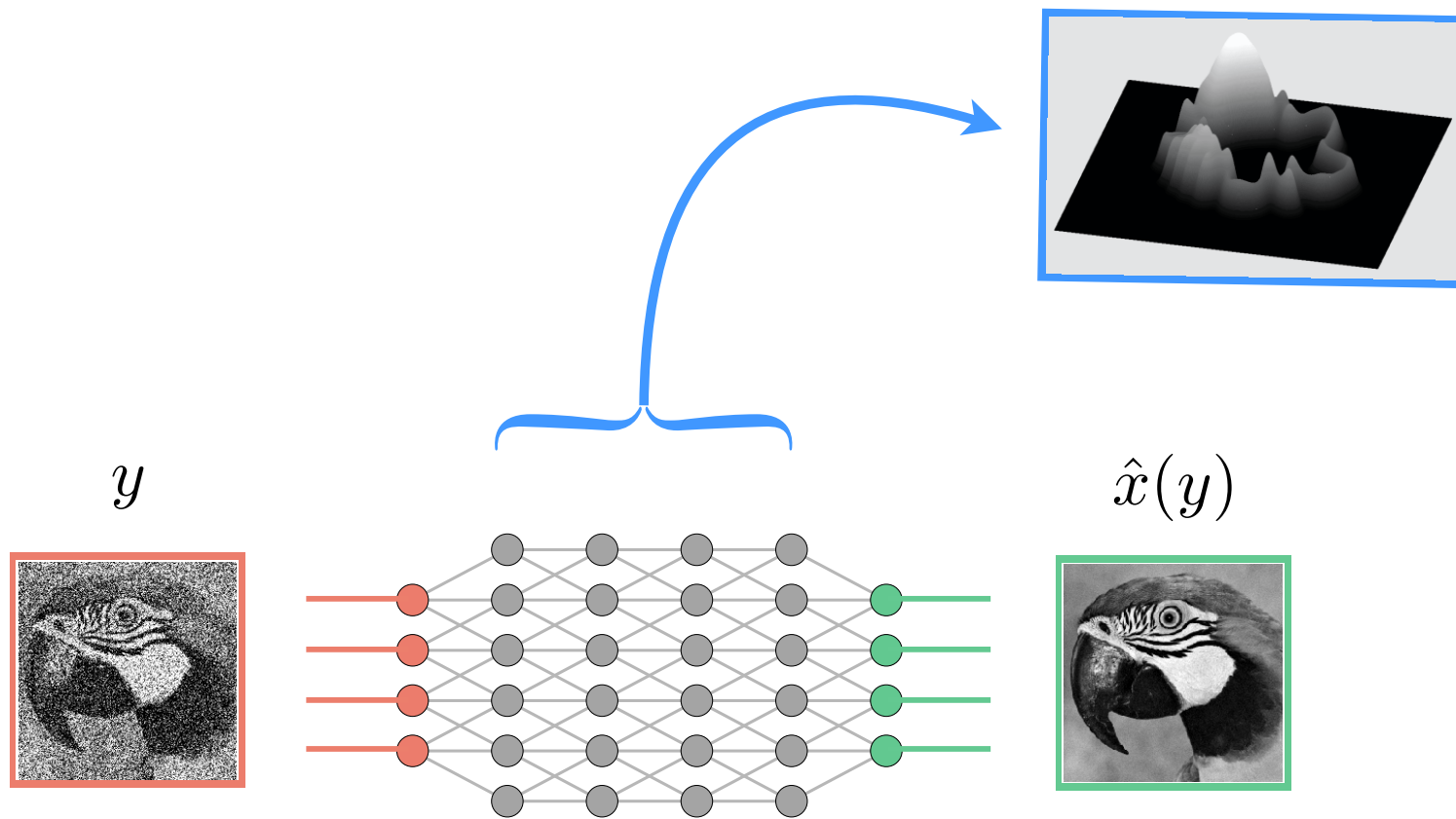


BF-CNN analysis suggests

- Local manifold tangents are proper subspaces
- Local subspace determined (adaptively) by binary settings of ReLUs
- Subspaces are continuously connected

Geometry: a generalized cone





Trained DNN denoisers contain an (implicit) image prior...

Can we use this prior to solve other inverse problems?

[Kadkhodaie and Simoncelli, arXiv, Jul 2020]

Miyasawa's Empirical Bayes formulation (1961)

BLS estimator: $\hat{x}(y) = \int x p(x|y) dx$

Prior predictive density (assuming AWGN), is blurred prior:

$$p(y) = \int p(y|x) p(x) dx \propto \int e^{-\|y-x\|^2/2\sigma^2} p(x) dx$$

$$\nabla_y p(y) = \frac{1}{\sigma^2} \int (x - y) p(y|x) p(x) dx = \frac{1}{\sigma^2} \int (x - y) p(y, x) dx$$

$$\sigma^2 \frac{\nabla_y p(y)}{p(y)} = \int x p(x|y) dx - \int y p(x|y) dx = \hat{x}(y) - y.$$

$$\Rightarrow \hat{x}(y) = y + \sigma^2 \nabla_y \log p(y)$$

- Equivalent to the BLS estimator (no approximations)
- Suggestive of gradient ascent, but non-iterative
- Prior is implicit (in blurred form) in the prior predictive density

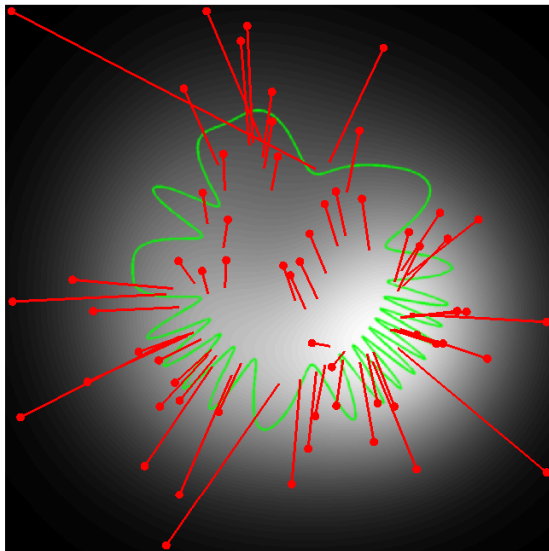
[see Raphan & Simoncelli (2011) for generalization to many other noise processes]

BLS solution, with a manifold prior

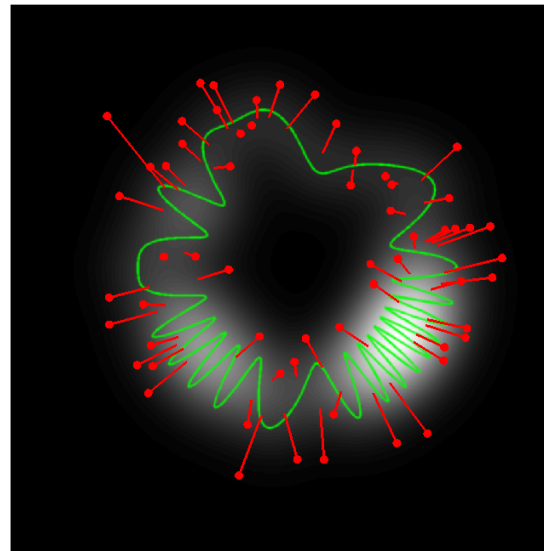
$$\hat{x}(y) = y + \sigma^2 \nabla_y \log p(y)$$

[Miyasawa, 1961]

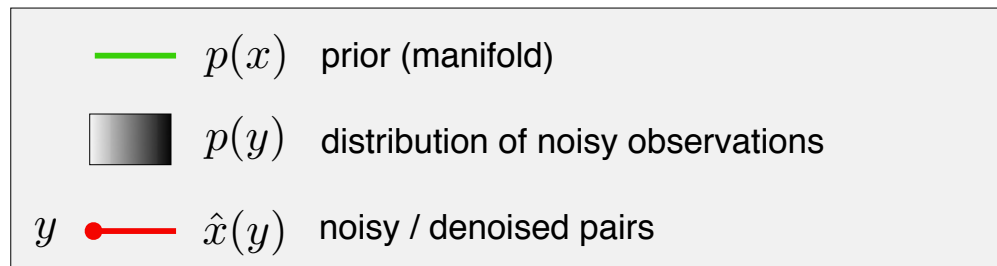
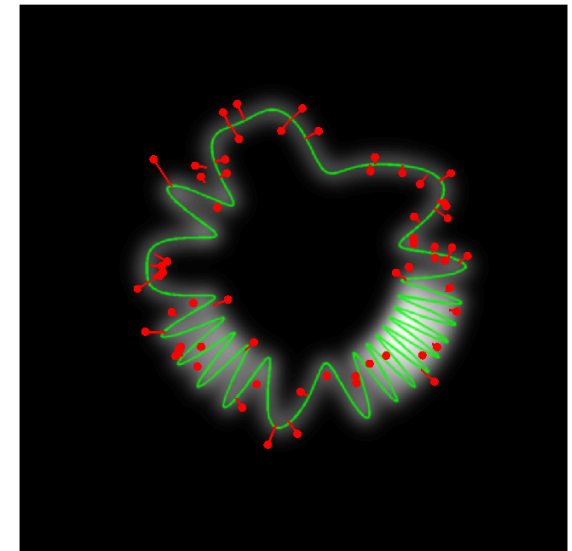
large σ



medium σ



small σ



Sampling from the implicit prior...

- Use denoiser-defined gradient to iteratively ascend probability
- On each step, effective “noise” decreases, and effective prior is less blurred
- This “coarse-to-fine” optimization procedure converges to a point on the manifold

Algorithm 0: Coarse-to-fine gradient ascent method for sampling from the implicit prior of a denoiser, using denoiser residual $f(y) = \hat{x}(y) - y$.

parameters: σ_0, σ_L, h

initialization: $t = 1$, draw $y_0 \sim \mathcal{N}(0.5, \sigma_0^2 I)$

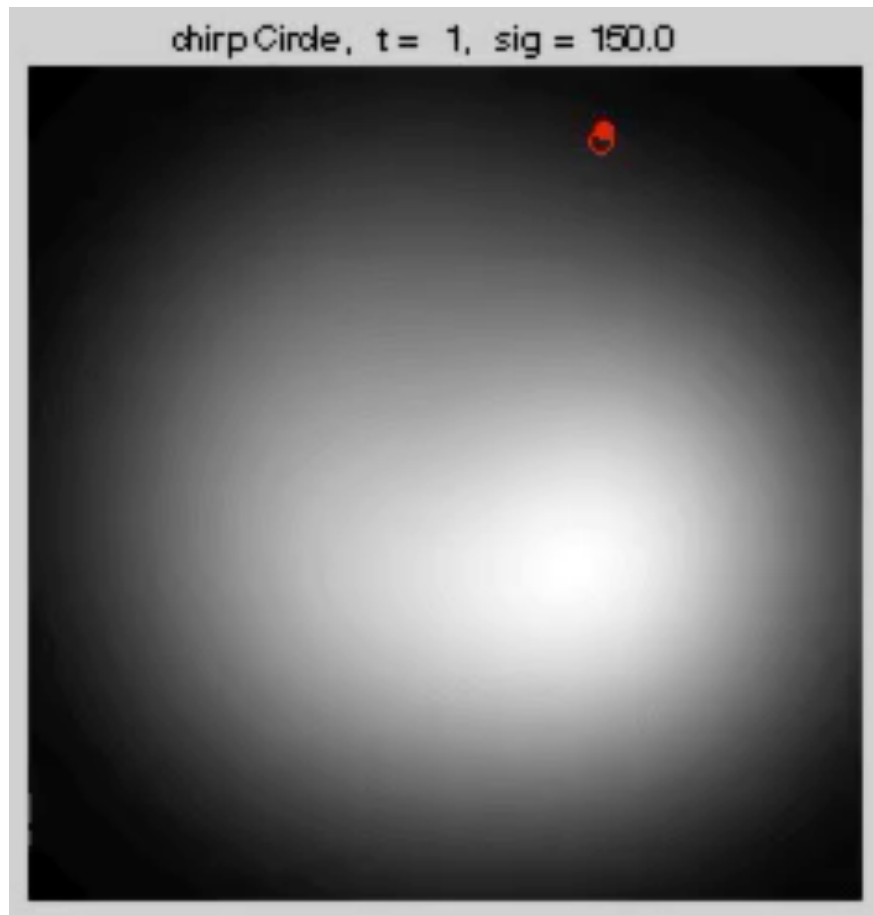
while $\sigma_{t-1} \leq \sigma_L$ **do**

$d_t = f(y_{t-1});$
 $y_t \leftarrow y_{t-1} + h d_t;$
 $t \leftarrow t + 1$

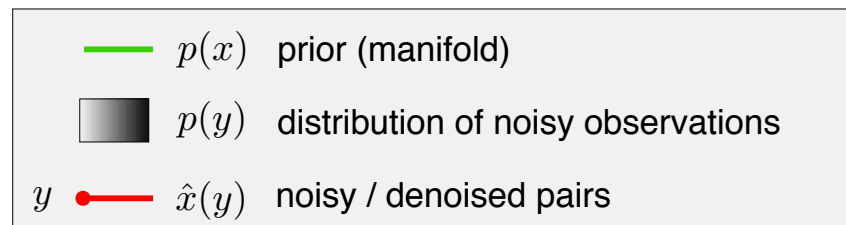
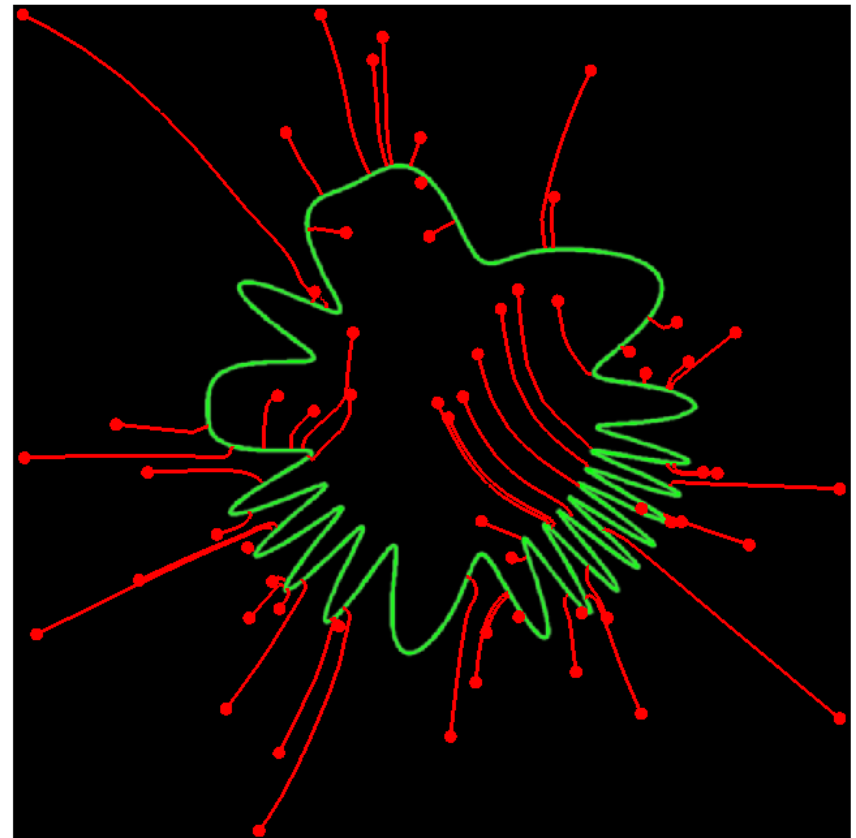
end

[related: Bengio et al 2013; Romano, Elad, Milanfar 2017; Bigdeli, Jin, Zwicker 2017; Saremi & Hyvarinen 2019; Song & Ermon 2019]

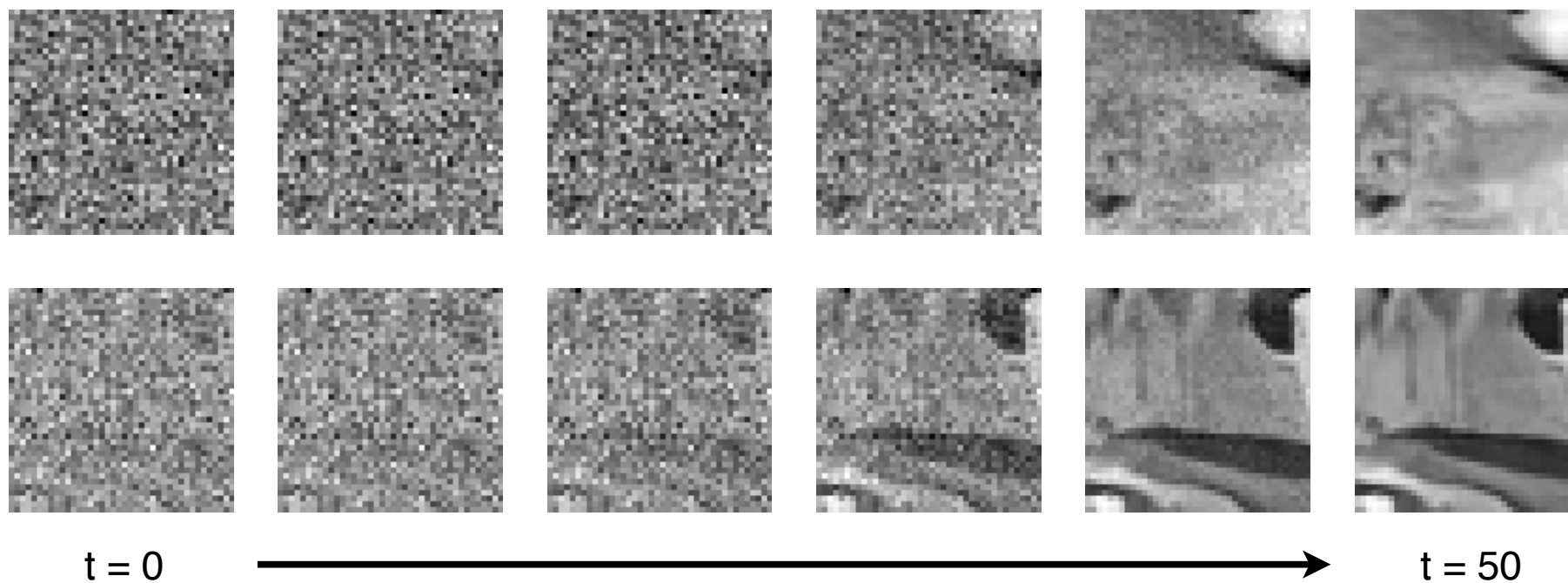
Iterative coarse-to-fine sampling



Example trajectories



Two example image synthesis sequences (different random initializations)



Enhancement 1: Inject noise on each iteration, to achieve higher local maxima

Algorithm 1: Coarse-to-fine stochastic ascent method for sampling from the implicit prior of a denoiser, using denoiser residual $f(y) = \hat{x}(y) - y$.

parameters: $\sigma_0, \sigma_L, h_0, \beta$

initialization: $t = 1$, draw $y_0 \sim \mathcal{N}(0.5, \sigma_0^2 I)$

while $\sigma_{t-1} \leq \sigma_L$ **do**

$$h_t = \frac{h_0 t}{1 + h_0(t-1)};$$

$$d_t = f(y_{t-1});$$

$$\sigma_t^2 = \frac{\|d_t\|^2}{N};$$

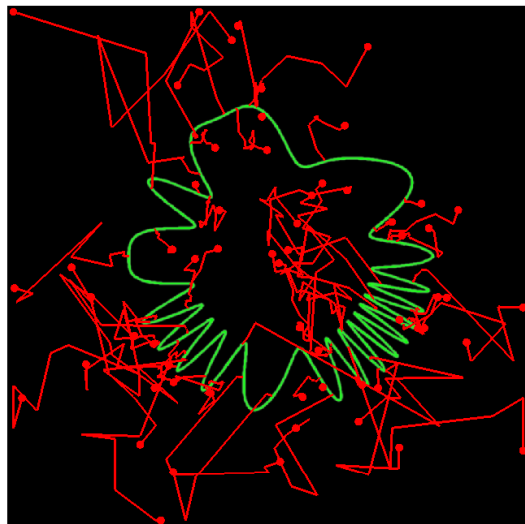
$$\gamma_t^2 = ((1 - \beta h_t)^2 - (1 - h_t)^2) \sigma_t^2;$$

Draw $z_t \sim \mathcal{N}(0, I)$;

$$y_t \leftarrow y_{t-1} + h_t d_t + \gamma_t z_t;$$

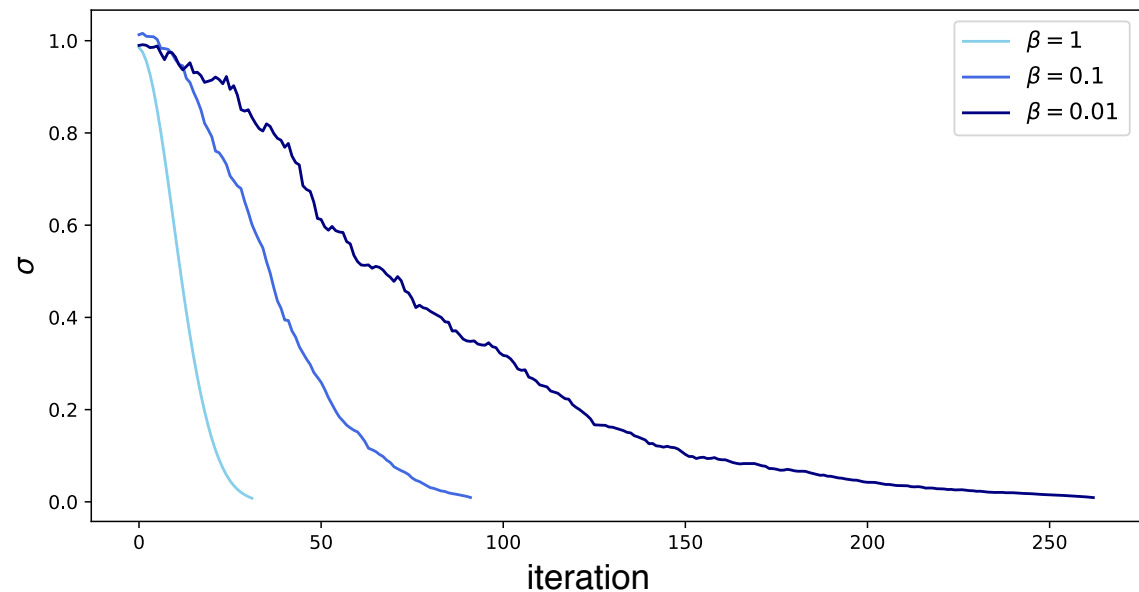
$$t \leftarrow t + 1$$

end



lots of noise ($\beta = 0.01$)

Convergence:



Synthesized images, different initializations

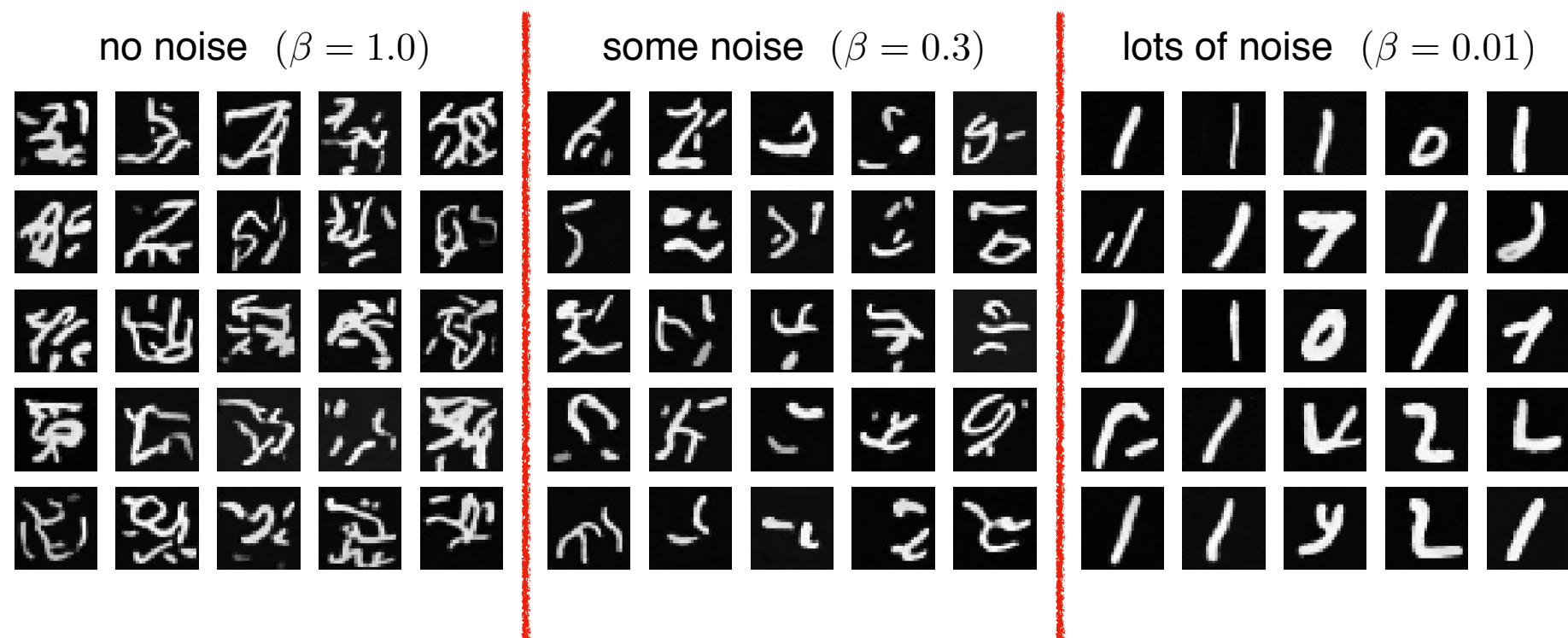
Moderate injected noise ($\beta = 0.5$)



Substantial injected noise ($\beta = 0.1$)



Samples from denoiser trained on the MNIST dataset (handwritten digits)



Enhancement 2: Incorporate a linear constraint

$$M^T x = x^c, \quad \text{with } M \text{ a low-rank matrix}$$

Sample from conditional $p(y|x^c)$ instead of $p(y) \dots$

Algorithm 2: Coarse-to-fine stochastic ascent method for sampling from $p(x|M^T x = x^c)$, based on the residual of a denoiser, $f(y) = \hat{x}(y) - y$. Note: e is an image of ones.

parameters: $\sigma_0, \sigma_L, h_0, \beta, M, x^c$

initialization: $t=1$; draw $y_0 \sim \mathcal{N}(0.5(I - MM^T)e + Mx^c, \sigma_0^2 I)$

while $\sigma_{t-1} \leq \sigma_L$ **do**

$$h_t = \frac{h_0 t}{1 + h_0(t-1)};$$

$$d_t = (I - MM^T)f(y_{t-1}) + (Mx^c - MM^T y_{t-1});$$

$$\sigma_t^2 = \frac{\|d_t\|^2}{N};$$

$$\gamma_t^2 = ((1 - \beta h_t)^2 - (1 - h_t)^2) \sigma_t^2;$$

Draw $z_t \sim \mathcal{N}(0, I)$;

$$y_t \leftarrow y_{t-1} + h_t d_t + \gamma_t z_t;$$

$$t \leftarrow t + 1$$

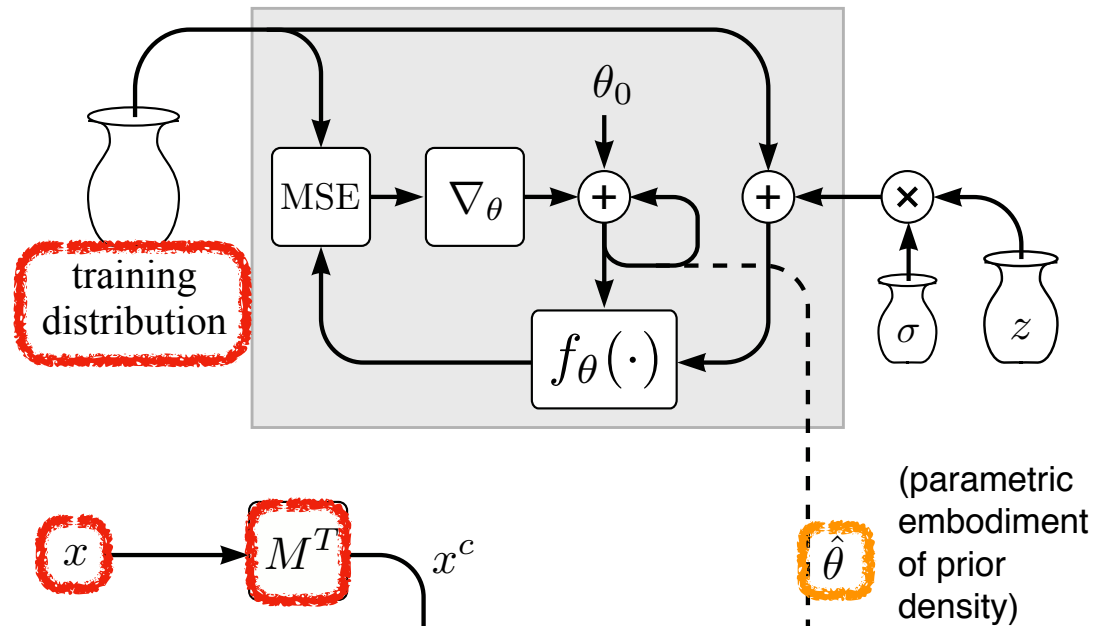
end

We can use this to solve *any* linear inverse problem!

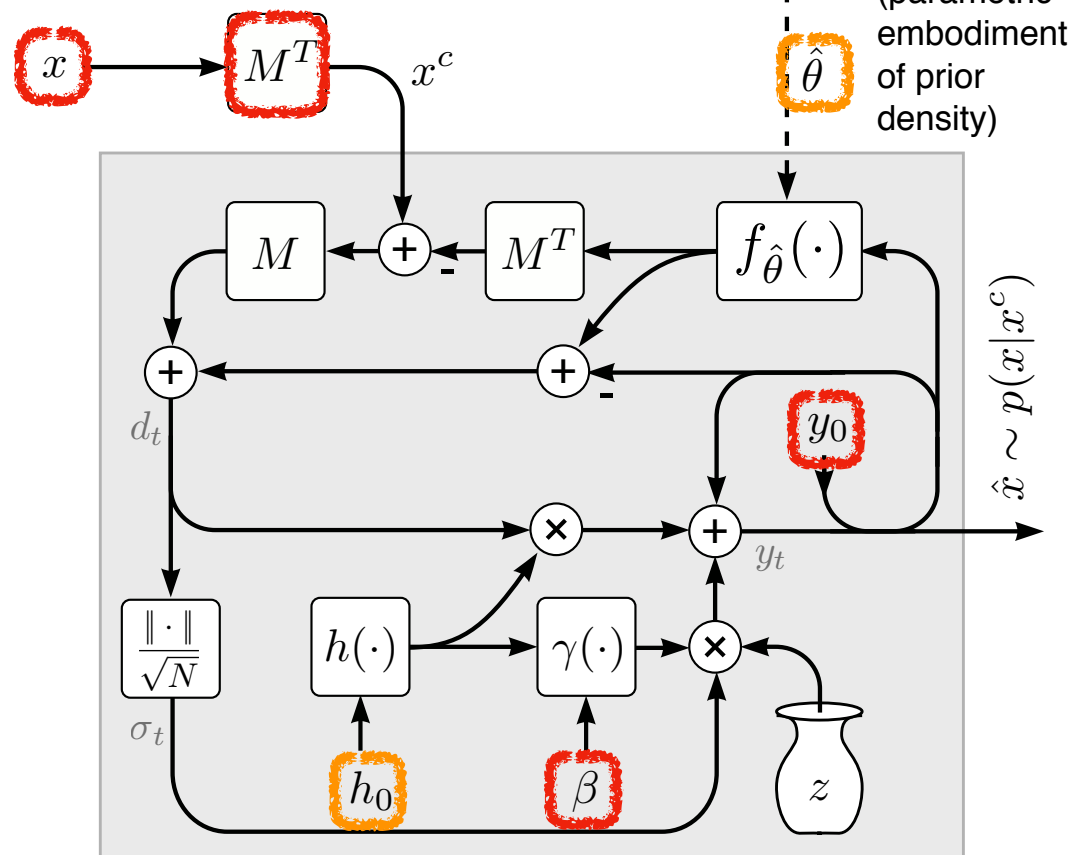
Note: no further training, or reliance on Gaussian noise assumption

Linear Inverse “Machine”

(1) Train denoiser, $f_{\theta}(\cdot)$

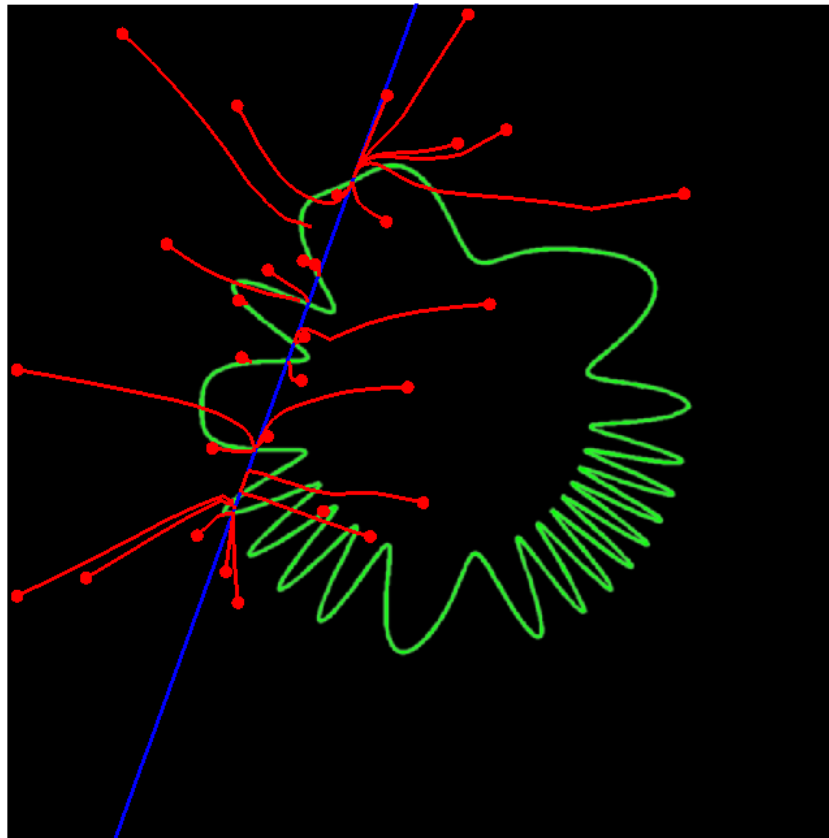


(2) Use trained denoiser to solve *any* linear inverse problem



Iterative coarse-to-fine sampler, with linear constraint

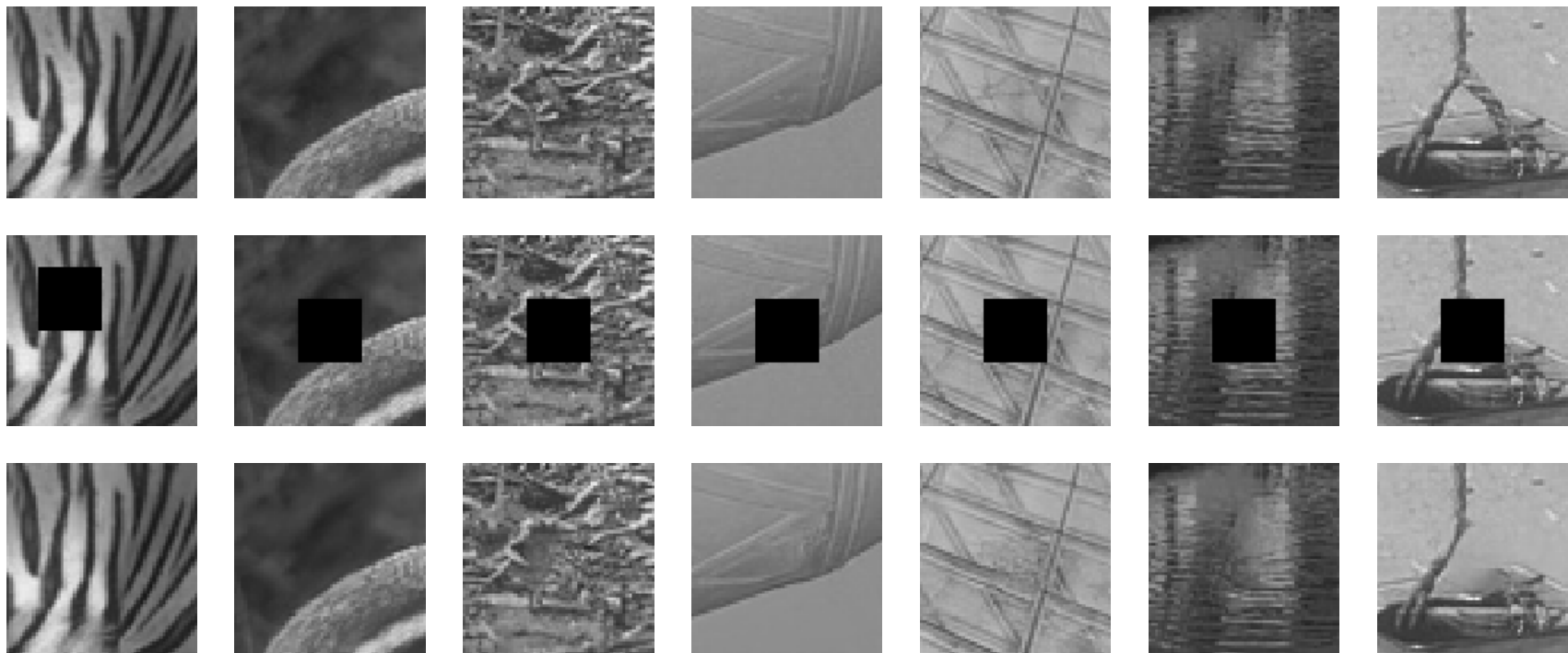
Example trajectories



Linear inverse example 1: “Inpainting” (fill in a hole)



Additional examples (Top: original, Middle: missing square, Bottom: restored)



Linear inverse example 2: Random pixel dropping

30% dimensionality (70% dropped)



10% dimensionality (90% dropped)



Top: original. Middle: pixel measurement. Bottom: recovered using our method.

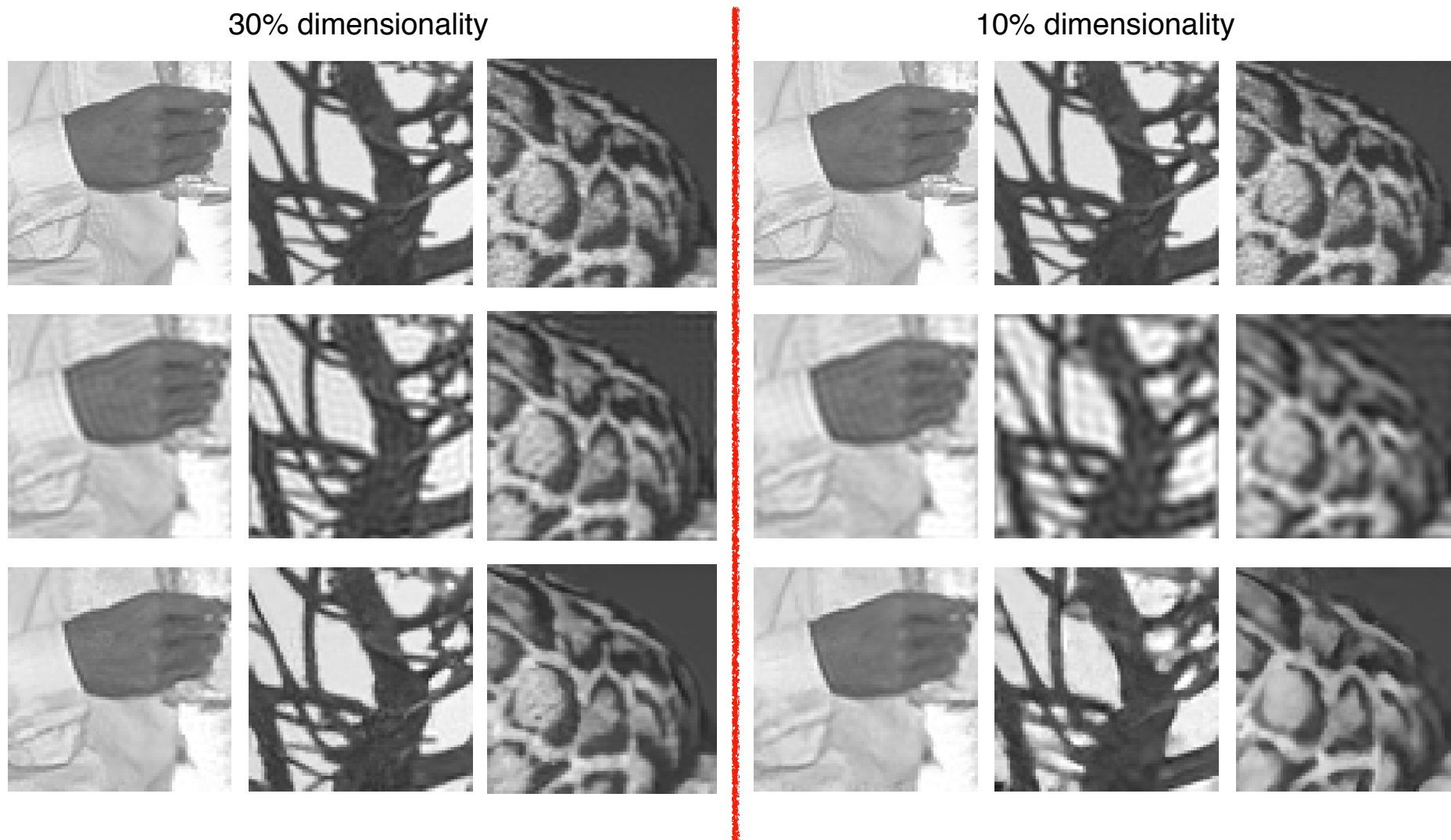
Linear inverse example 3: Super-resolution (4x4 block averaging)

6.25% dimensionality (1/16)



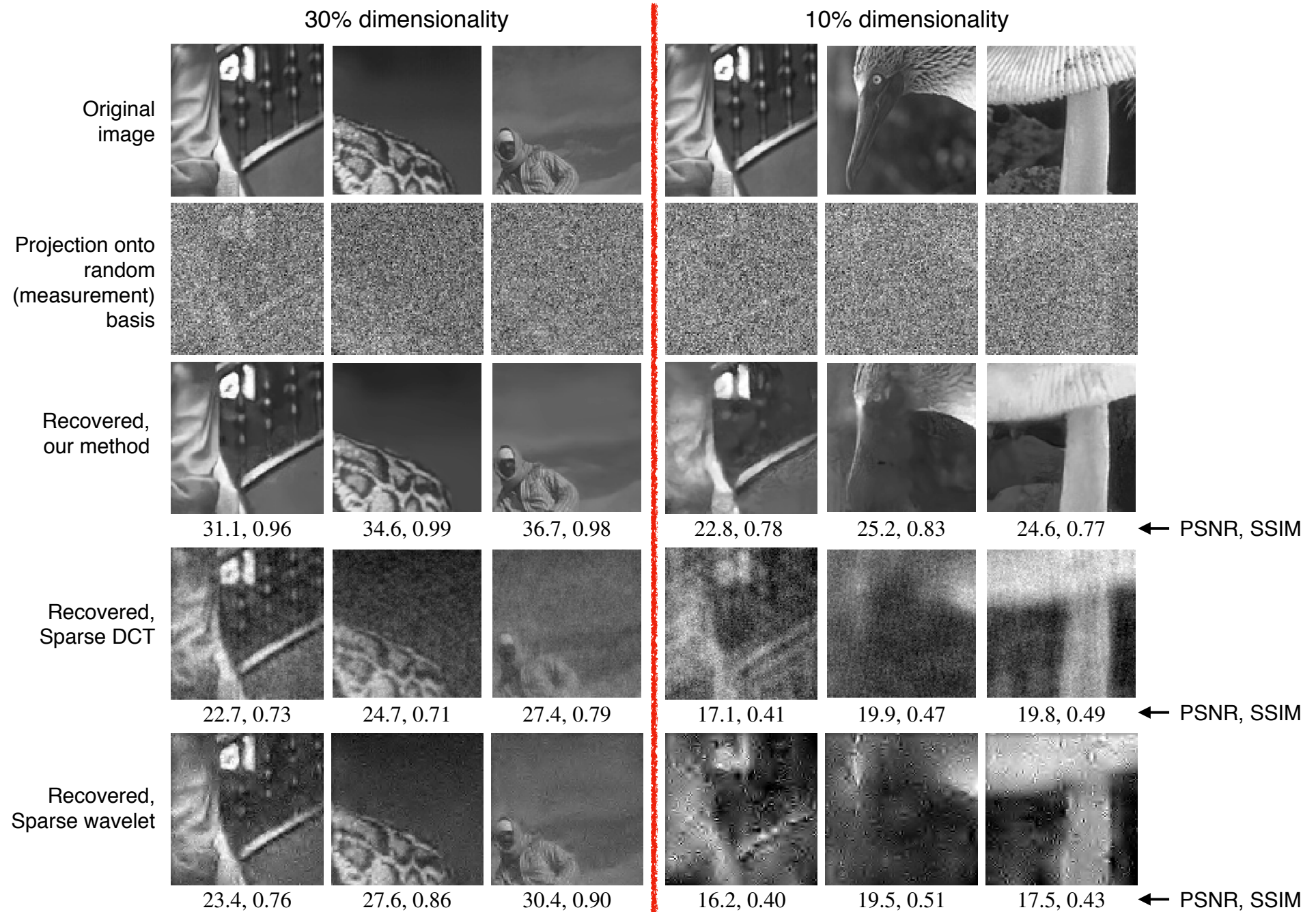
Top: original. Middle: block measurements. Bottom: recovered using our method.

Linear inverse example 4: Blurring (discard high frequencies)



Top: original. Middle: low-frequency measurement. Bottom: recovered using our method.

Linear inverse example 5: Random projections



ToDo

Generalization to other inverse problems:

- Nonlinear measurements (quantization, texture statistics)
- Noisy measurements (MAP estimates)

Can a local denoising objective be used for general unsupervised learning of priors? In biological systems?

Solving linear inverse problems using the prior implicit in a denoiser

Z Kadkhodaie and E P Simoncelli. arXiv.org 2007.13640, Jul 2020.

To appear: Adv. Neural Information Processing Systems (NeurIPS*20),
Workshop on Deep Learning and Inverse Problems, Dec 2020.



Zahra Kadkhodaie



Sreyas Mohan



Carlos
Fernandez-Granda

hhmi

Howard Hughes
Medical Institute

SIMONS
FOUNDATION



National
Eye
Institute

NATIONAL INSTITUTES OF HEALTH