Modern AI for Inverse Problems (Data Driven Inverse Problems)

Mahdi Soltanolkotabi
USC Center on AI Foundations for the Sciences (AIF4S)
ECE, CS, & ISE
USC

Lorentz Math Center
Leiden, Netherlands
August 7, 2024

Collaborator on Slides:

Collaborator on Slides:

Zalan Fabian

Motivation

Many success stories

Many success stories

Recently generative AI has achieved impressive performance

Many success stories

Recently generative AI has achieved impressive performance

Many success stories

Recently generative AI has achieved impressive performance

Many success stories

Recently generative AI has achieved impressive performance

Many success stories

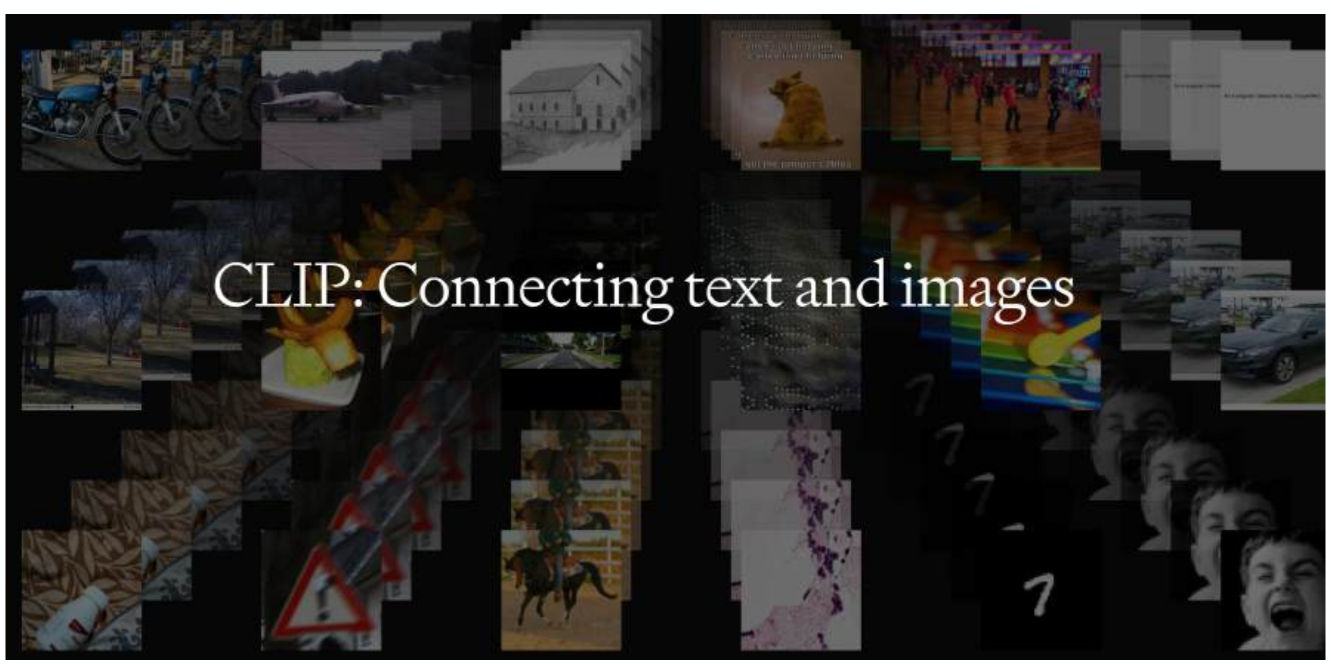
Recently generative AI has achieved impressive performance

Multi-modal

Many success stories

Recently generative AI has achieved impressive performance

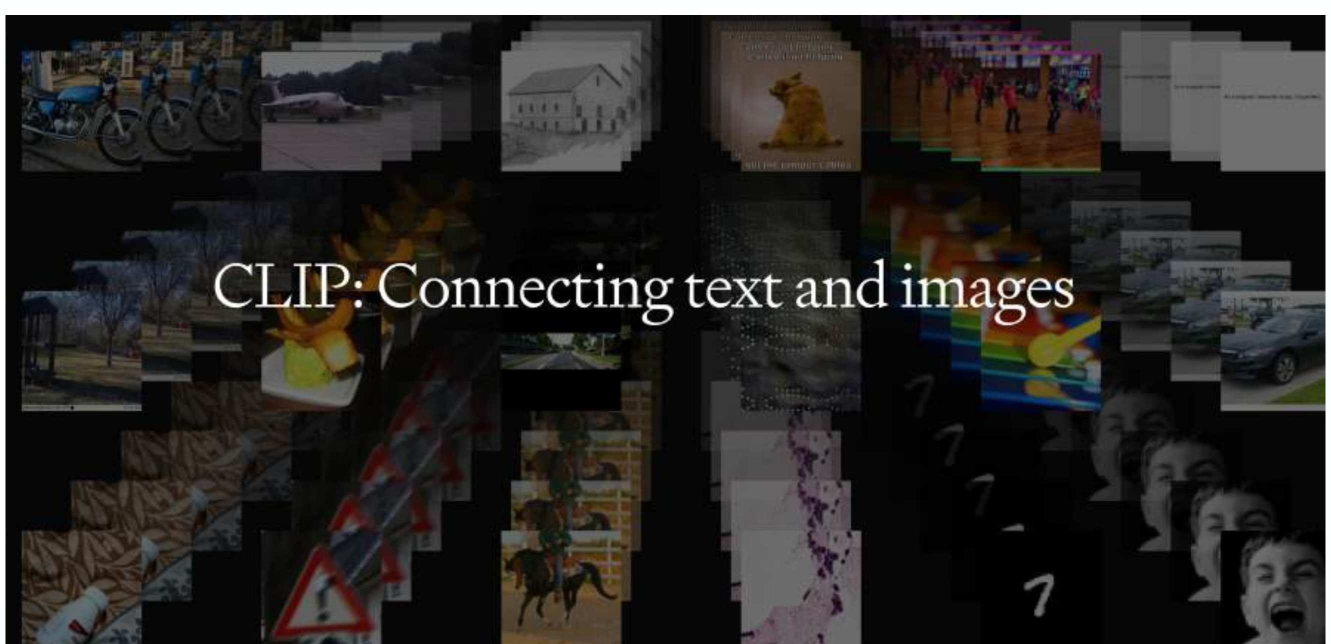
Multi-modal



Many success stories

Recently generative AI has achieved impressive performance

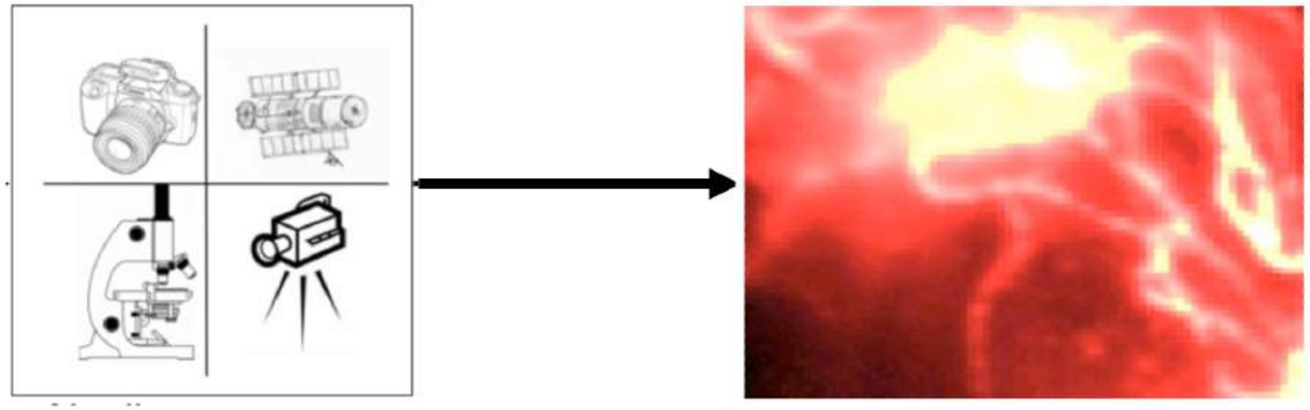
Multi-modal



What about Inverse Problems?

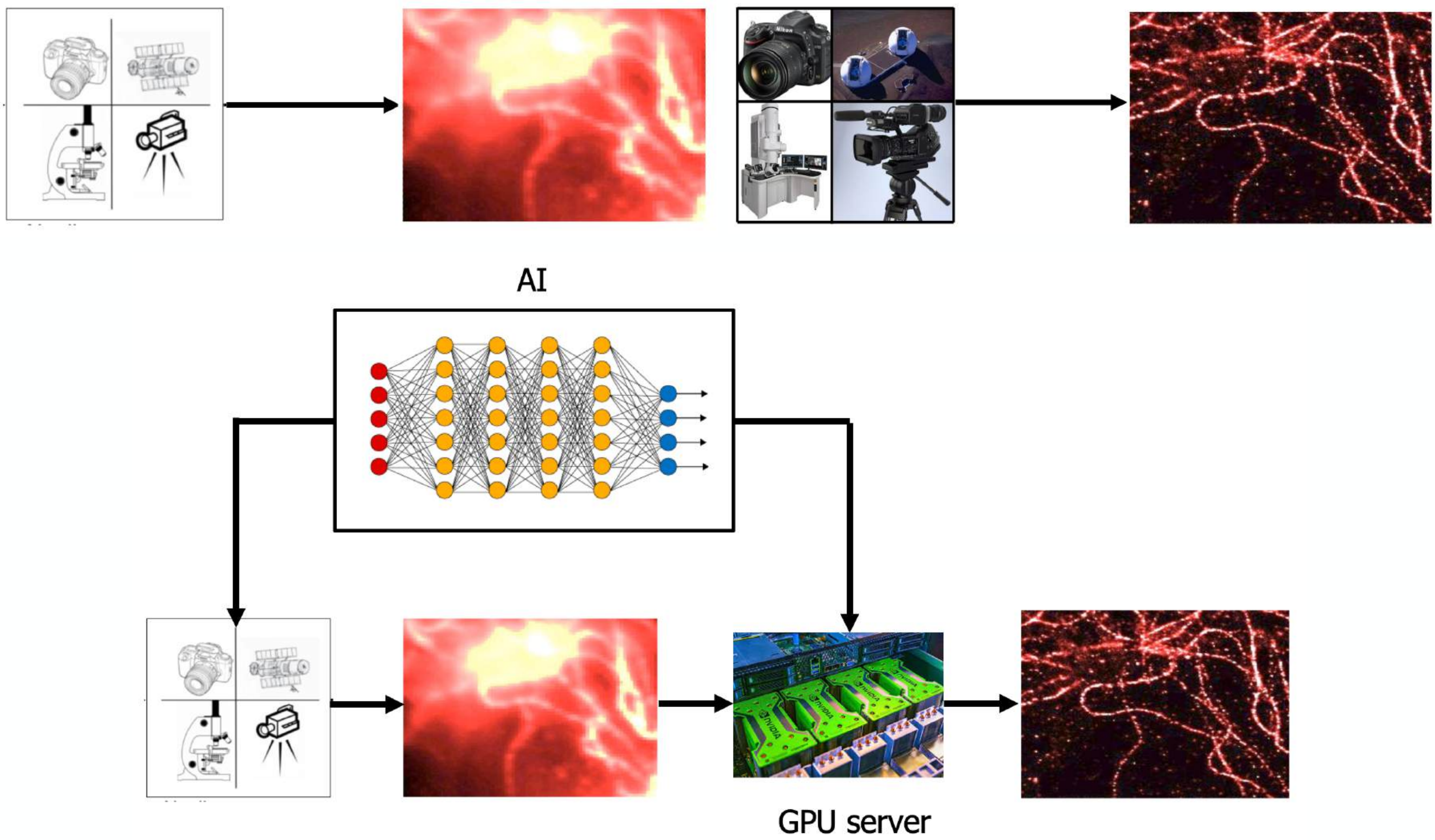
AI for Comp Imaging/Inverse Problems

AI for Comp Imaging/Inverse Problems



AI for Comp Imaging/Inverse Problems

AI for Comp Imaging/Inverse Problems



This Tutorial:
Opportunities and Challenges of AI for Inverse Problems

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Ground truth
(signal to be recovered)

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Forward operator
(possibly non-linear)

Ground truth
(signal to be recovered)

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Forward operator
(possibly non-linear)

Ground truth
(signal to be recovered)

Noise

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Measurements
(observations)

Forward operator
(possibly non-linear)

Ground truth
(signal to be recovered)

Noise

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Measurements
(observations)

Forward operator
(possibly non-linear)

Ground truth
(signal to be recovered)

Noise

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Measurements
(observations)

Forward operator
(possibly non-linear)

Ground truth
(signal to be recovered)

Noise

  - Typically we have more unknowns than measurements

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Measurements (observations)

Forward operator (possibly non-linear)

Ground truth (signal to be recovered)

Noise

  - Typically we have more unknowns than measurements

Inverse problems

- Formulation

$$y = \mathscr{A}(x) + \varepsilon$$

Measurements (observations)

Forward operator (possibly non-linear)

Ground truth (signal to be recovered)

Noise

- Typically we have more unknowns than measurements
- Infinitely many solutions can be consistent with the observations

Common inverse problems (1)

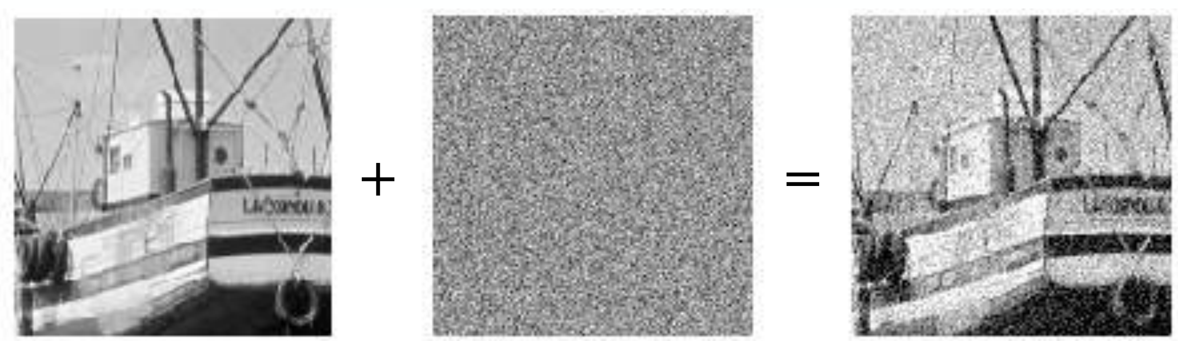
- Denoising

$$A = I, \qquad y = x + \varepsilon$$

Common inverse problems (1)

- Denoising

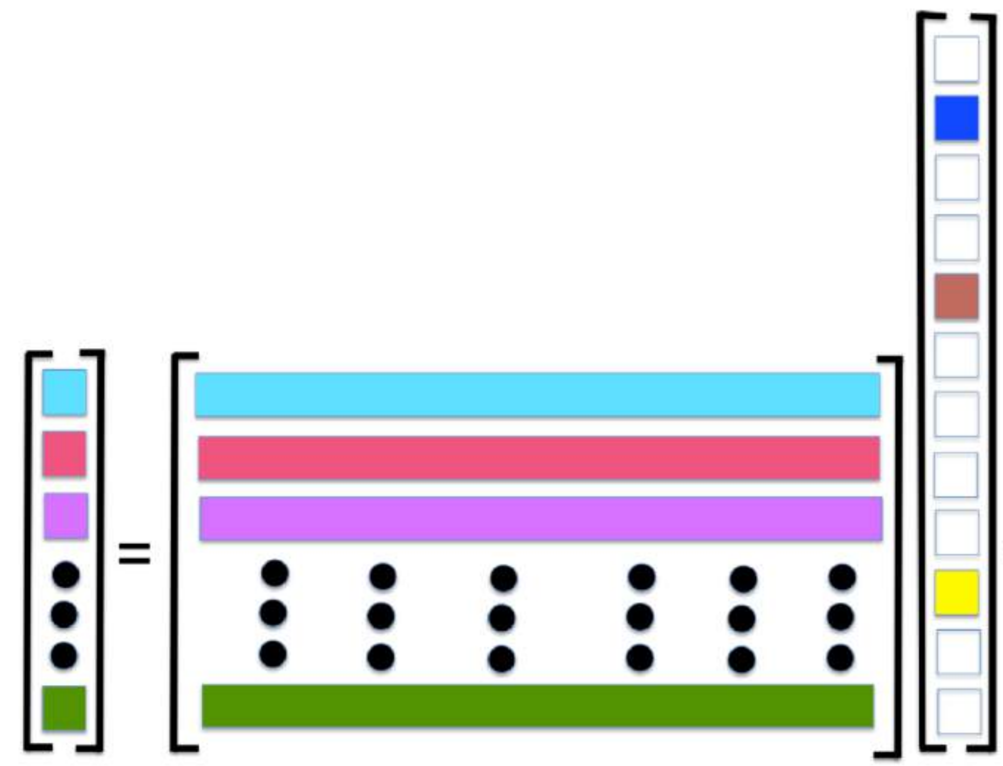
$$A = I, \qquad y = x + \varepsilon$$



Common inverse problems (2)

- Compressed sensing

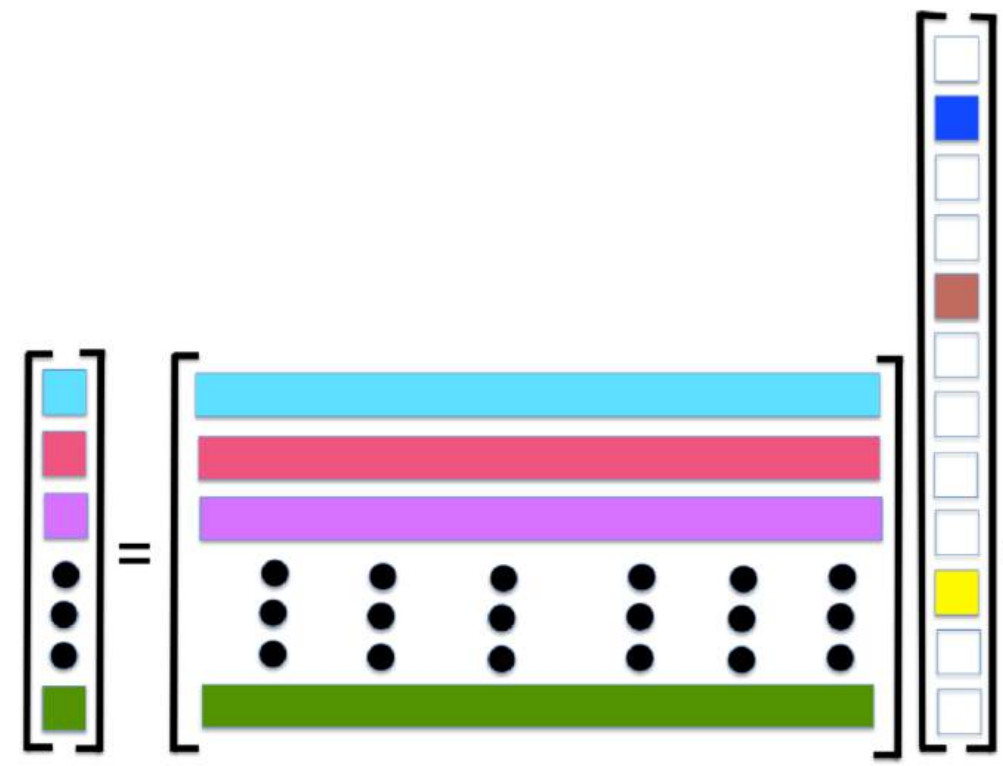
$$A = \begin{cases} \text{DFT + subsampling} \\ \text{Gaussian ensemble} \\ \text{Bernoulli ensemble} \end{cases}$$



Common inverse problems (2)

- Compressed sensing

$$A = \begin{cases} \text{DFT + subsampling} \\ \text{Gaussian ensemble} \\ \text{Bernoulli ensemble} \end{cases}$$

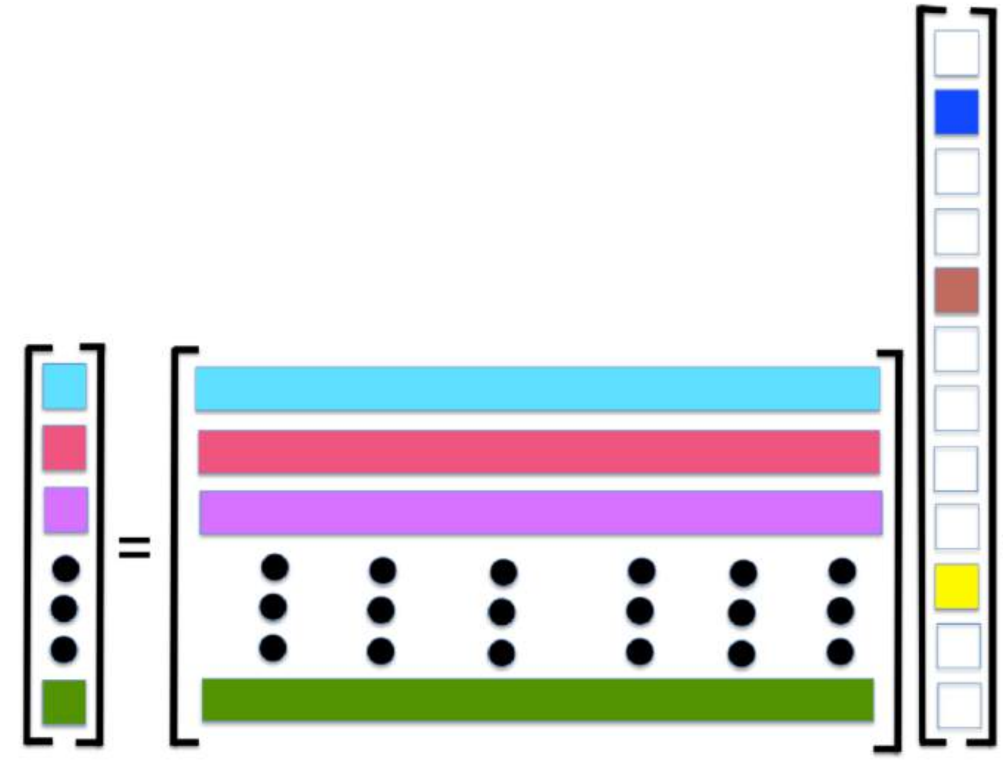


  - nice theoretical guarantees on recovery

Common inverse problems (2)

- Compressed sensing

$$A = \begin{cases} \text{DFT + subsampling} \\ \text{Gaussian ensemble} \\ \text{Bernoulli ensemble} \end{cases}$$

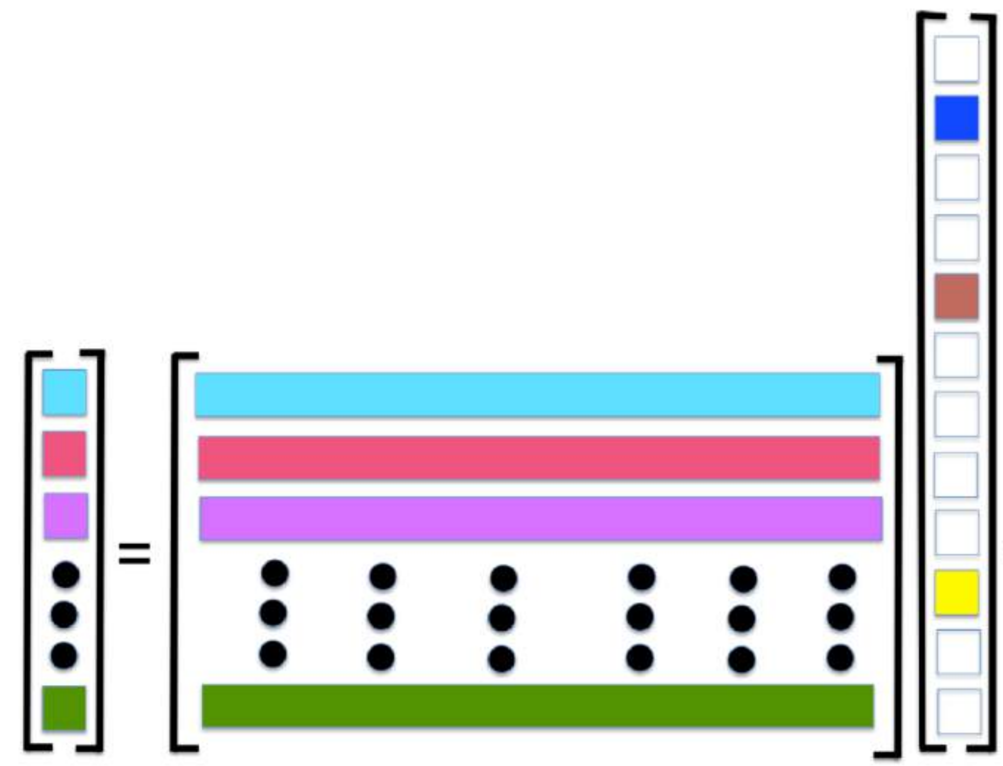


- nice theoretical guarantees on recovery
- exploiting sparsity in transform domain

Common inverse problems (2)

- Compressed sensing

$$A = \begin{cases} \text{DFT + subsampling} \\ \text{Gaussian ensemble} \\ \text{Bernoulli ensemble} \end{cases}$$



- nice theoretical guarantees on recovery
- exploiting sparsity in transform domain
- special case: MRI reconstruction

Common inverse problems (3)

- Deconvolution

$$\mathscr{A}(x) = h * x$$

Common inverse problems (3)

- Deconvolution

$$\mathscr{A}(x) = h * x$$

h : **blur kernel**

  - if unknown: blind deconvolution

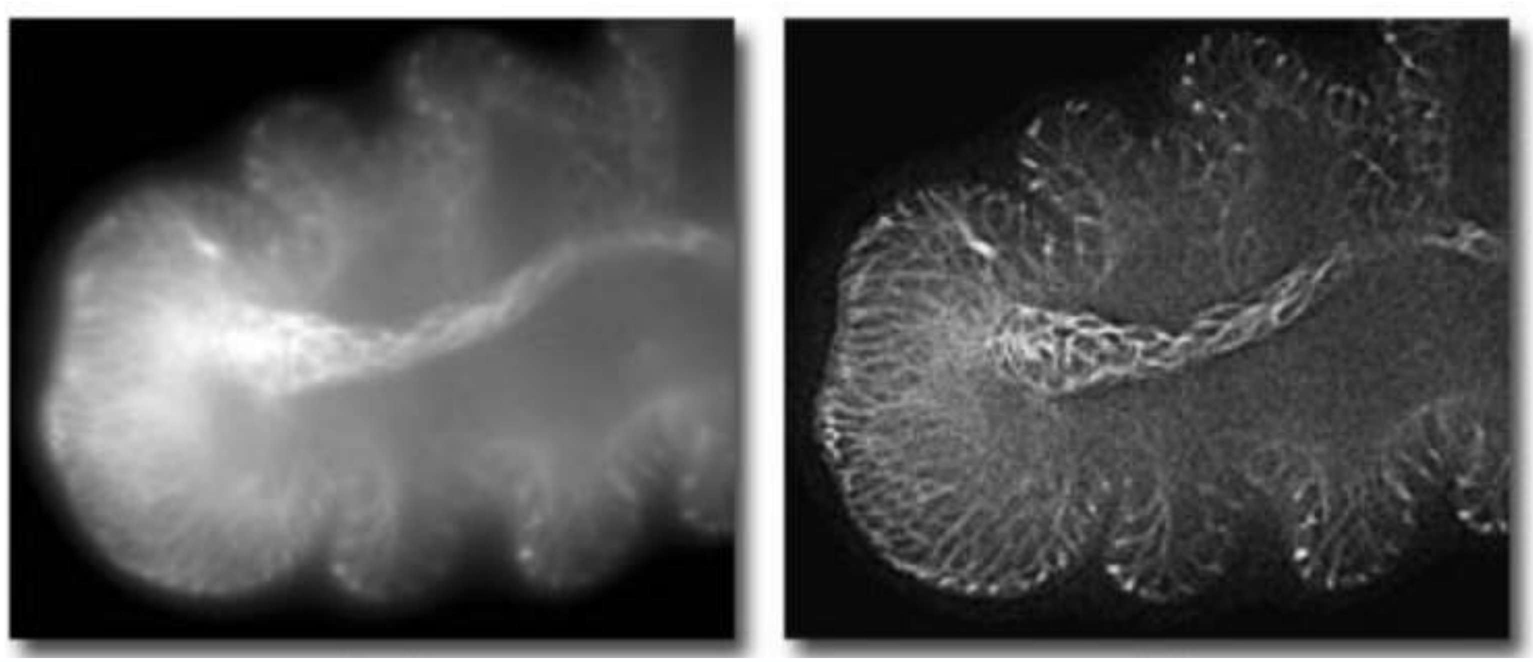
Common inverse problems (3)

- Deconvolution

$$\mathscr{A}(x) = h * x$$

h : **blur kernel**

- if unknown: blind deconvolution



Common inverse problems (4)

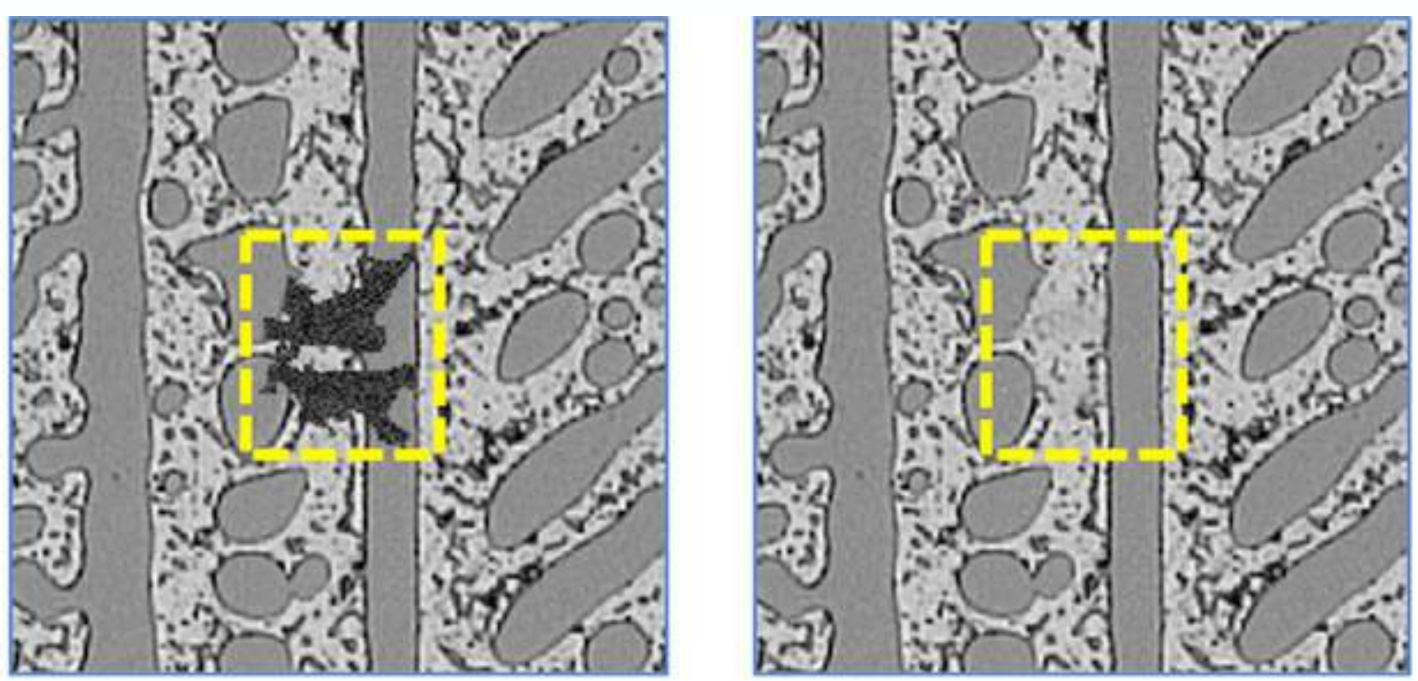
- Inpainting

$$A = S \quad \text{diagonal operator, with} \quad S_{ii} = \begin{cases} 1 & \text{,if } x_i \text{ is sampled} \\ 0 & \text{,otherwise} \end{cases}$$

Common inverse problems (4)

- Inpainting

$$A = S \quad \text{diagonal operator, with} \quad S_{ii} = \begin{cases} 1 & \text{,if } x_i \text{ is sampled} \\ 0 & \text{,otherwise} \end{cases}$$



Damaged image *Restored image*

Common inverse problems (5)

- Phase retrieval

$$\mathscr{A}(x) = |Ax|^2$$

Common inverse problems (5)

- Phase retrieval

$$\mathscr{A}(x) = |Ax|^2$$

  - recover phase information from magnitude-only measurements

Common inverse problems (5)

- Phase retrieval

$$\mathscr{A}(x) = |Ax|^2$$

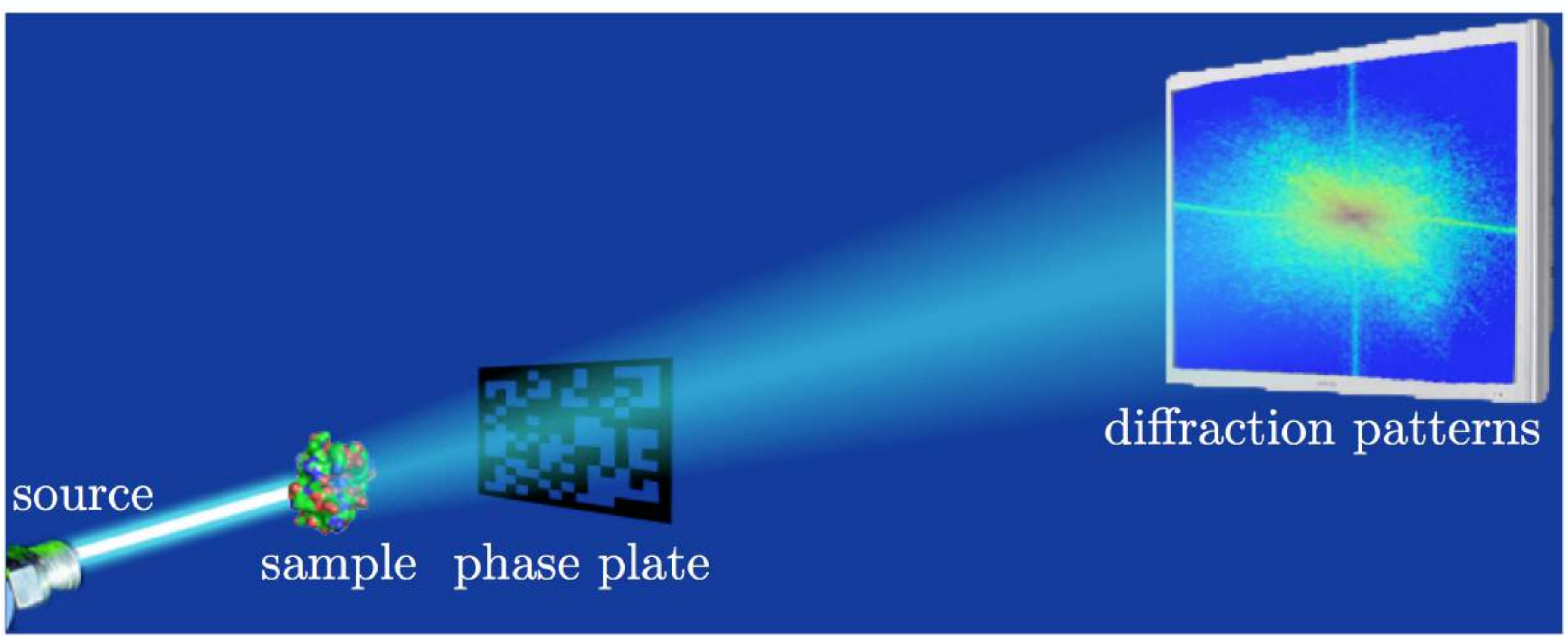
  - recover phase information from magnitude-only measurements
  - A is often DFT-like measurement matrix

Common inverse problems (5)

- Phase retrieval

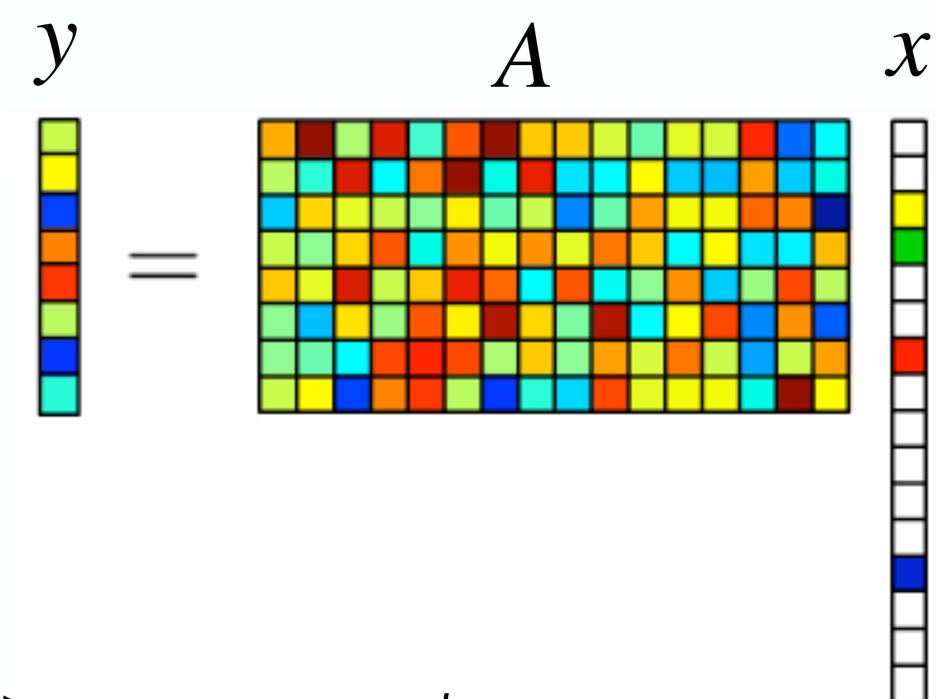
$$\mathscr{A}(x) = |Ax|^2$$

  - recover phase information from magnitude-only measurements
  - A is often DFT-like measurement matrix



Tackling inverse problems (1)

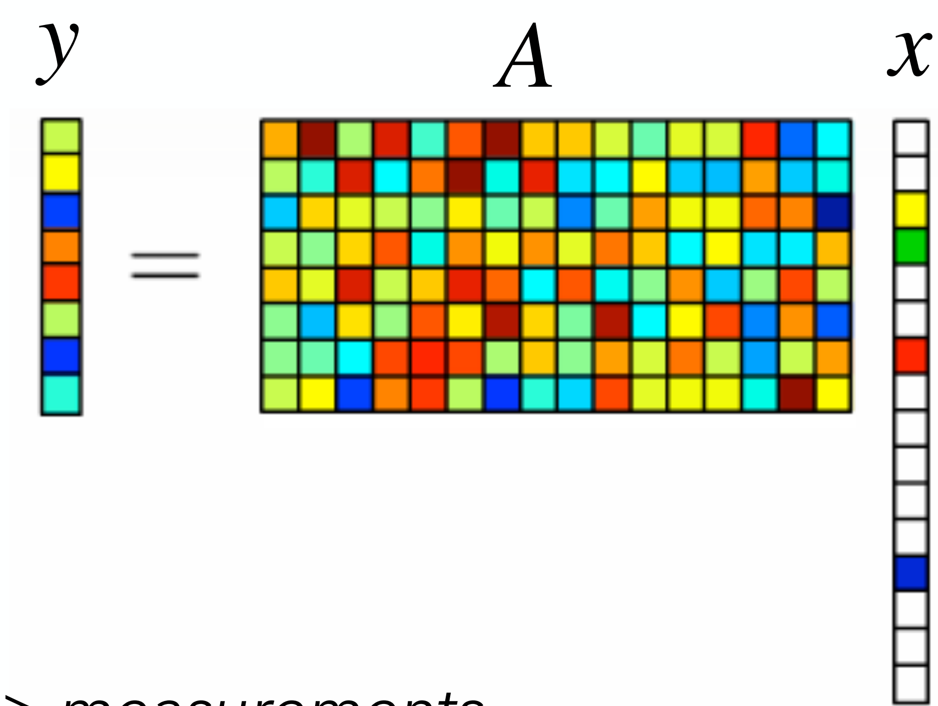
- Simplest case



  - Typically: *unknowns >> measurements*
  - No unique solution
  - **Goal**: find the "best" solution that is consistent with the measurements

Tackling inverse problems (1)

- Simplest case

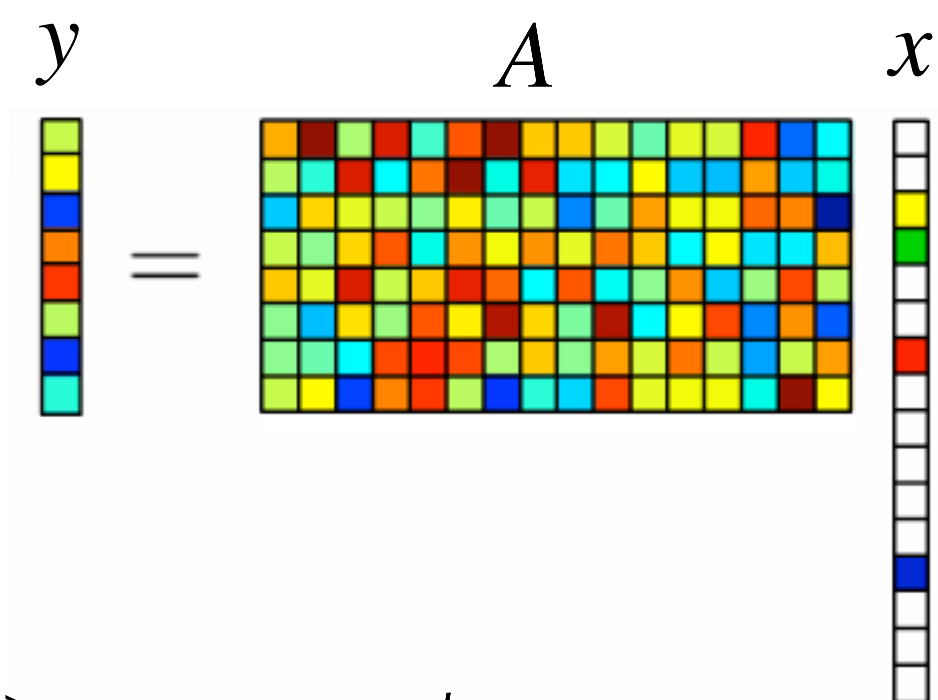


  - Typically: *unknowns >> measurements*
  - No unique solution
  - **Goal**: find the "best" solution that is consistent with the measurements

Minimum norm solution: $\hat{x} = A^T(AA^T)^{-1}y$

Tackling inverse problems (1)

- Simplest case



  - Typically: *unknowns >> measurements*
  - No unique solution
  - **Goal**: find the "best" solution that is consistent with the measurements

Minimum norm solution: $\hat{x} = A^T(AA^T)^{-1}y$

Can we do better by leveraging side-information?

Tackling inverse problems (2)

- Compressed sensing formulation

Tackling inverse problems (2)

- Compressed sensing formulation

$$\min_x \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

data consistency

prior knowledge

Tackling inverse problems (2)

- Compressed sensing formulation

$$\min_x \underbrace{\|\mathscr{A}(x) - y\|^2}_{\text{data consistency}} + \underbrace{\mathscr{R}(x)}_{\text{prior knowledge}}$$

  - Regularizer enforces prior knowledge on signal structure

Tackling inverse problems (2)

- Compressed sensing formulation

$$\underbrace{\min_x \|\mathscr{A}(x) - y\|^2}_{\text{data consistency}} + \underbrace{\mathscr{R}(x)}_{\text{prior knowledge}}$$

  - Regularizer enforces prior knowledge on signal structure
  - Most often: sparsity in some transform domain (Fourier, Wavelet)

Tackling inverse problems (2)

- Compressed sensing formulation

$$\min_x \|\underbrace{\mathscr{A}(x) - y\|^2}_{\text{data consistency}} + \underbrace{\mathscr{R}(x)}_{\text{prior knowledge}}$$

  - Regularizer enforces prior knowledge on signal structure
  - Most often: sparsity in some transform domain (Fourier, Wavelet)
  - Solution: numerical, iterative algorithms

Tackling inverse problems (3)

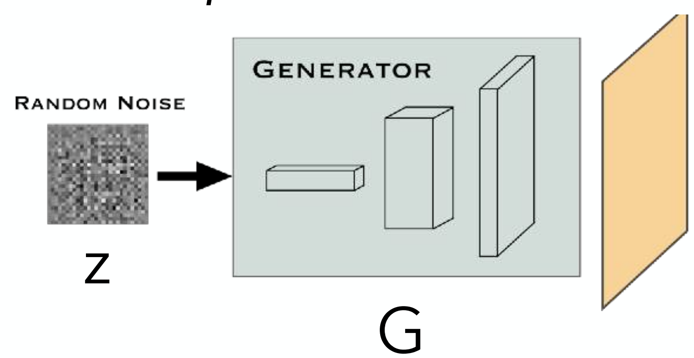
- Data-driven methods

Tackling inverse problems (3)

- Data-driven methods
  - *Generative prior-based methods:* learn distribution of x from data

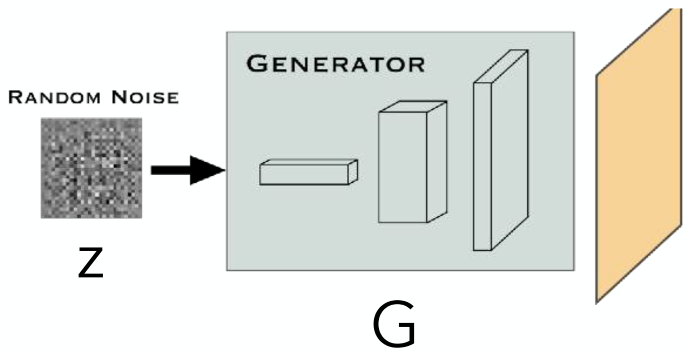
Tackling inverse problems (3)

- Data-driven methods
  - *Generative prior-based methods:* learn distribution of x from data



Tackling inverse problems (3)

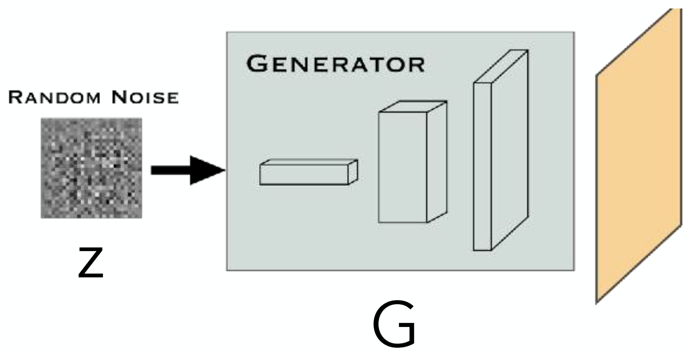
- Data-driven methods
  - *Generative prior-based methods:* learn distribution of x from data



$$\min_{z} \mathscr{L}(AG(z), y)$$

Tackling inverse problems (3)

- Data-driven methods
  - *Generative prior-based methods:* learn distribution of x from data

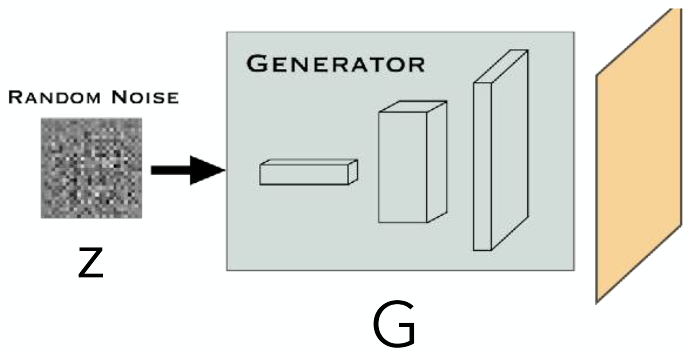


$$\min_z \mathscr{L}(AG(z), y)$$

  - *End-to-end methods*: learn mapping from measurements to reconstruction directly

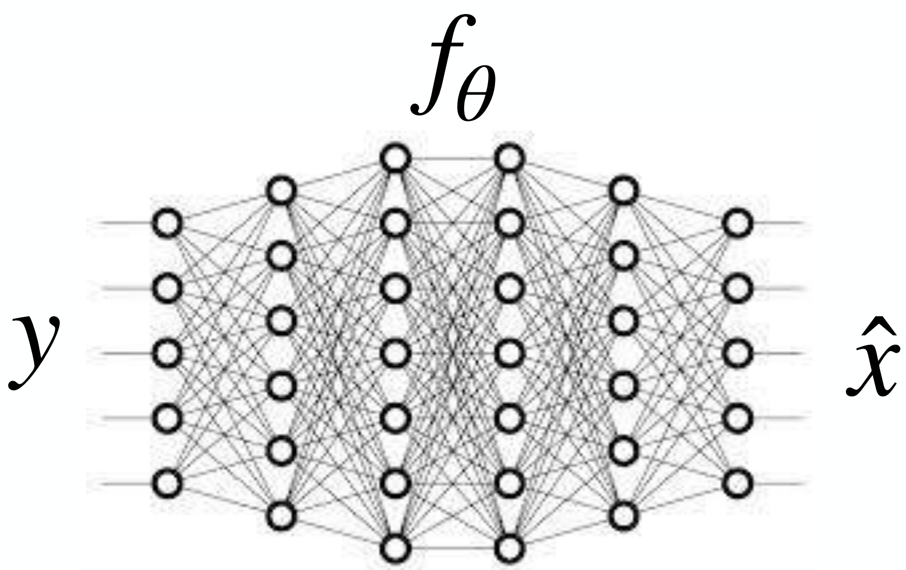
Tackling inverse problems (3)

- Data-driven methods
  - *Generative prior-based methods:* learn distribution of x from data



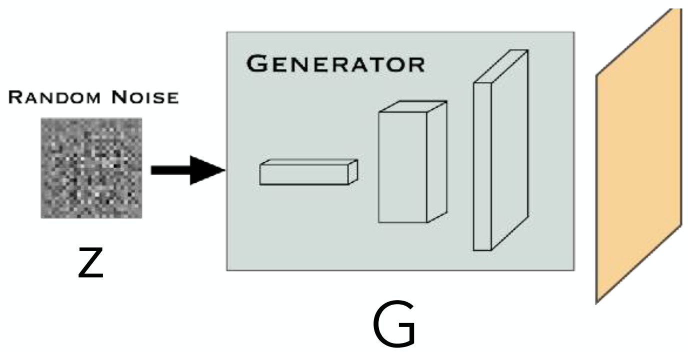
$$\min_{z} \mathscr{L}(AG(z), y)$$

  - *End-to-end methods*: learn mapping from measurements to reconstruction directly



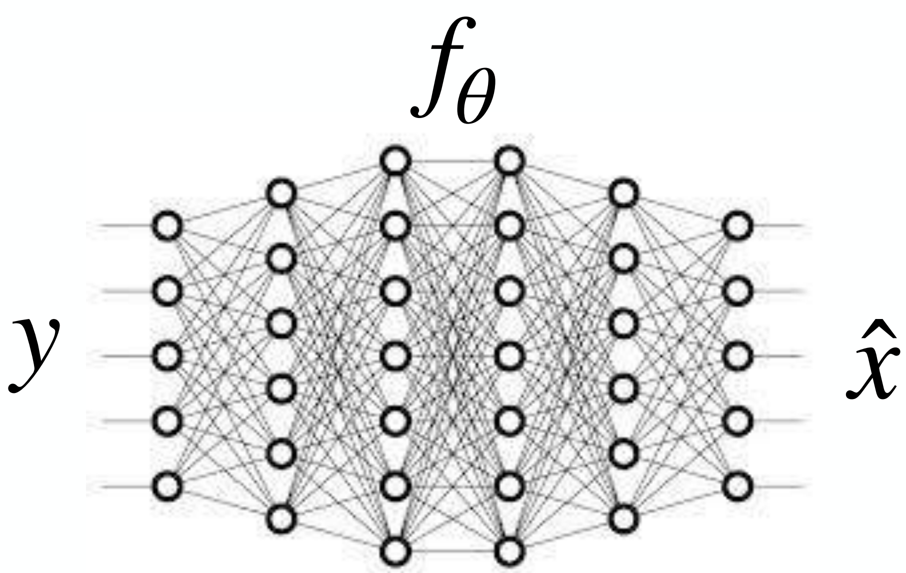
Tackling inverse problems (3)

- Data-driven methods
  - *Generative prior-based methods:* learn distribution of x from data



$$\min_{z} \mathscr{L}(AG(z), y)$$

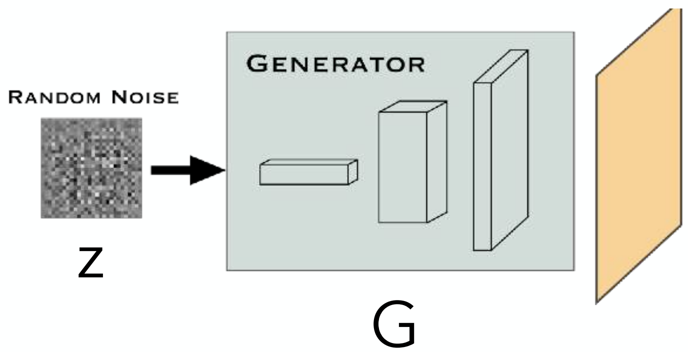
  - *End-to-end methods*: learn mapping from measurements to reconstruction directly



$$\min_{\theta} \mathscr{L}(f_\theta(y), x)$$

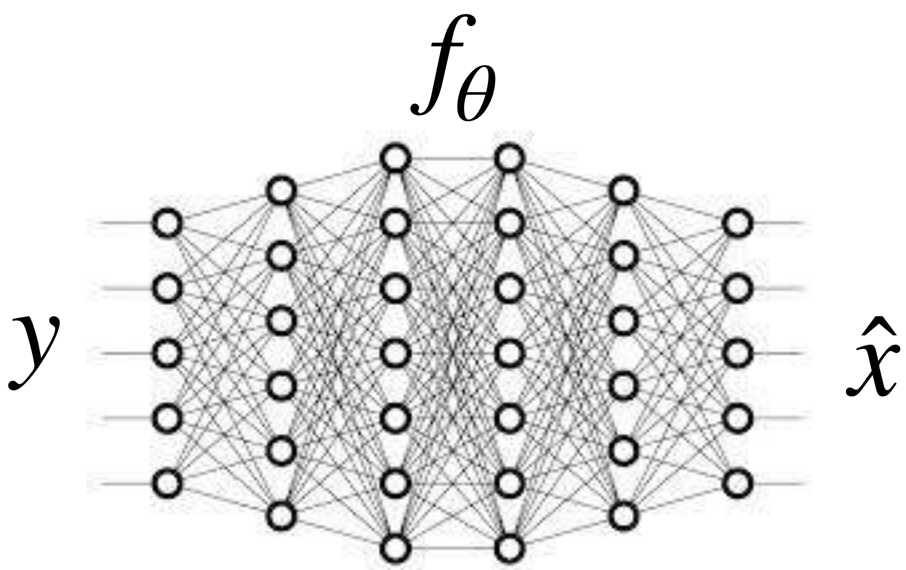
Tackling inverse problems (3)

- Data-driven methods
  - *Generative prior-based methods:* learn distribution of x from data



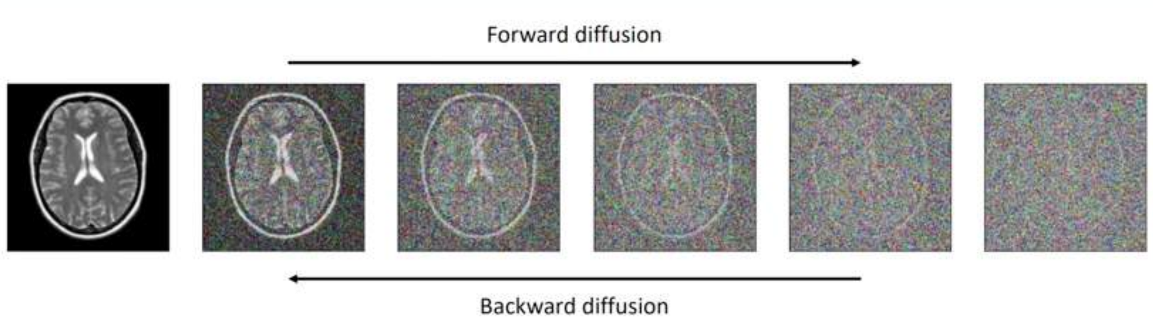
$$\min_z \mathscr{L}(AG(z), y)$$

  - *End-to-end methods*: learn mapping from measurements to reconstruction directly



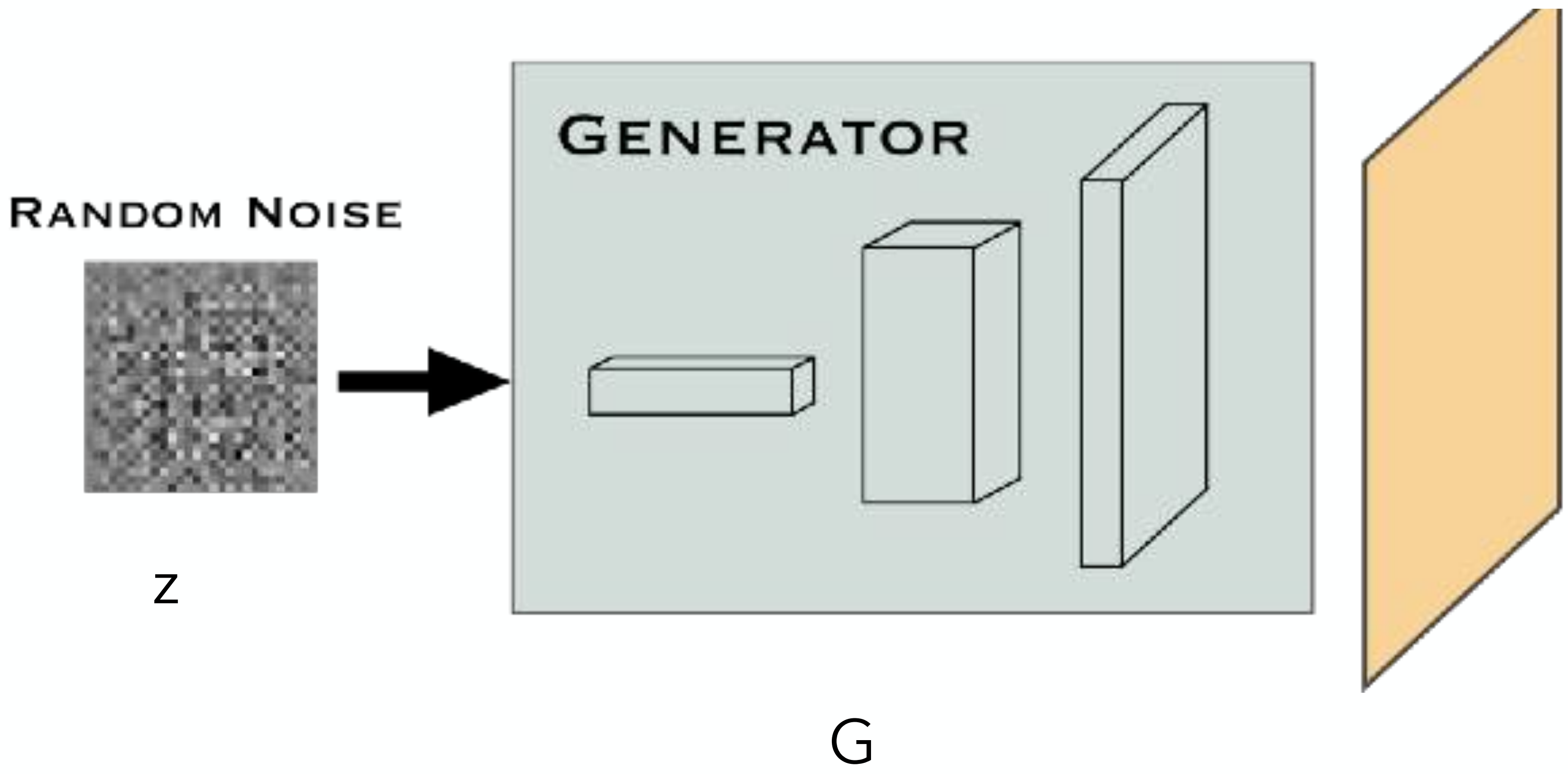
$$\min_\theta \mathscr{L}(f_\theta(y), x)$$

  - *Diffusion solvers*



Deep Generative Models
In Inverse Problems

Deep Generative Models In Inverse Problems



Deep Generative Models (DGM)

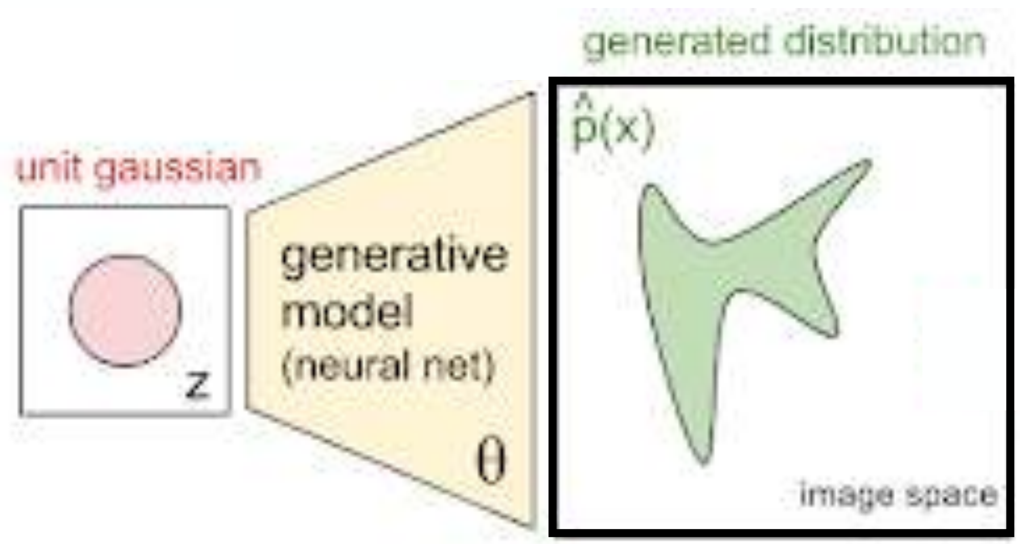
- Challenge: efficient modeling of high-dimensional distributions

Deep Generative Models (DGM)

- Challenge: efficient modeling of high-dimensional distributions
- Learn mapping from simple “seed” distribution to complex data distribution

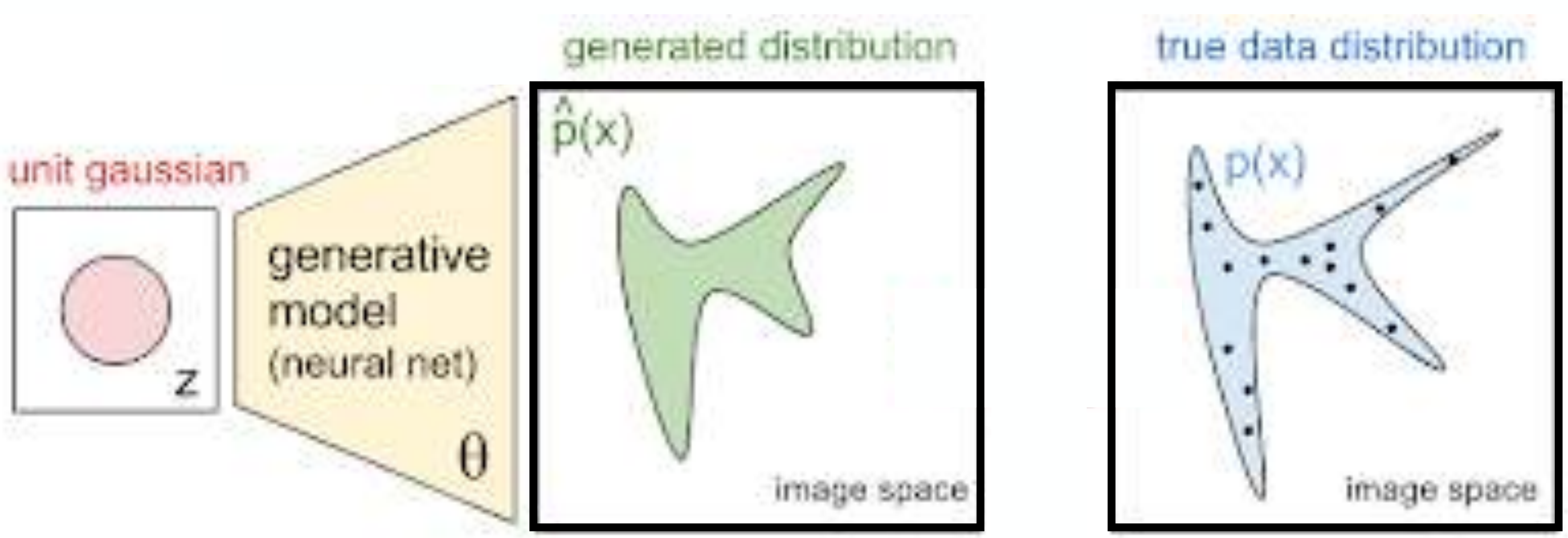
Deep Generative Models (DGM)

- Challenge: efficient modeling of high-dimensional distributions
- Learn mapping from simple “seed” distribution to complex data distribution
- Parameterize mapping as a deep neural network



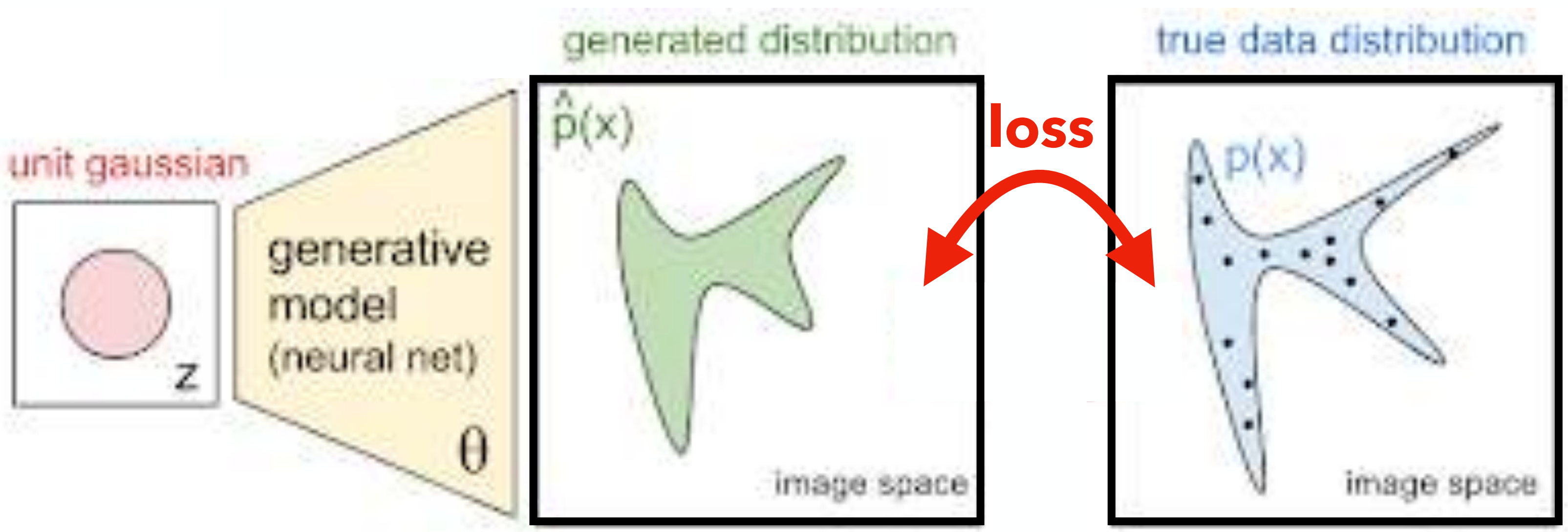
Deep Generative Models (DGM)

- Challenge: efficient modeling of high-dimensional distributions
- Learn mapping from simple “seed” distribution to complex data distribution
- Parameterize mapping as a deep neural network



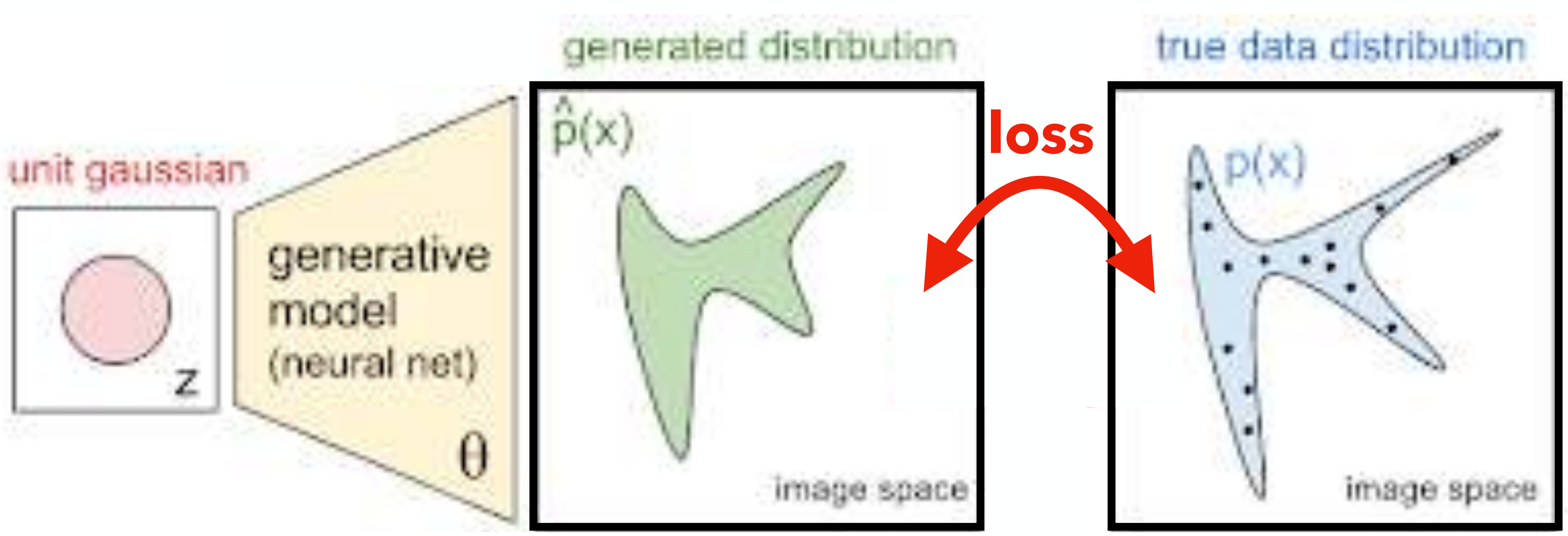
Deep Generative Models (DGM)

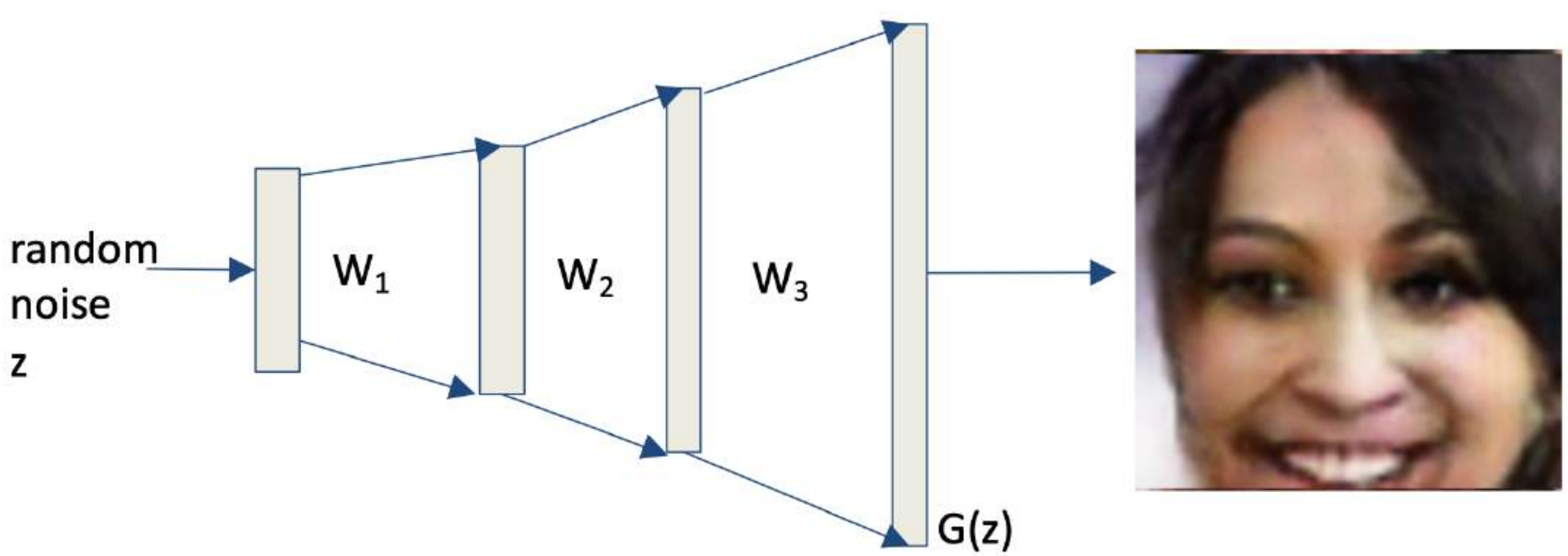
- Challenge: efficient modeling of high-dimensional distributions
- Learn mapping from simple “seed” distribution to complex data distribution
- Parameterize mapping as a deep neural network



Deep Generative Models (DGM)

- Challenge: efficient modeling of high-dimensional distributions
- Learn mapping from simple “seed” distribution to complex data distribution
- Parameterize mapping as a deep neural network

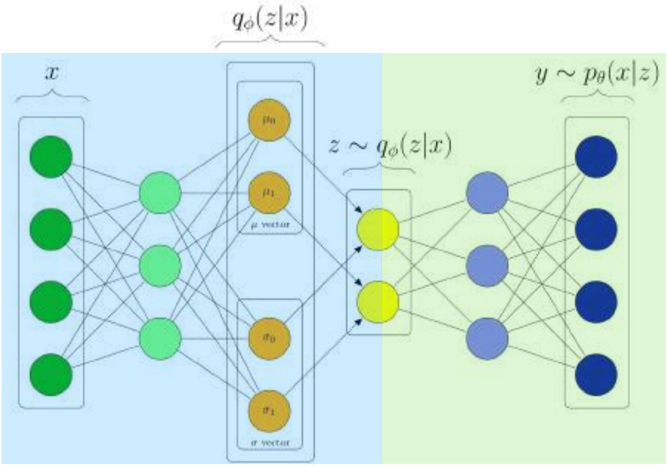




Timeline of Deep Generative Architectures

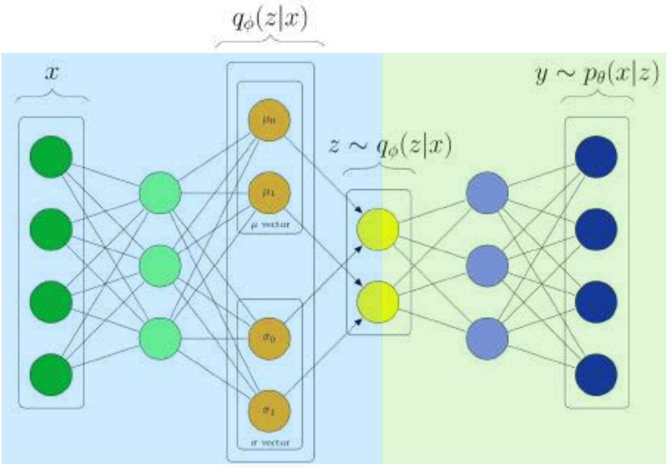
Timeline of Deep Generative Architectures

- Variational Autoencoders (2013)

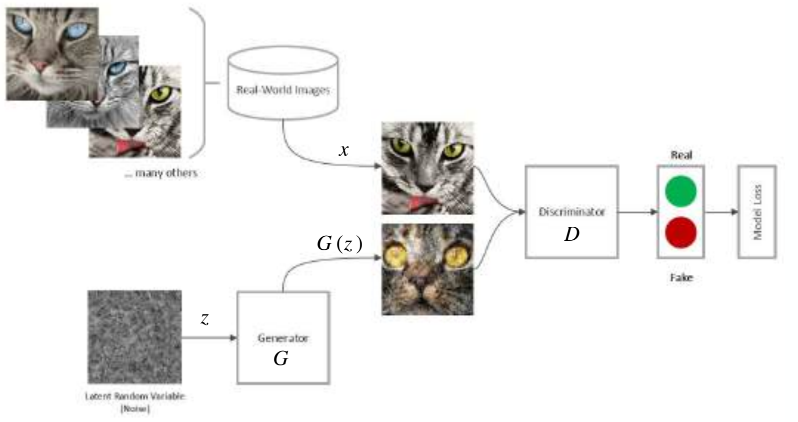


Timeline of Deep Generative Architectures

- Variational Autoencoders (2013)

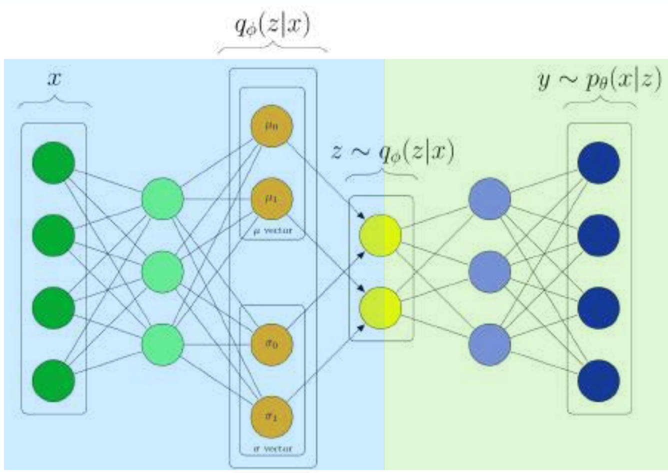


- Generative Adversarial Networks (2014)

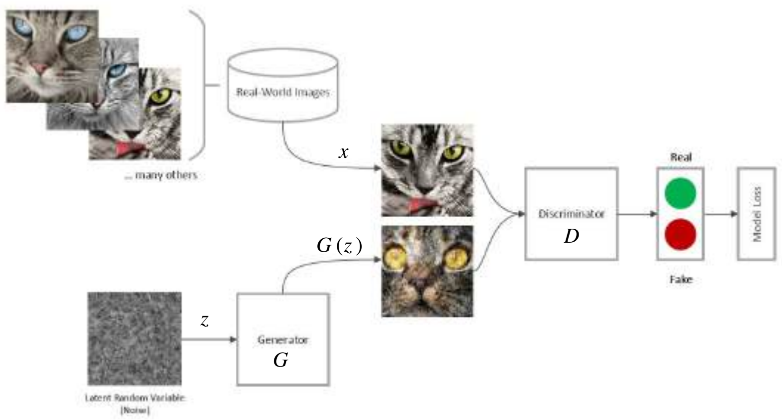


Timeline of Deep Generative Architectures

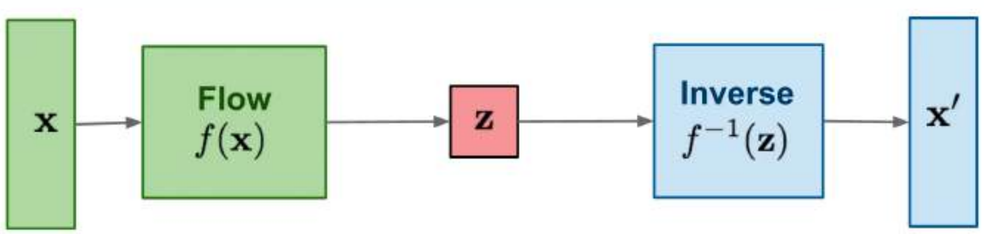
- Variational Autoencoders (2013)



- Generative Adversarial Networks (2014)

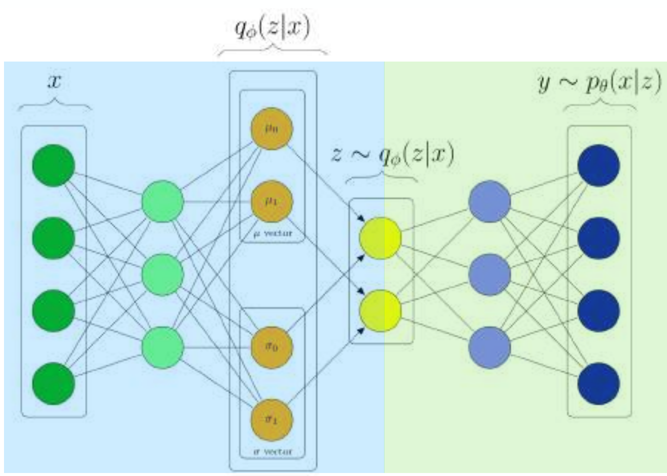


- Normalizing Flow Models (2015)

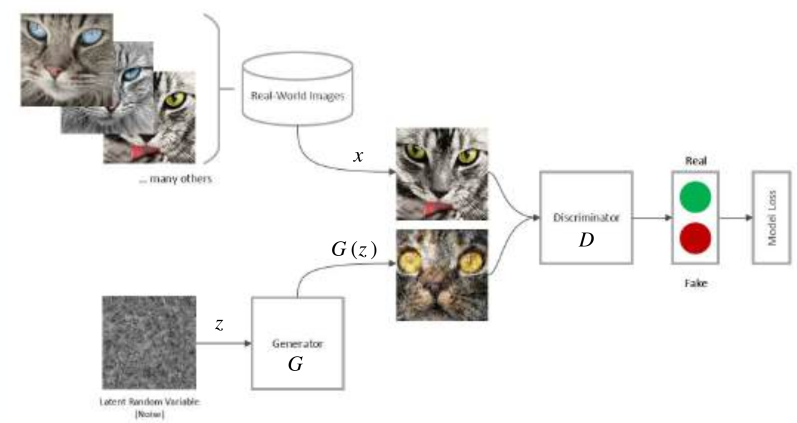


Timeline of Deep Generative Architectures

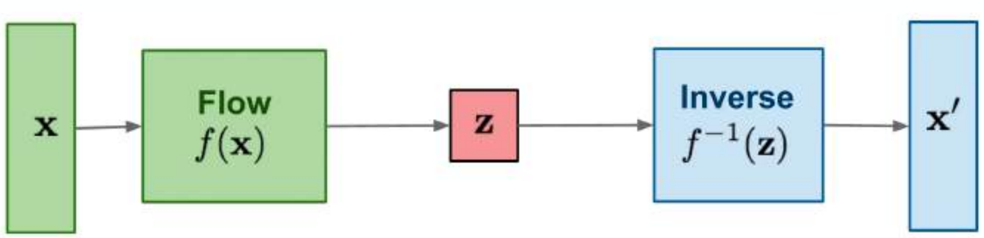
- Variational Autoencoders (2013)



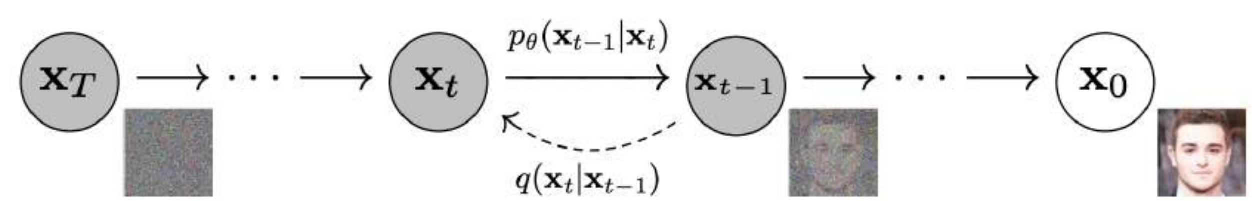
- Generative Adversarial Networks (2014)



- Normalizing Flow Models (2015)



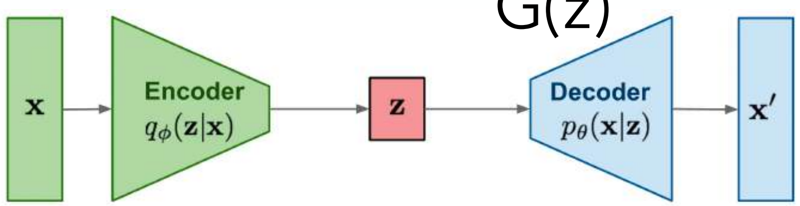
- Diffusion Models (2015, dominantly since 2020)



Compressed Sensing using Generative Models

Step 1: Pre-train generator on available training data

G(z)



VAE (example)

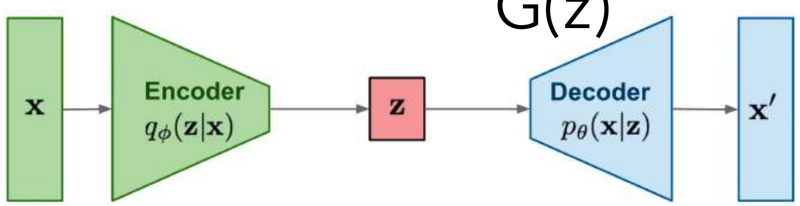
$$\min_{\theta} \quad \sum_{i=1}^{n} \|G_\theta(\mathbf{z}_i) - x_i\|_{\ell_2}^2$$

Bora, A., Jalal, A., Price, E. and Dimakis, A.G., 2017, July. Compressed sensing using generative models. In *International Conference on Machine Learning* (pp. 537-546). PMLR.

Compressed Sensing using Generative Models

Step 1: Pre-train generator on available training data

G(z)



VAE (example)

$$\min_{\theta} \quad \sum_{i=1}^{n} \|G_\theta(\mathbf{z}_i) - x_i\|_{\ell_2}^2$$

Step 2: Leverage generative prior for reconstruction

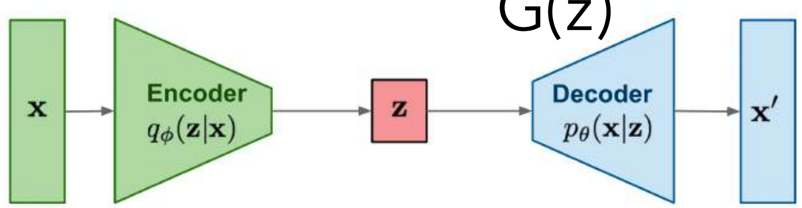
$$\min_{z} ||AG_\theta(z) - y||^2$$

Bora, A., Jalal, A., Price, E. and Dimakis, A.G., 2017, July. Compressed sensing using generative models. In *International Conference on Machine Learning* (pp. 537-546). PMLR.

Compressed Sensing using Generative Models

Step 1: Pre-train generator on available training data

G(z)



VAE (example)

$$\min_\theta \sum_{i=1}^{n} \|G_\theta(\mathbf{z}_i) - x_i\|_{\ell_2}^2$$

Step 2: Leverage generative prior for reconstruction

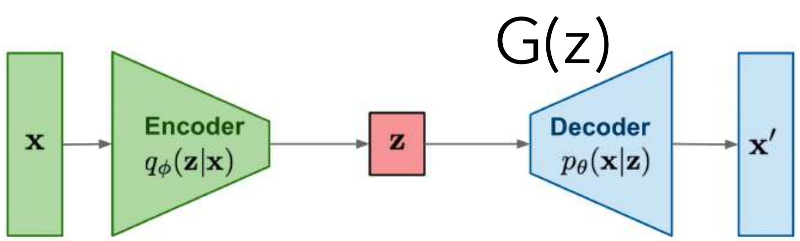
$$\boxed{\min_z} \, || A G_\theta(z) - y ||^2$$

- find vector in latent space of a generator

Bora, A., Jalal, A., Price, E. and Dimakis, A.G., 2017, July. Compressed sensing using generative models. In *International Conference on Machine Learning* (pp. 537-546). PMLR.

Compressed Sensing using Generative Models

Step 1: Pre-train generator on available training data



VAE (example)

$$\min_{\theta} \sum_{i=1}^{n} \|G_\theta(\mathbf{z}_i) - x_i\|_{\ell_2}^2$$

Step 2: Leverage generative prior for reconstruction

$$\min_{z} ||AG_\theta(z) - y||^2$$

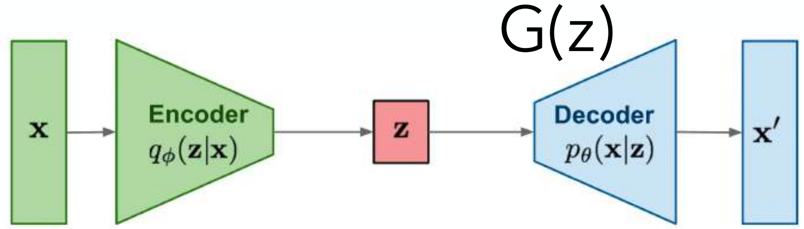
- find vector in latent space of a generator
- such that corresponding sample in data space

Bora, A., Jalal, A., Price, E. and Dimakis, A.G., 2017, July. Compressed sensing using generative models. In *International Conference on Machine Learning* (pp. 537-546). PMLR.

Compressed Sensing using Generative Models

Step 1: Pre-train generator on available training data

G(z)



VAE (example)

$$\min_{\theta} \sum_{i=1}^{n} \|G_\theta(\mathbf{z}_i) - x_i\|_{\ell_2}^2$$

Step 2: Leverage generative prior for reconstruction

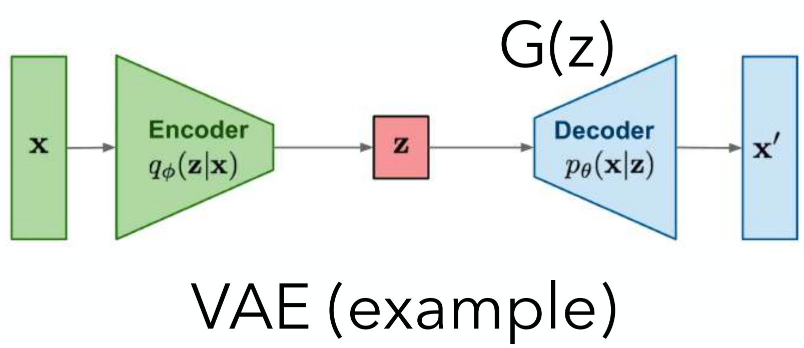
$$\min_{z} ||AG_\theta(z) - y||^2$$

- find vector in latent space of a generator
- such that corresponding sample in data space
- is consistent with the observations

Bora, A., Jalal, A., Price, E. and Dimakis, A.G., 2017, July. Compressed sensing using generative models. In *International Conference on Machine Learning* (pp. 537-546). PMLR.

Compressed Sensing using Generative Models

Step 1: Pre-train generator on available training data



$$\min_\theta \sum_{i=1}^{n} \|G_\theta(\mathbf{z}_i) - x_i\|_{\ell_2}^2$$

Step 2: Leverage generative prior for reconstruction

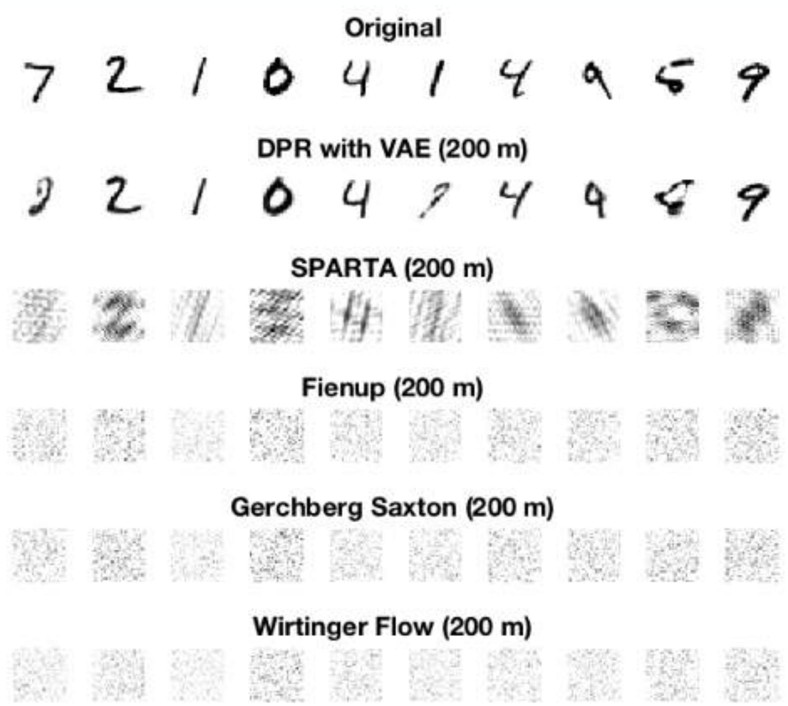
$$\min_z ||AG_\theta(z) - y||^2$$

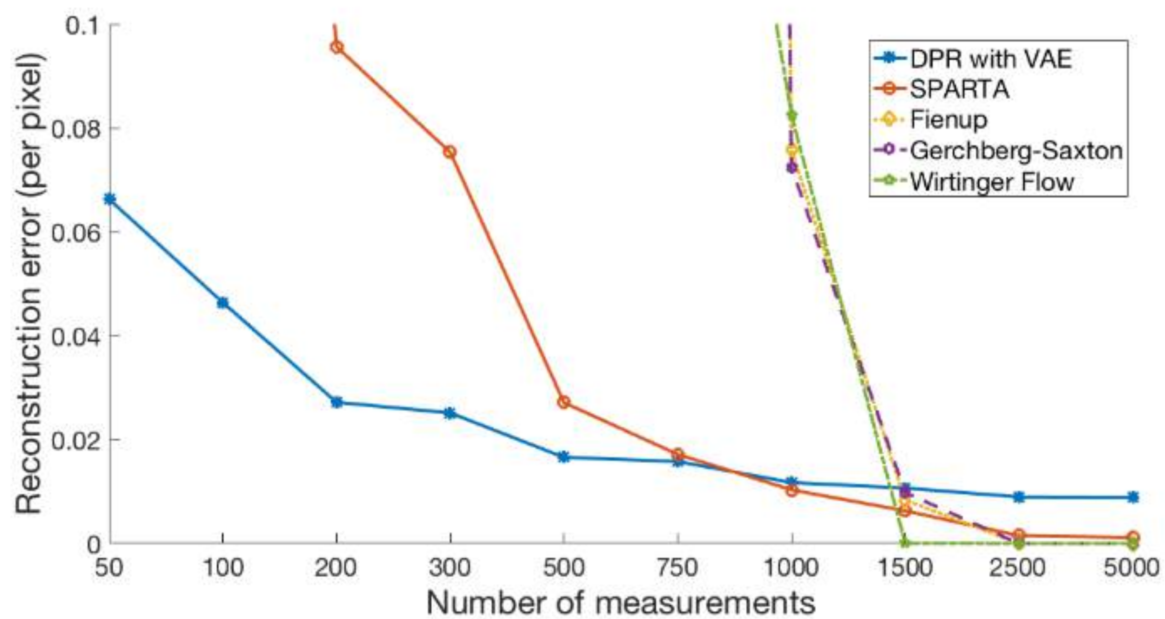
- find vector in latent space of a generator
- such that corresponding sample in data space
- is consistent with the observations

For given MSE, 5-10x less measurements than classical sparsity-based methods!

Bora, A., Jalal, A., Price, E. and Dimakis, A.G., 2017, July. Compressed sensing using generative models. In *International Conference on Machine Learning* (pp. 537-546). PMLR.

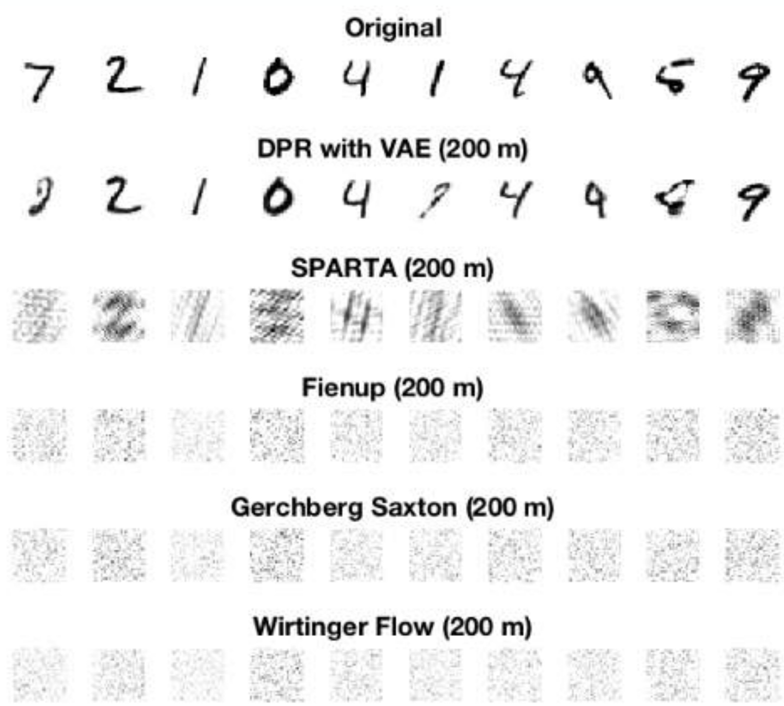
DGM in Phase Retrieval

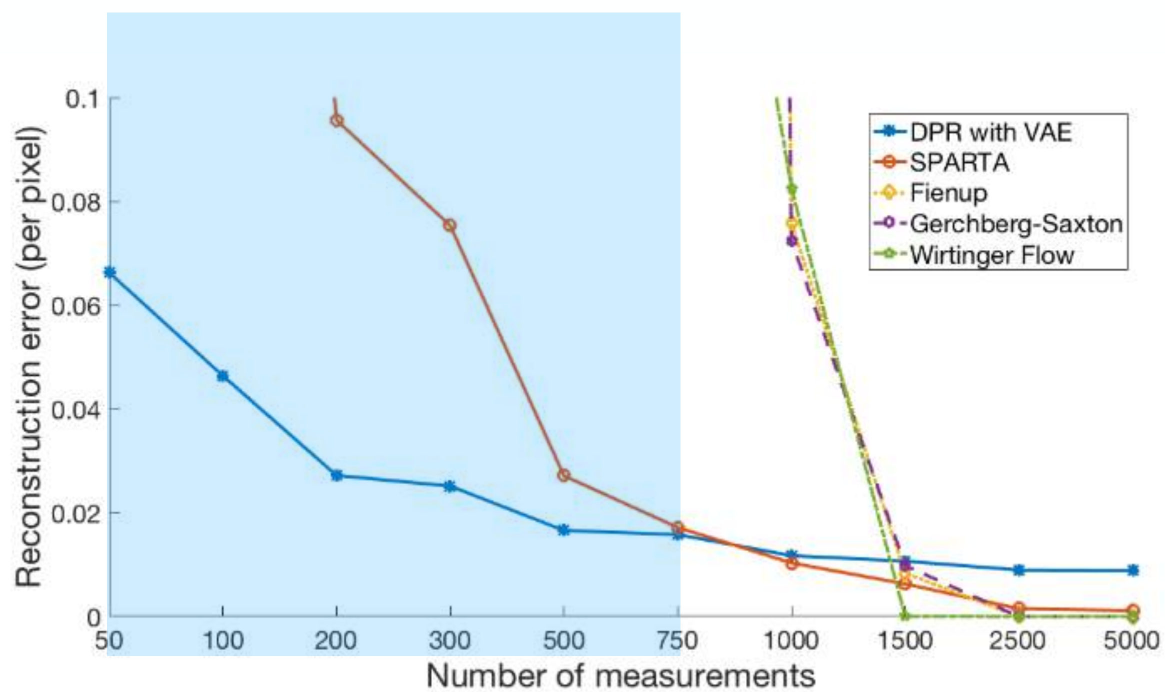




Hand, P., Leong, O. and Voroninski, V., 2018. Phase retrieval under a generative prior. *Advances in Neural Information Processing Systems*, *31*.

DGM in Phase Retrieval



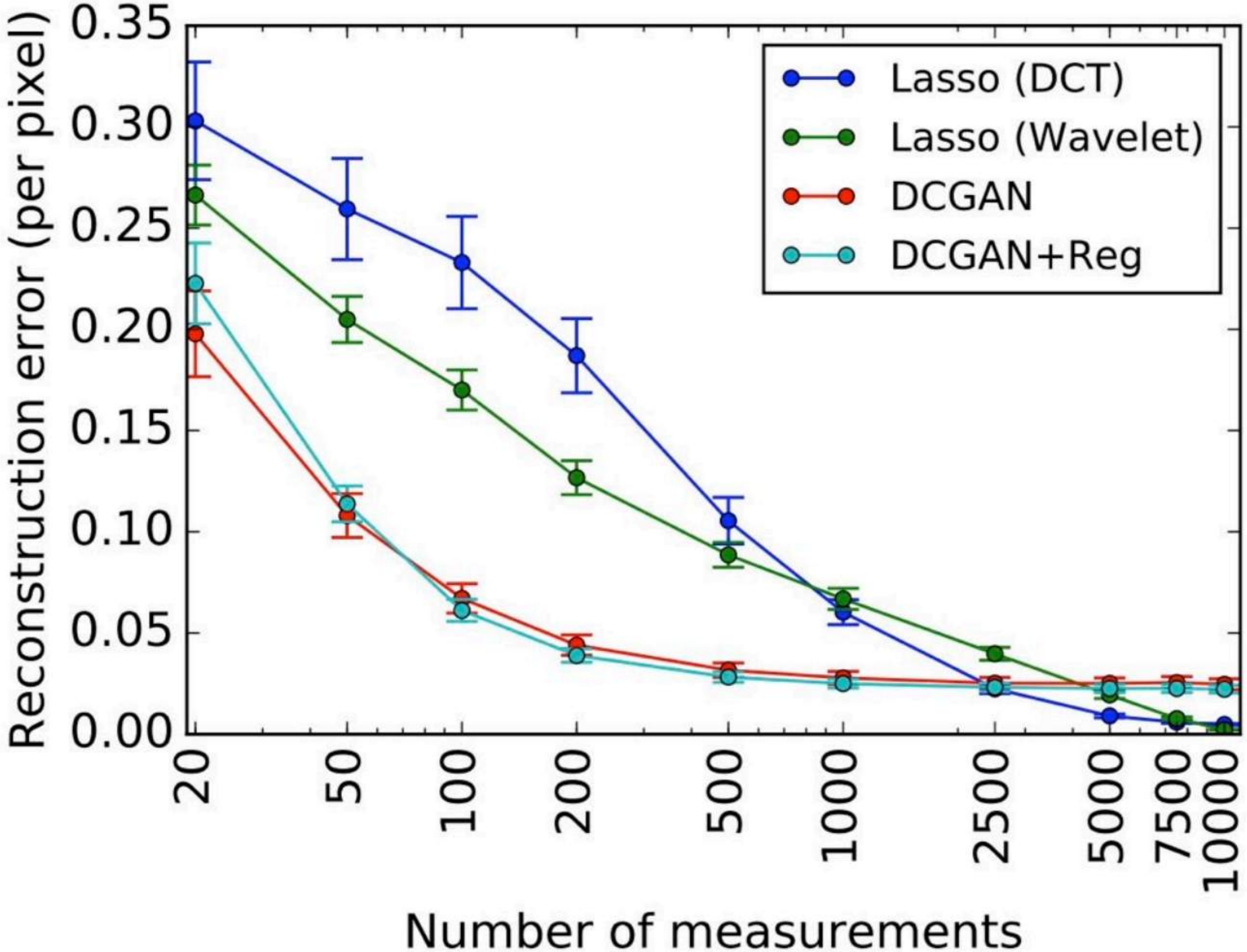


Best performance with low # of observations

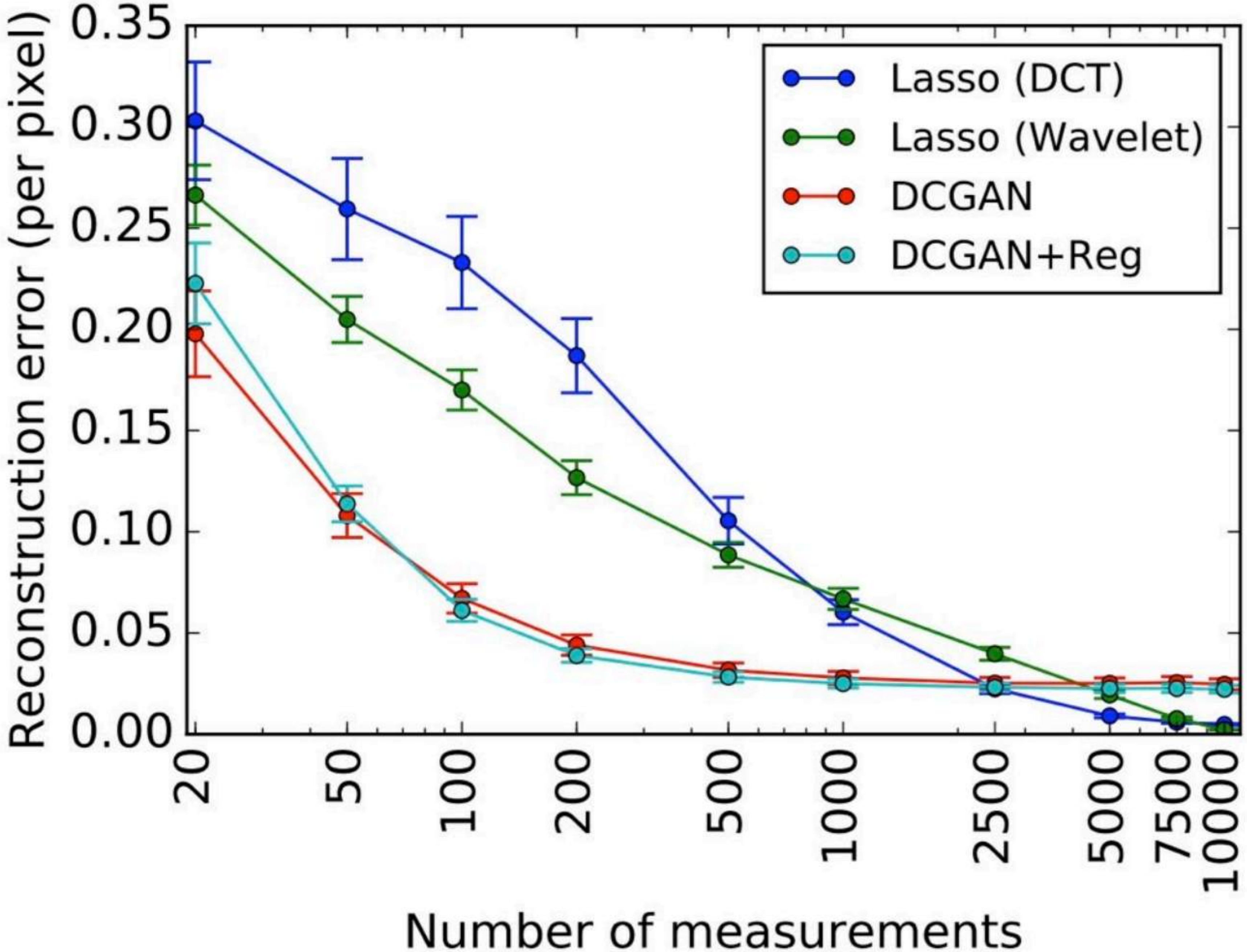
Hand, P., Leong, O. and Voroninski, V., 2018. Phase retrieval under a generative prior. *Advances in Neural Information Processing Systems*, *31*.

Existing Barrier to using DGM in Inverse Problems

Existing Barrier to using DGM in Inverse Problems



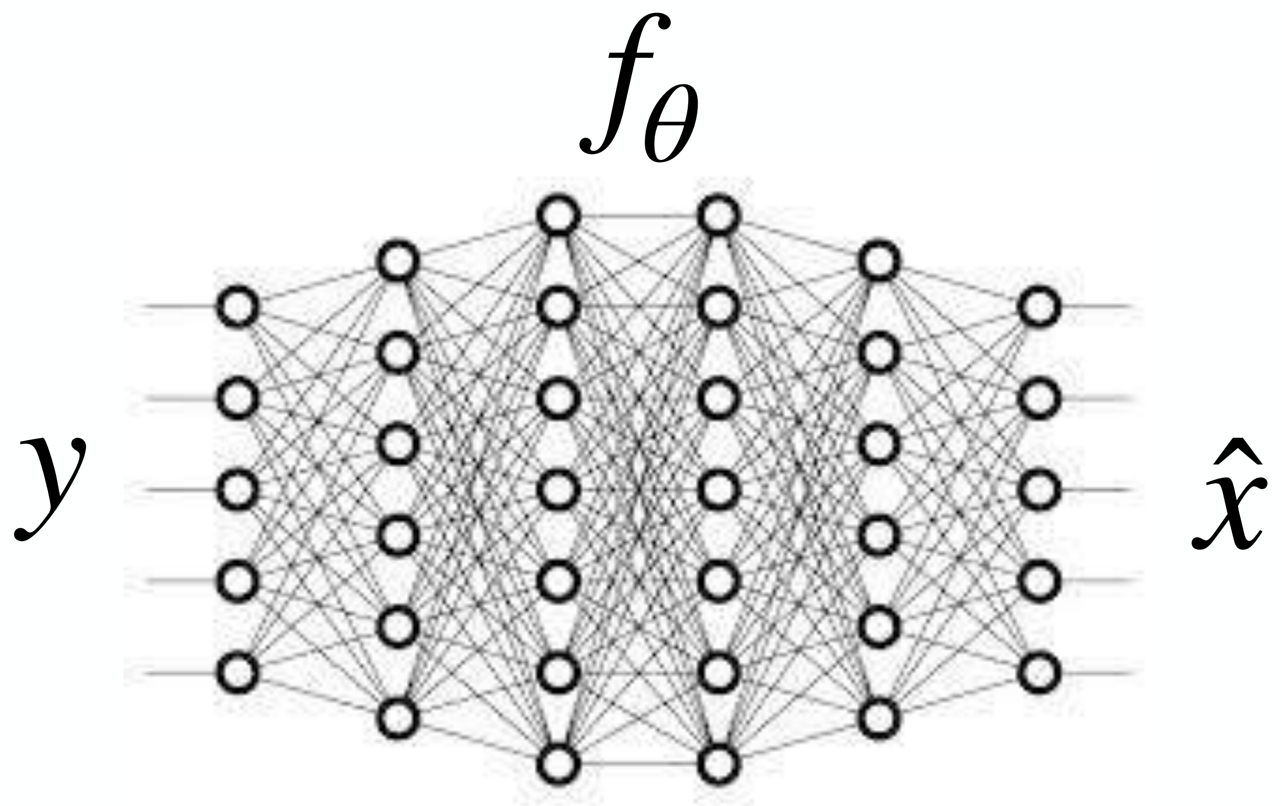
Existing Barrier to using DGM in Inverse Problems



Compressed sensing using generative models, Bora et al. ICML 2017

End-to-end methods

End-to-end methods



End-to-end methods

- We want to solve the inverse problem

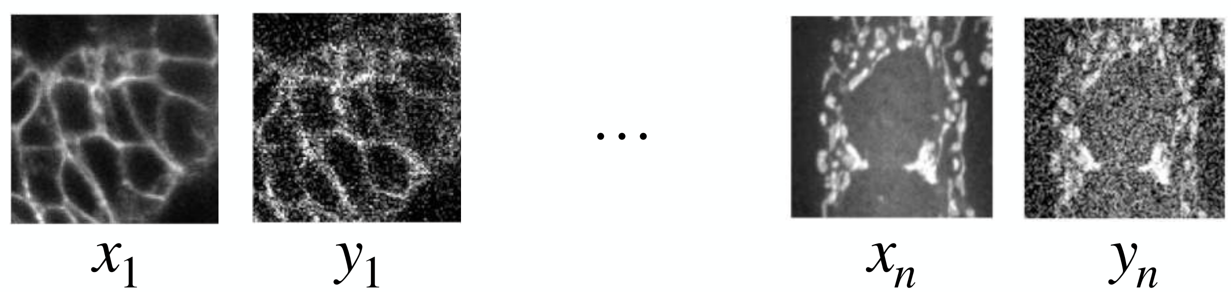
$$y = \mathscr{A}(x) + \varepsilon$$

End-to-end methods

- We want to solve the inverse problem

$$y = \mathscr{A}(x) + \varepsilon$$

- We have access to a large number of supervised data pairs (x, y)

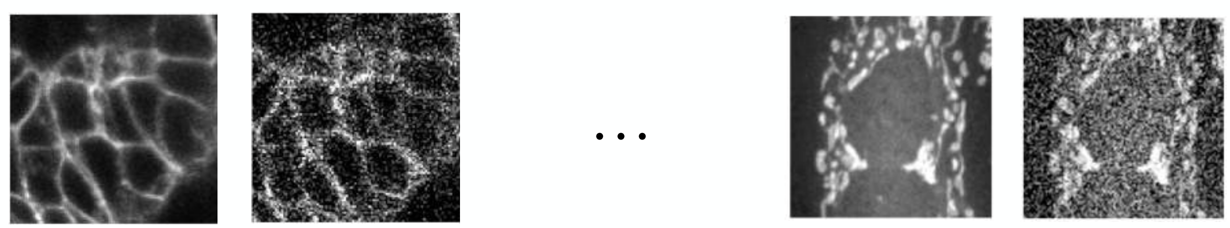


End-to-end methods

- We want to solve the inverse problem

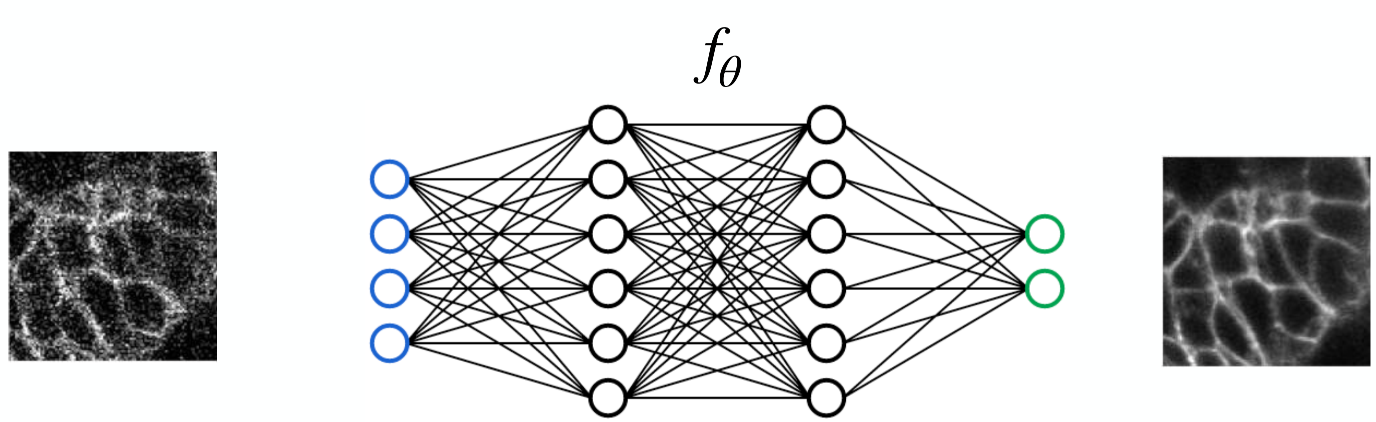
$$y = \mathscr{A}(x) + \varepsilon$$

- We have access to a large number of supervised data pairs (x, y)



x_1 y_1 $\dots$ x_n y_n

- Directly learn the mapping from observations to reconstructions from data



$$\min_\theta \mathscr{L}(f_\theta(y), x)$$

End-to-end methods in MRI reconstruction

- fastMRI public leaderboard

AIRS-Net 11/5/2020	8x	0.0070	0.9022	38.1	✓
HUMUS-Net 3/3/2022	8x	0.0081	0.8945	37.3	✓
HUMUS-Net 3/4/2022	8x	0.0086	0.8936	37.0	✓
fastMRI Repo End-to-End VarNet 11/11/2020	8x	0.0085	0.8920	37.1	✓
SubtleMR 6/23/2020	8x	0.0085	0.8919	37.1	✓
Deneme4 10/7/2021	8x	0.0085	0.8919	37.1	✓

End-to-end methods in MRI reconstruction

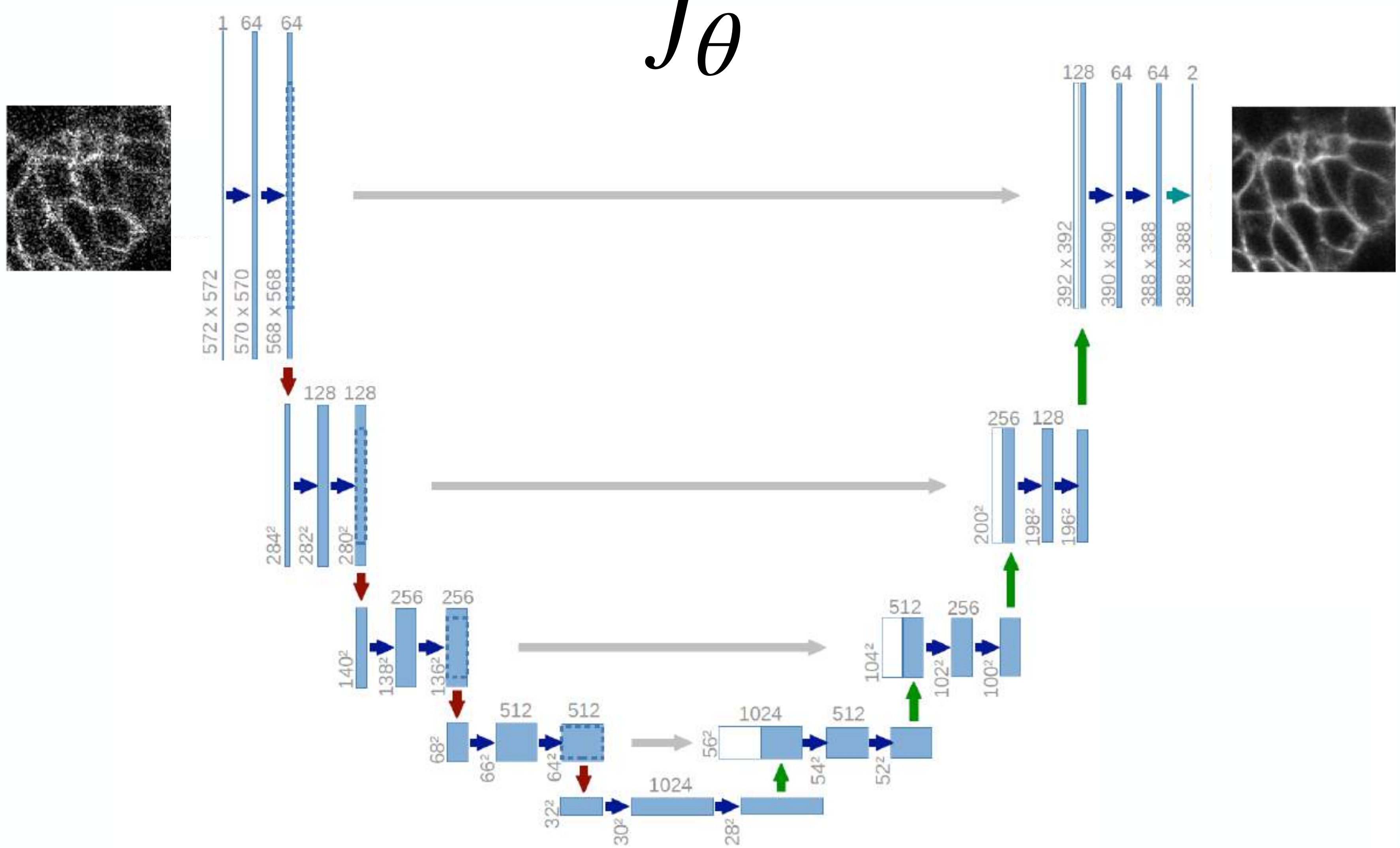
- fastMRI public leaderboard

AIRS-Net 11/5/2020	8x	0.0070	0.9022	38.1	✓
HUMUS-Net 3/3/2022	8x	0.0081	0.8945	37.3	✓
HUMUS-Net 3/4/2022	8x	0.0086	0.8936	37.0	✓
fastMRI Repo End-to-End VarNet 11/11/2020	8x	0.0085	0.8920	37.1	✓
SubtleMR 6/23/2020	8x	0.0085	0.8919	37.1	✓
Deneme4 10/7/2021	8x	0.0085	0.8919	37.1	✓

All end-to-end methods!

U-Net

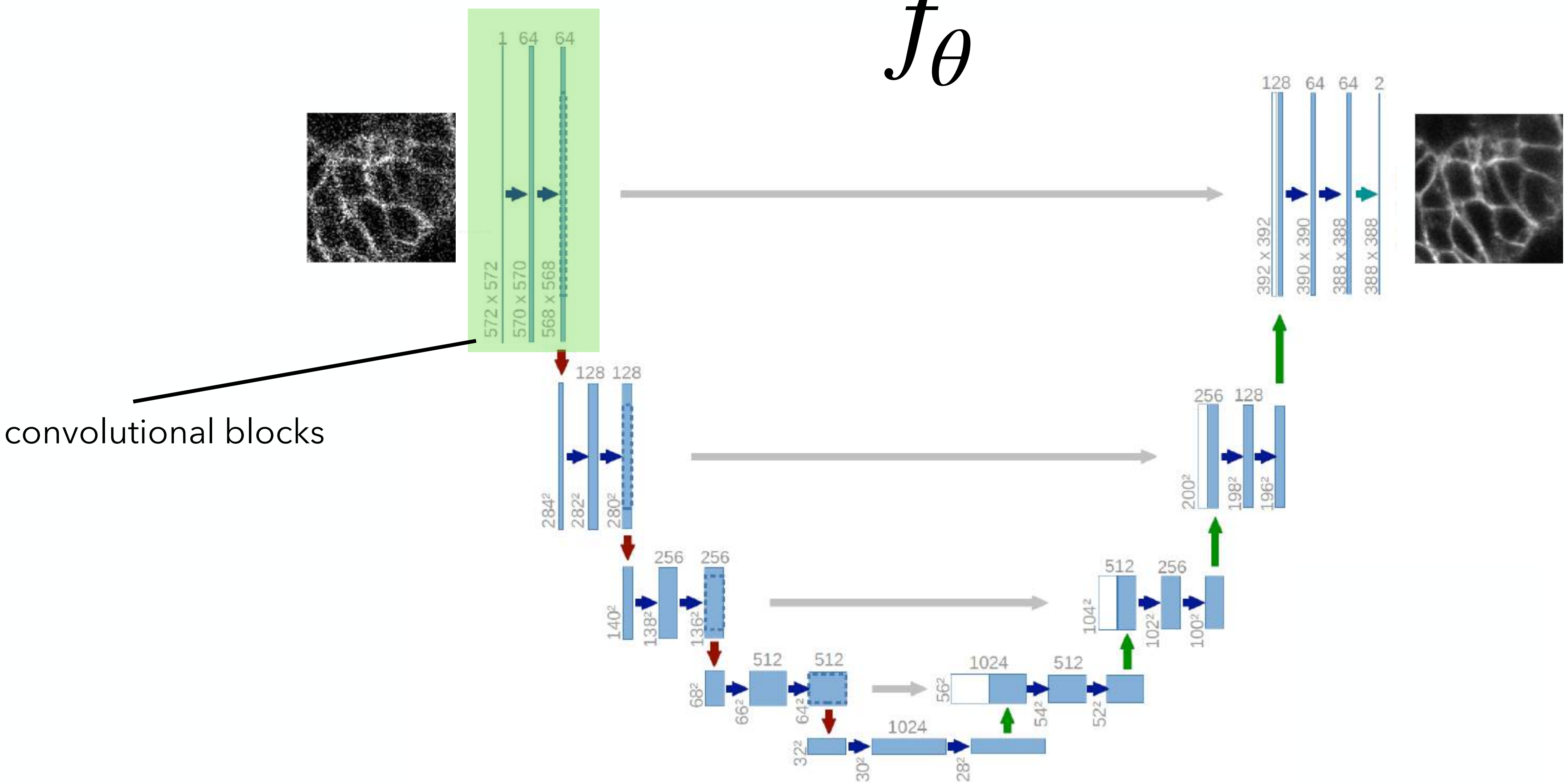
f_θ



Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

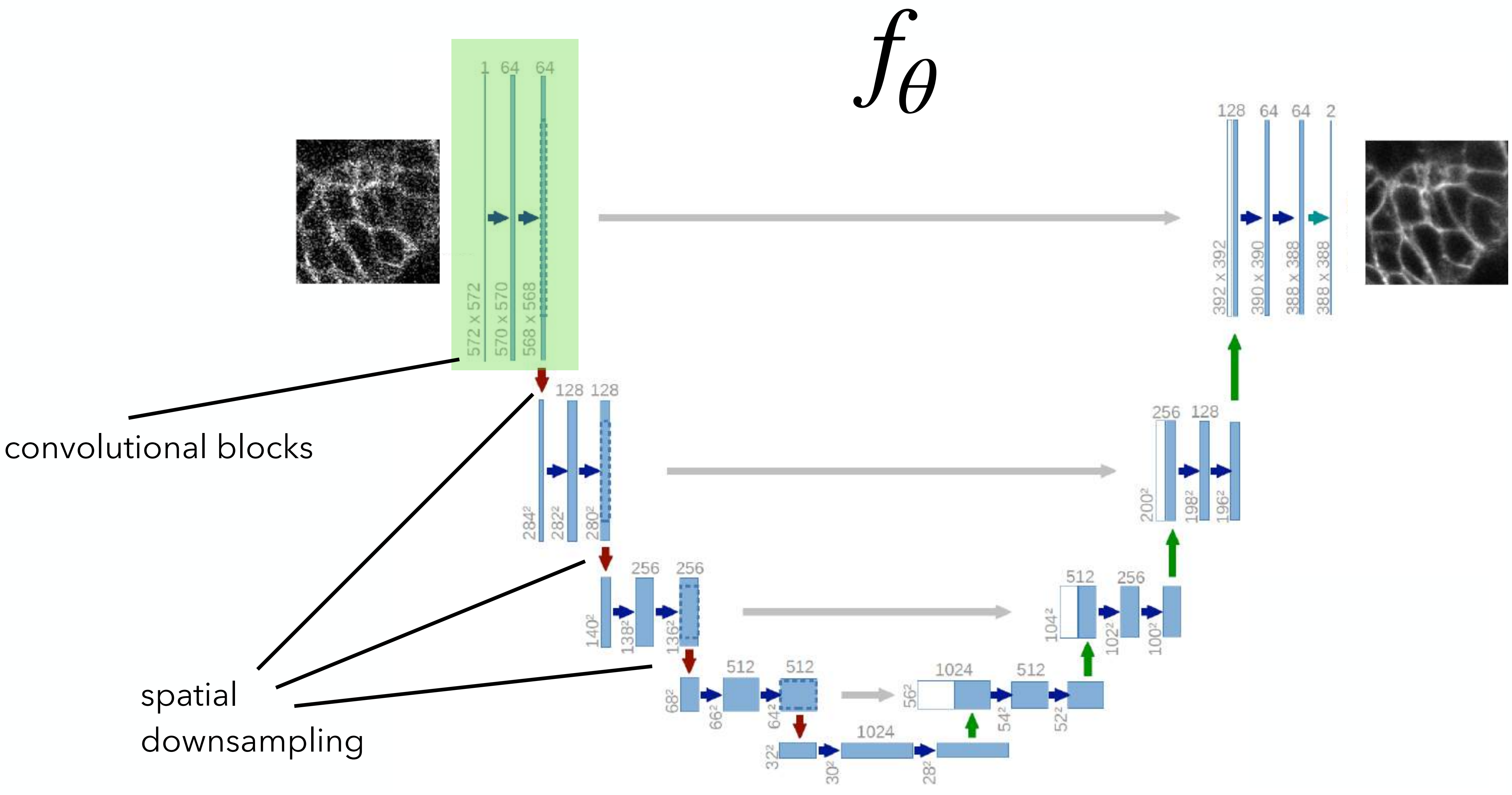
U-Net

f_θ



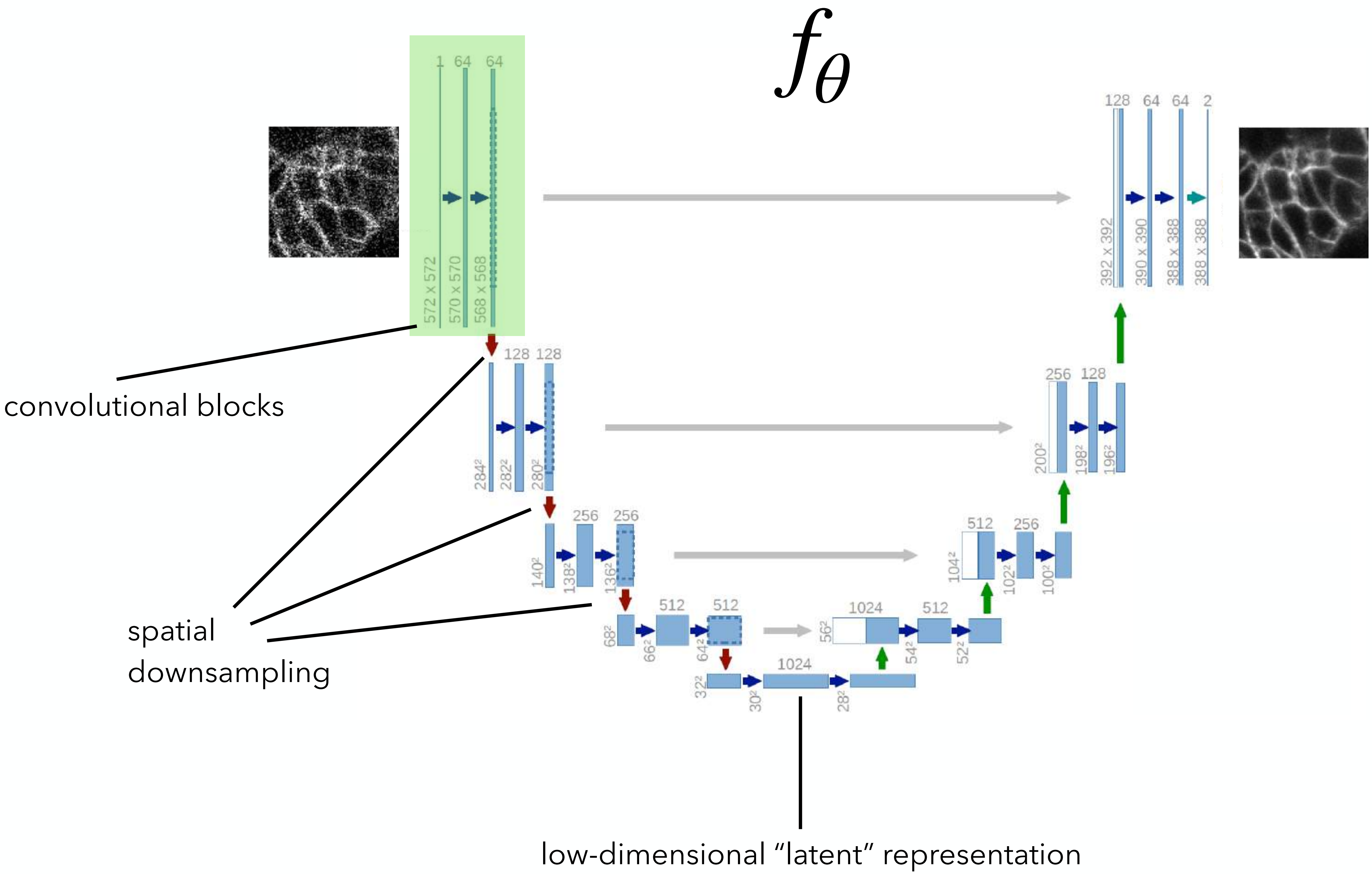
Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

U-Net



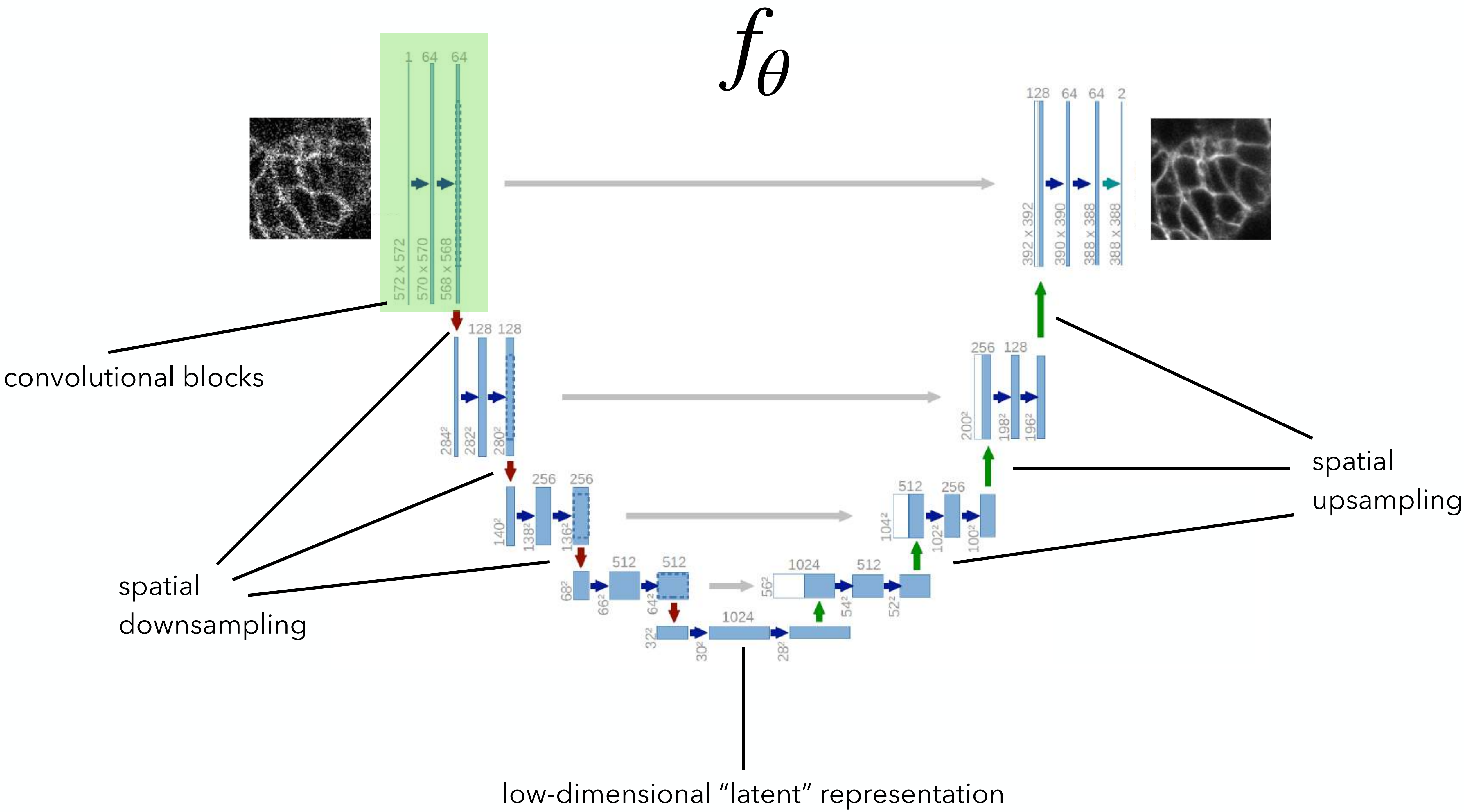
Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

U-Net



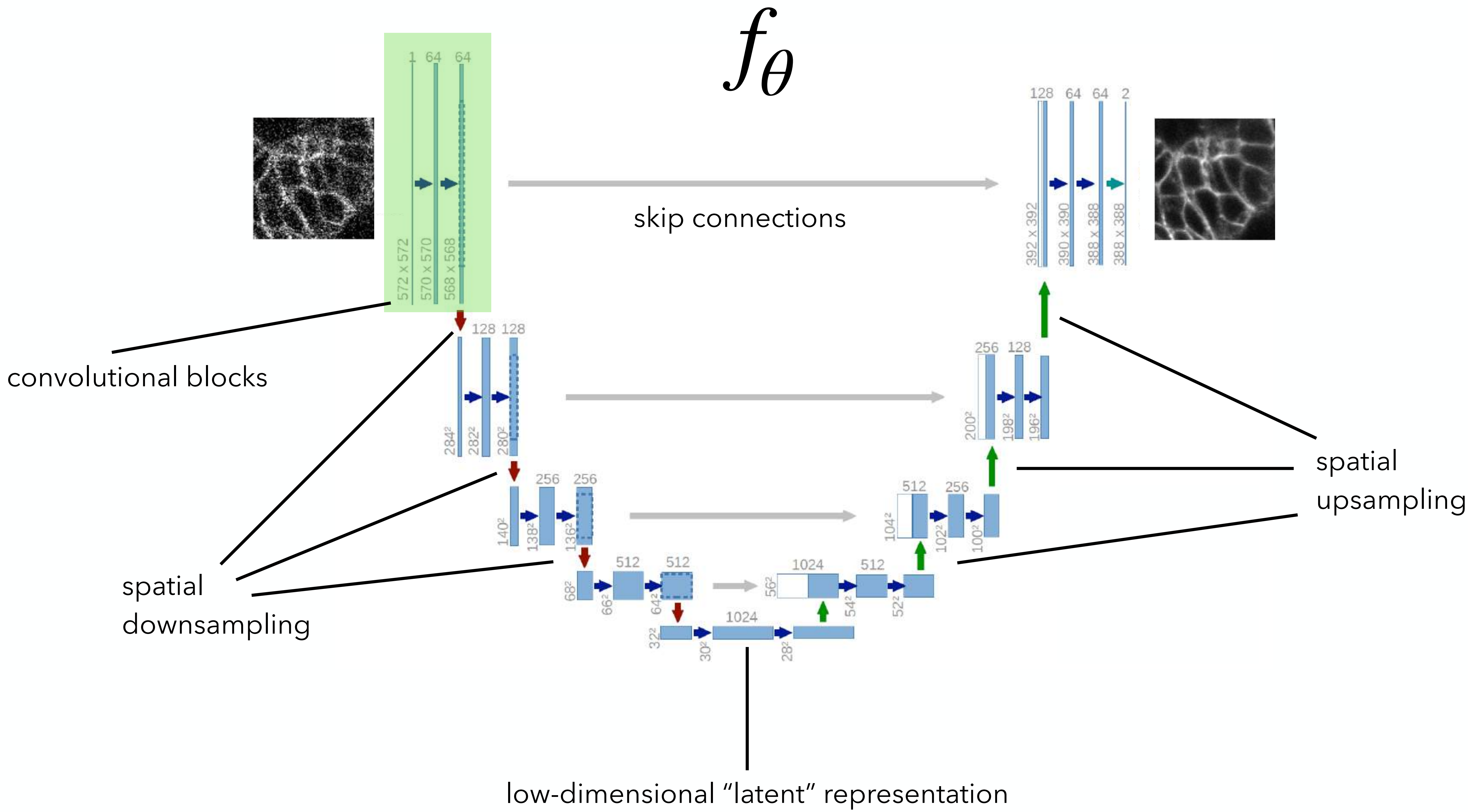
Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

U-Net



Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

U-Net



Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

Unrolled Networks

- Inverse problem formulation

$$\hat{x} = arg\,\min_{x} \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \nabla \mathscr{R}(x^t)\right]$$

Unrolled Networks

- Inverse problem formulation

$$\hat{x} = arg \min_x \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \nabla \mathscr{R}(x^t)\right]$$

What is the best regularizer?

Unrolled Networks

- Inverse problem formulation

$$\hat{x} = arg\min_{x} \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right]$$

Parameterize regularizer gradient as NN!

Unrolled Networks

- Inverse problem formulation

$$\hat{x} = arg \min_{x} \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right]$$

Parameterize regularizer gradient as NN!

$$\xrightarrow{\quad x_0 \quad}$$

Unrolled Networks

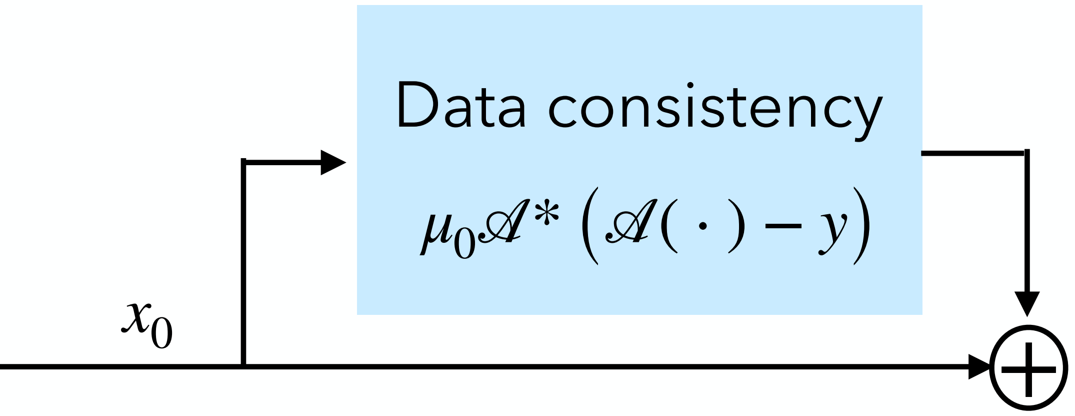
- Inverse problem formulation

$$\hat{x} = arg\min_x \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right]$$

Parameterize regularizer gradient as NN!



Unrolled Networks

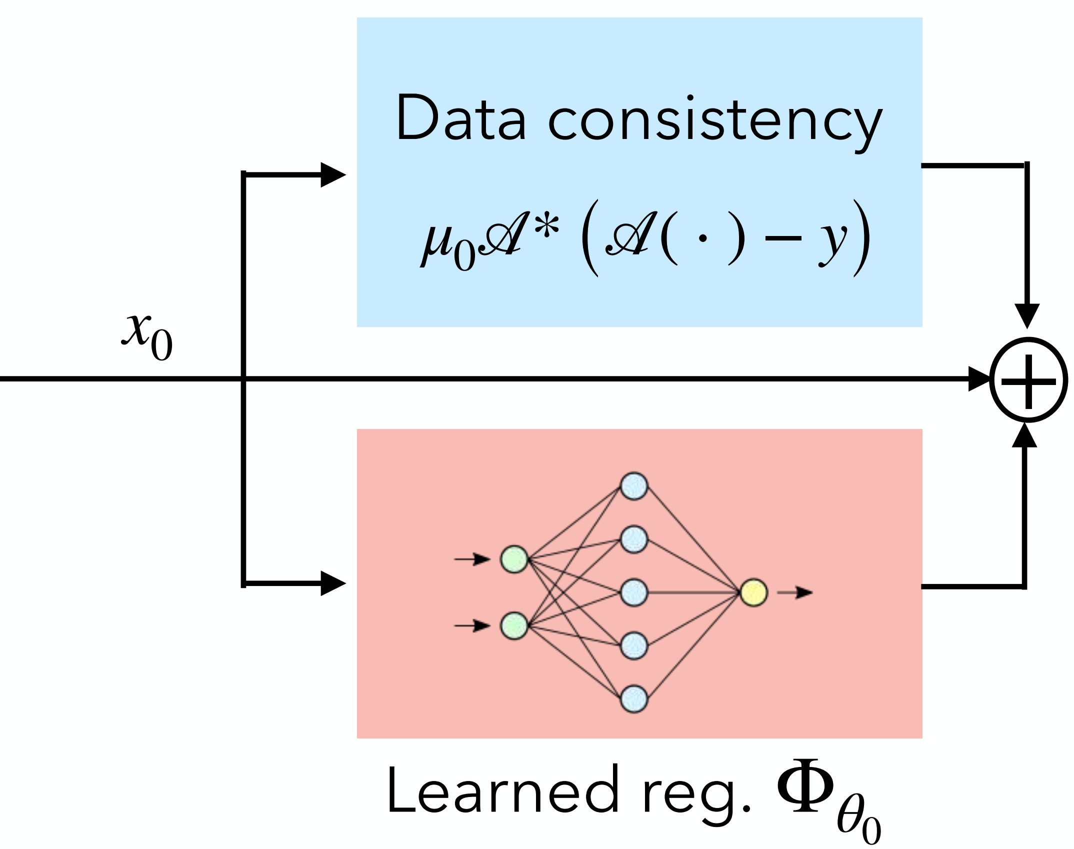
- Inverse problem formulation

$$\hat{x} = arg \min_x \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right]$$

Parameterize regularizer gradient as NN!



Unrolled Networks

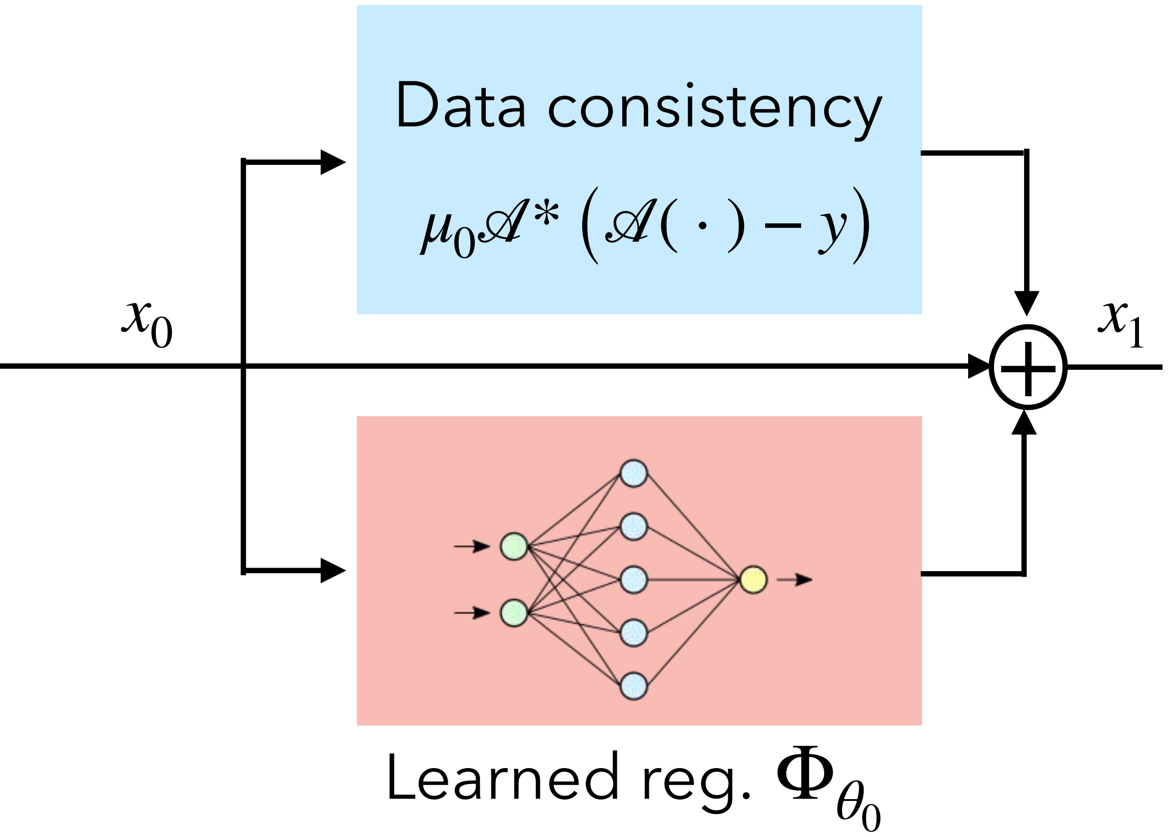
- Inverse problem formulation

$$\hat{x} = arg\min_x \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right]$$

Parameterize regularizer gradient as NN!



Unrolled Networks

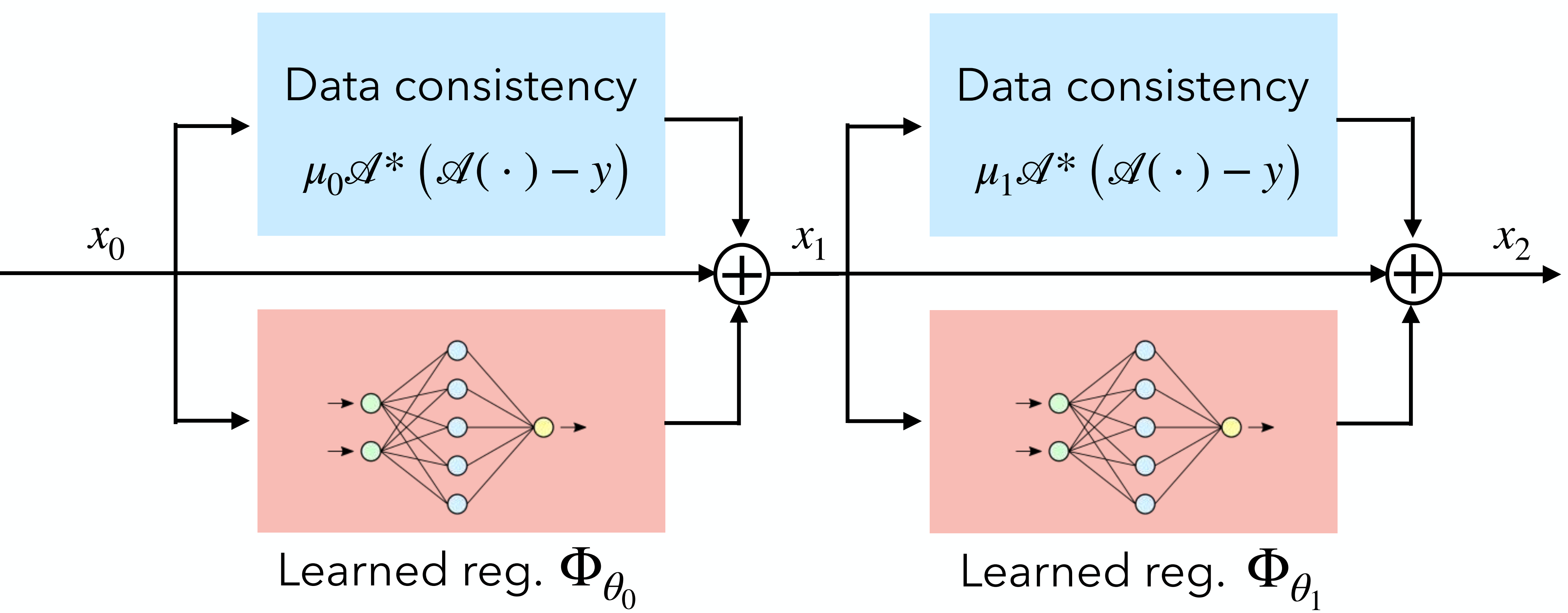
- Inverse problem formulation

$$\hat{x} = arg \min_x \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right]$$

Parameterize regularizer gradient as NN!



Unrolled Networks

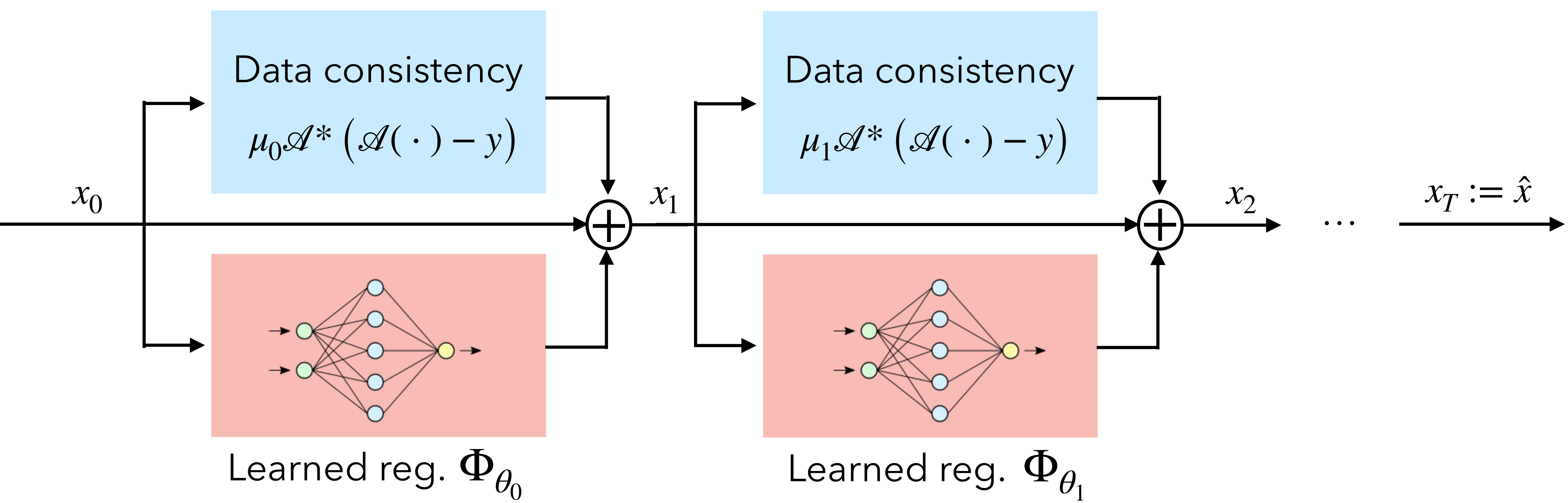
- Inverse problem formulation

$$\hat{x} = arg \min_x \|\mathscr{A}(x) - y\|^2 + \mathscr{R}(x)$$

- Iterative solution via GD

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right]$$

Parameterize regularizer gradient as NN!



E2E VarNet

- Unroll GD iterations in k-space

Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

E2E VarNet

- Unroll GD iterations in k-space

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y \right) + \Phi_\theta(x^t) \right]$$

Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

E2E VarNet

- Unroll GD iterations in k-space

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y \right) + \Phi_\theta(x^t) \right] \quad \xrightarrow{\mathscr{F}}$$

Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

E2E VarNet

- Unroll GD iterations in k-space

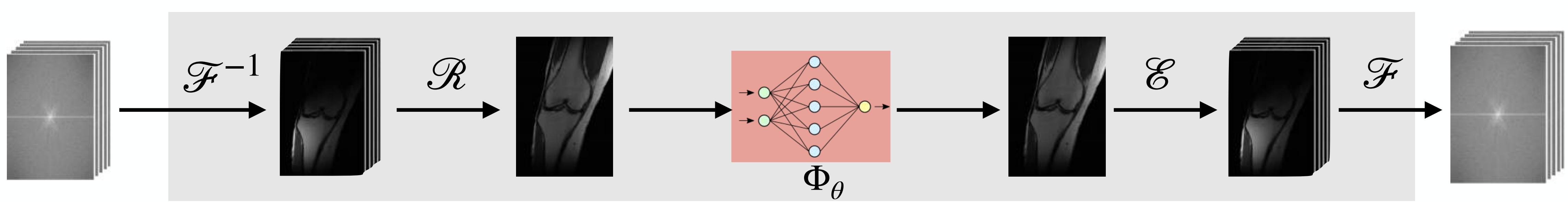
$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y \right) + \Phi_\theta(x^t) \right] \xrightarrow{\mathscr{F}} k^{t+1} = k^t - \mu^t M(k^t - \tilde{k}) + \Psi(k^t)$$

Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

E2E VarNet

- Unroll GD iterations in k-space

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right] \xrightarrow{\mathscr{F}} k^{t+1} = k^t - \mu^t M(k^t - \tilde{k}) + \Psi(k^t)$$

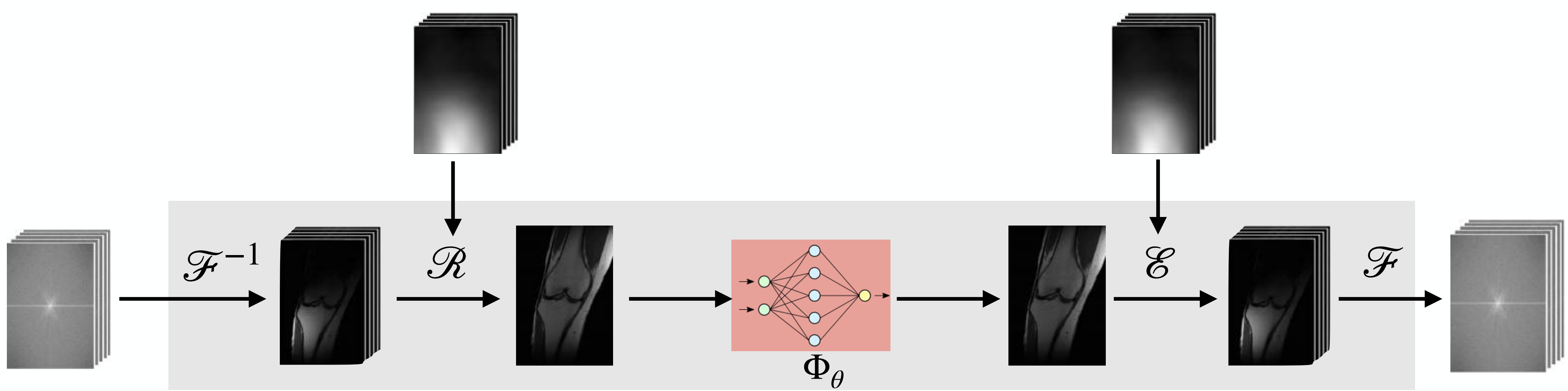


Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

E2E VarNet

- Unroll GD iterations in k-space

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right] \xrightarrow{\mathscr{F}} k^{t+1} = k^t - \mu^t M(k^t - \tilde{k}) + \Psi(k^t)$$

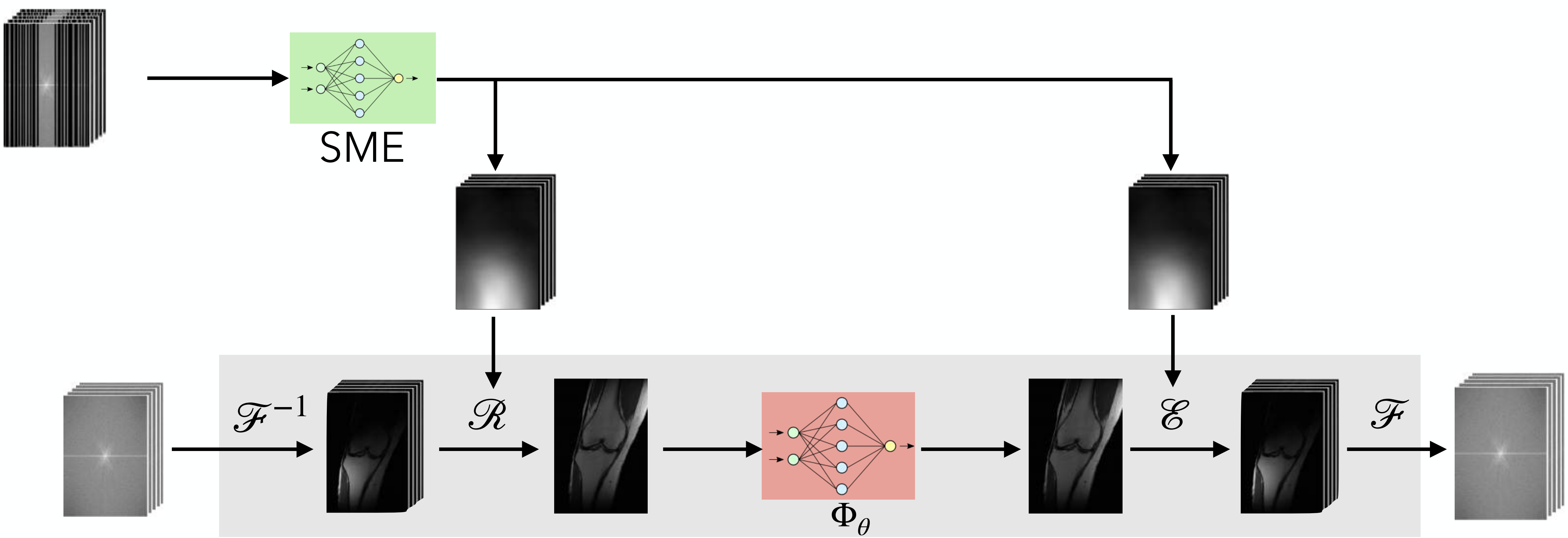


Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

E2E VarNet

- Unroll GD iterations in k-space

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y \right) + \Phi_\theta(x^t) \right] \xrightarrow{\mathscr{F}} k^{t+1} = k^t - \mu^t M(k^t - \tilde{k}) + \Psi(k^t)$$

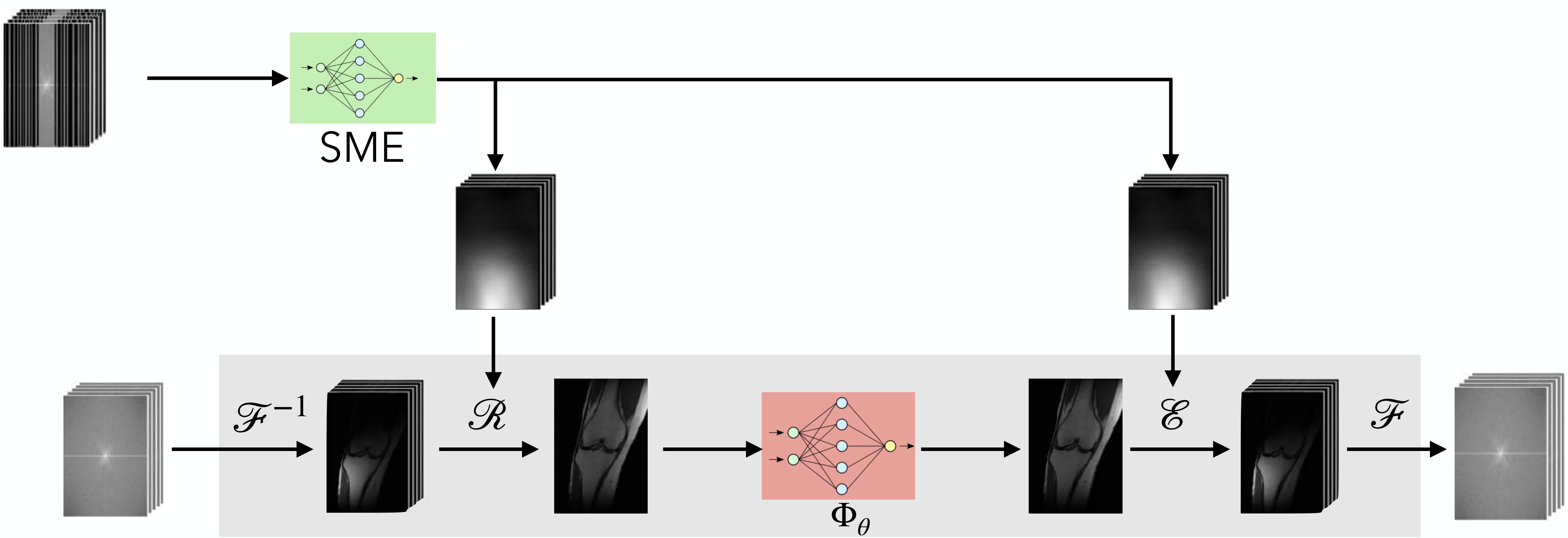


Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

E2E VarNet

- Unroll GD iterations in k-space

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y\right) + \Phi_\theta(x^t)\right] \xrightarrow{\mathscr{F}} k^{t+1} = k^t - \mu^t M(k^t - \tilde{k}) + \Psi(k^t)$$



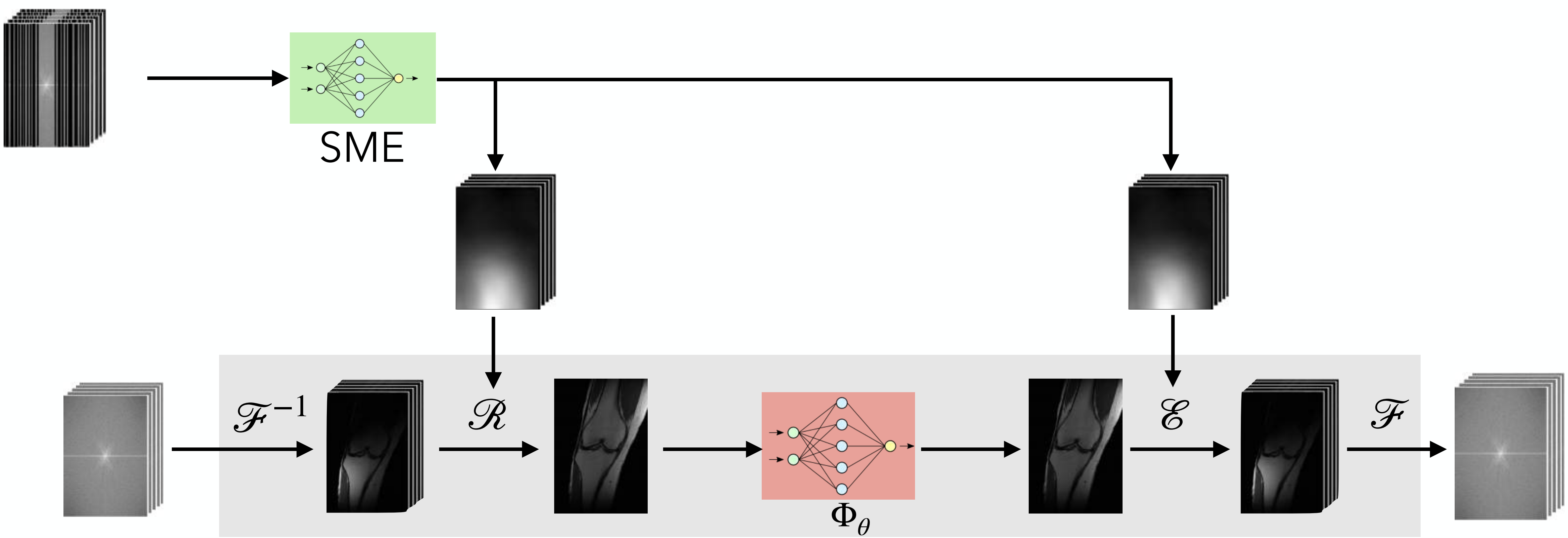
- Denoiser Φ_θ is a U-Net

Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

E2E VarNet

- Unroll GD iterations in k-space

$$x^{t+1} = x^t - \mu^t \left[\mathscr{A}^* \left(\mathscr{A}(x^t) - y \right) + \Phi_\theta(x^t) \right] \xrightarrow{\mathscr{F}} k^{t+1} = k^t - \mu^t M(k^t - \tilde{k}) + \Psi(k^t)$$



- Denoiser Φ_θ is a U-Net

Can we do better with modern architectures?

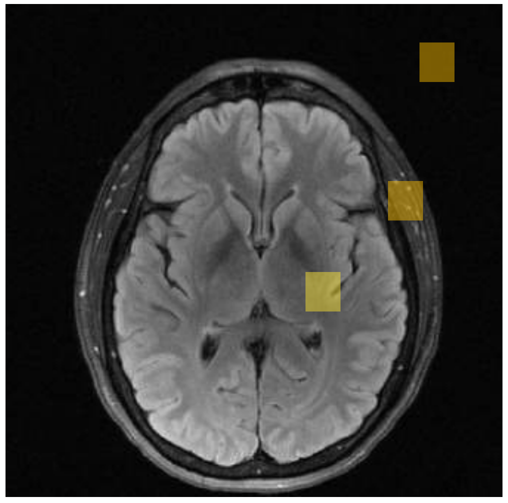
Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., ... & Johnson, P. (2020). End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 64-73). Springer, Cham.

Transformers?

- Benefits of Transformers

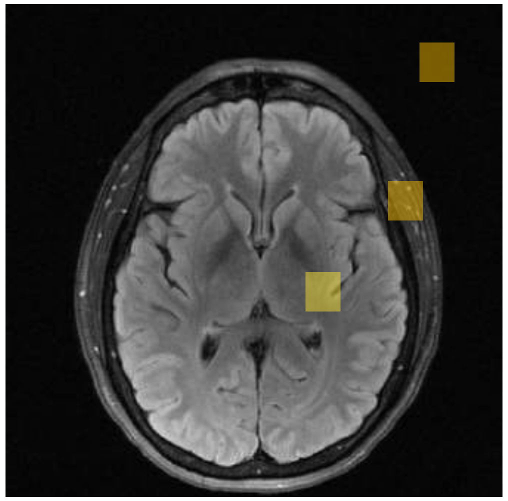
Transformers?

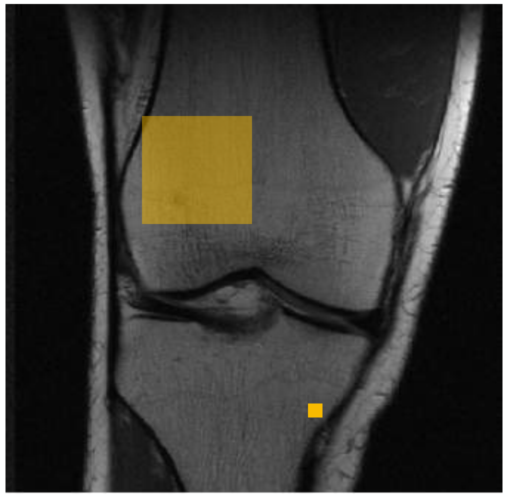
- Benefits of Transformers
  - conv kernels are content-independent



Transformers?

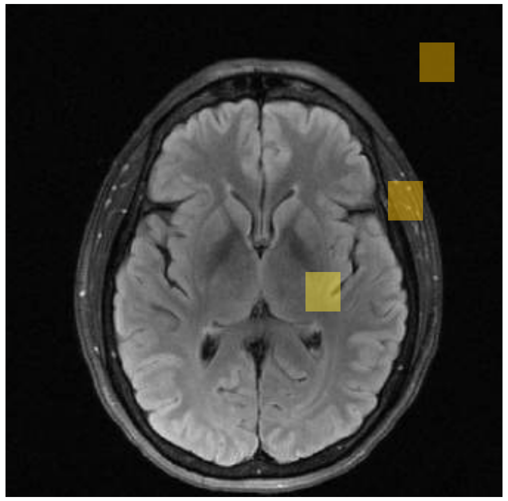
- Benefits of Transformers
  - conv kernels are content-independent
  - conv is not efficient for long-range dependency modelling

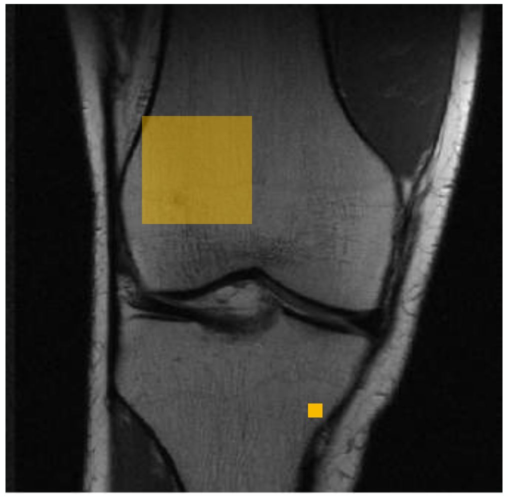




Transformers?

- Benefits of Transformers
    - conv kernels are content-independent
    - conv is not efficient for long-range dependency modelling

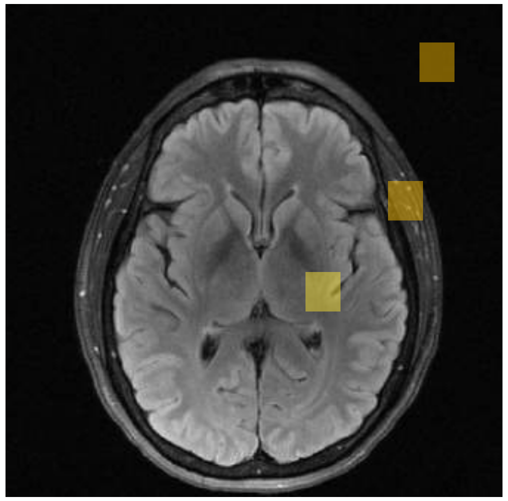


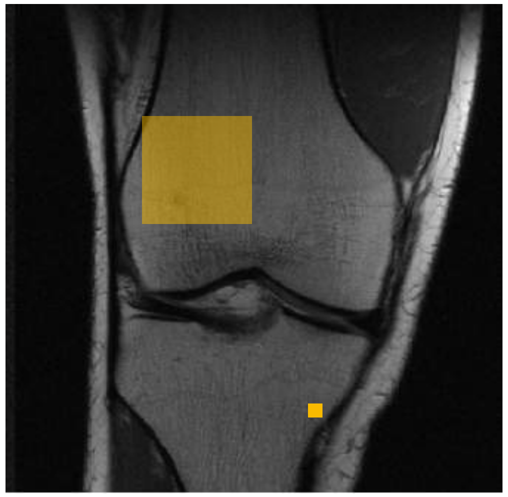


- Self-attention mechanism

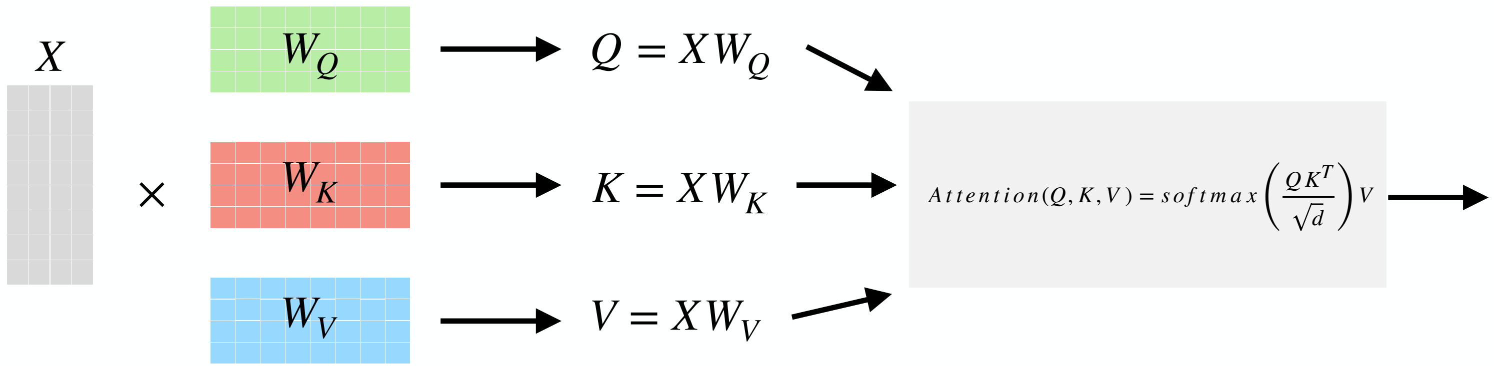
Transformers?

- Benefits of Transformers
  - conv kernels are content-independent
  - conv is not efficient for long-range dependency modelling



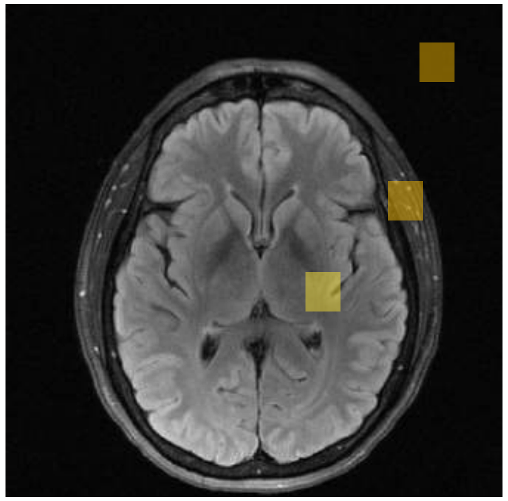


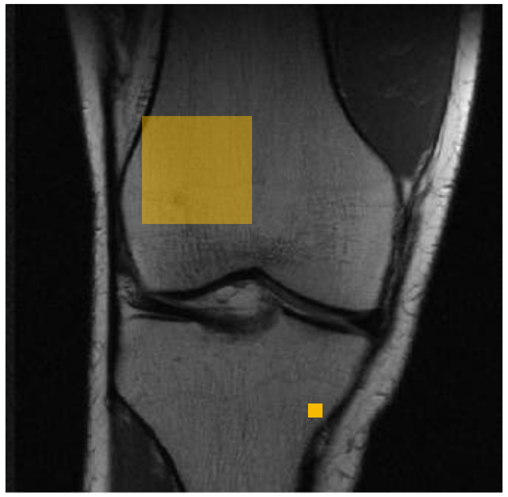
- Self-attention mechanism



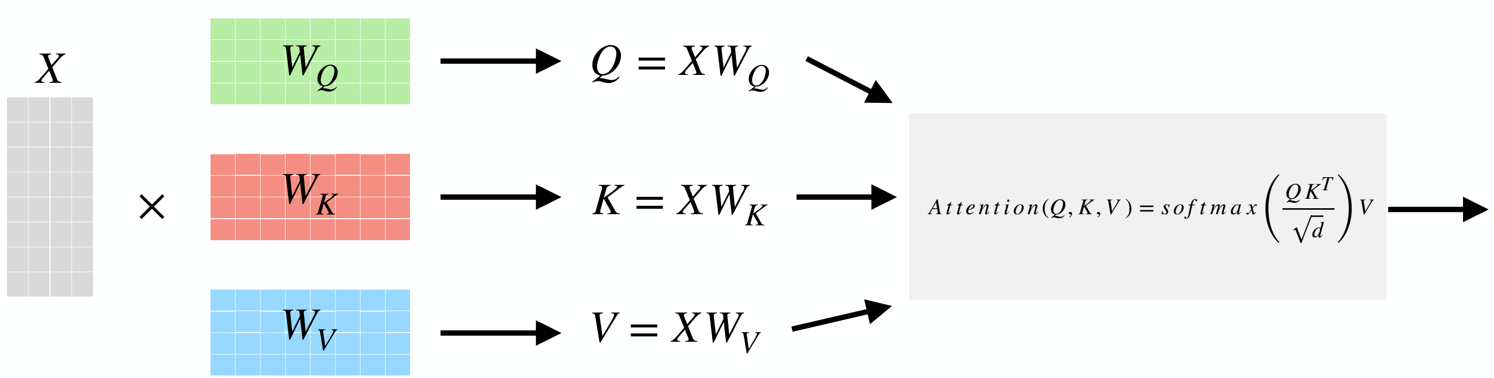
Transformers?

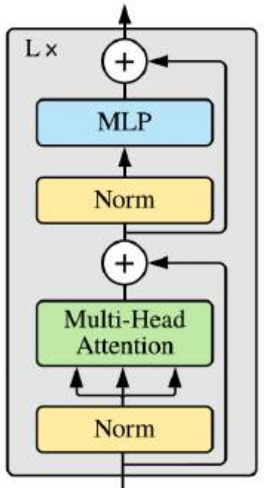
- Benefits of Transformers
  - conv kernels are content-independent
  - conv is not efficient for long-range dependency modelling





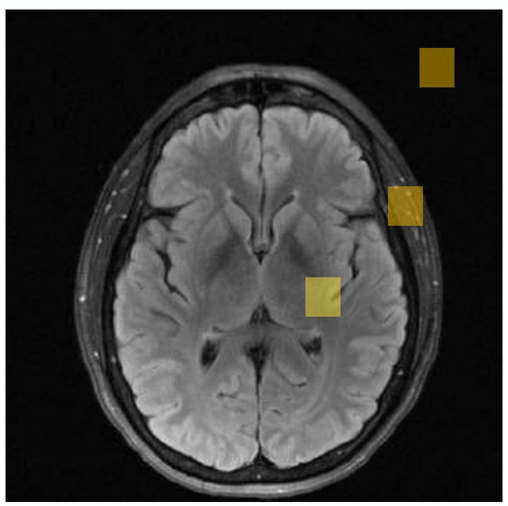
- Self-attention mechanism

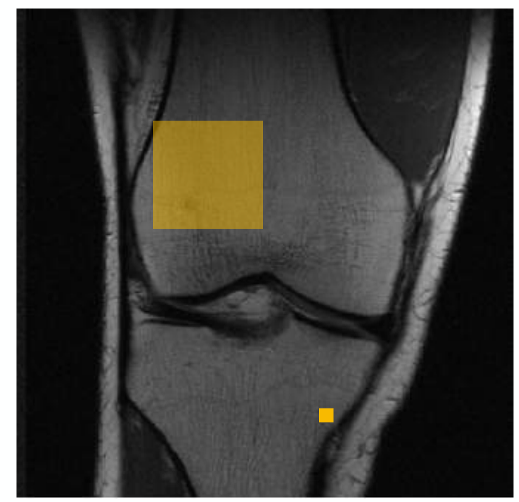




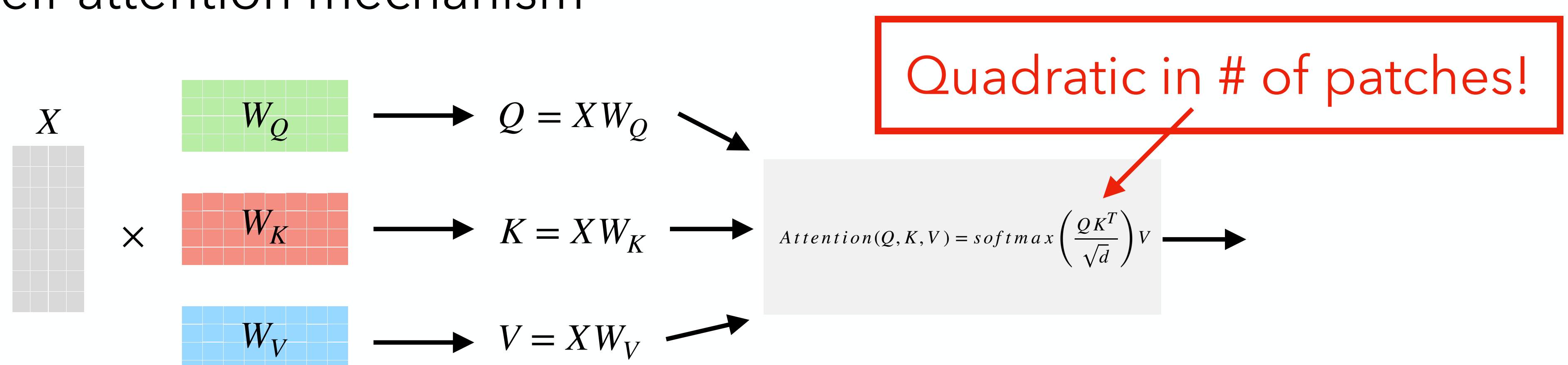
Transformers?

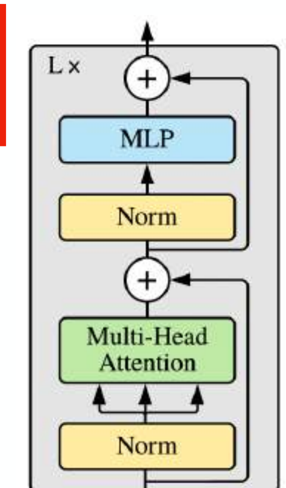
- Benefits of Transformers
  - conv kernels are content-independent
  - conv is not efficient for long-range dependency modelling





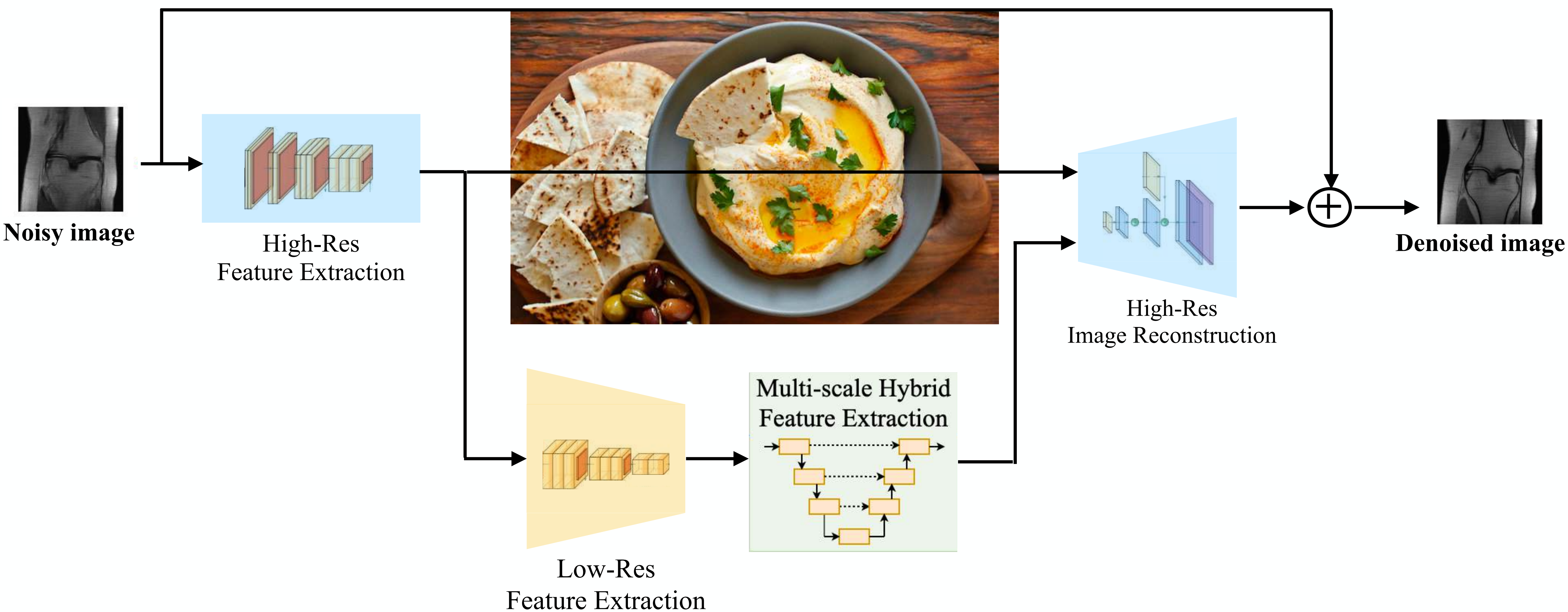
- Self-attention mechanism





HUMUS-Net

Hybrid Unrolled Multi-scale Network Architecture for Accelerated MRI Reconstruction



HUMUS-Net Experiments

fastMRI multi-coil knee, 8x acceleration

Model	Test SSIM	Test PSNR

HUMUS-Net Experiments

fastMRI multi-coil knee, 8x acceleration

Model	Test SSIM	Test PSNR
E2E-VarNet	0.8920	37.1

HUMUS-Net Experiments

fastMRI multi-coil knee, 8x acceleration

Model	Test SSIM	Test PSNR
E2E-VarNet	0.8920	37.1
HUMUS-Net	**0.8951**	**37.4**

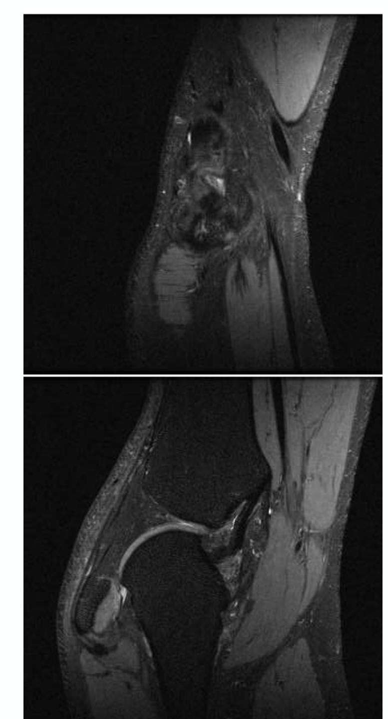
HUMUS-Net Experiments

fastMRI multi-coil knee, 8x acceleration

Model	Test SSIM	Test PSNR
E2E-VarNet	0.8920	37.1
HUMUS-Net	**0.8951**	**37.4**

Stanford3D multi-coil knee, 8x acceleration

Model	Val. SSIM	Val. PSNR



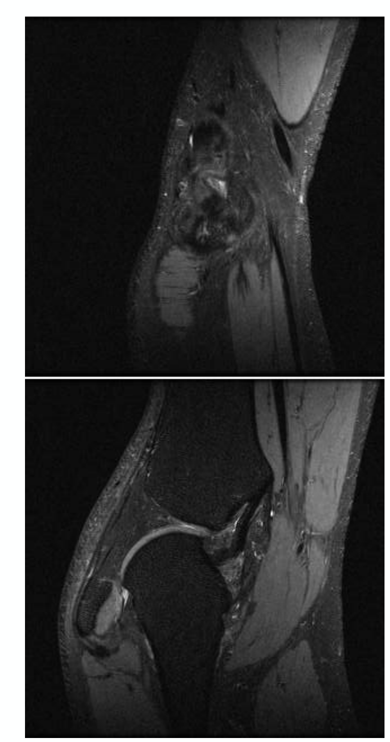
HUMUS-Net Experiments

fastMRI multi-coil knee, 8x acceleration

Model	Test SSIM	Test PSNR
E2E-VarNet	0.8920	37.1
HUMUS-Net	**0.8951**	**37.4**

Stanford3D multi-coil knee, 8x acceleration

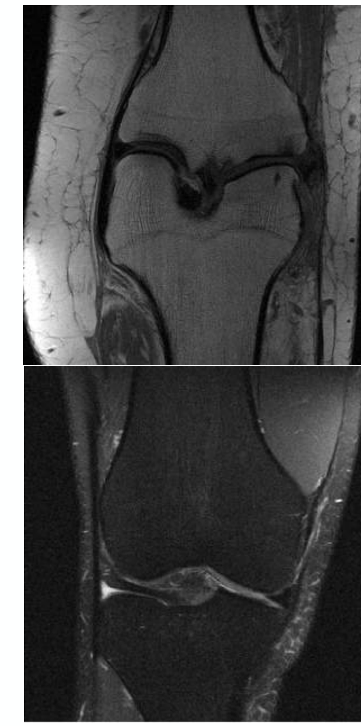
Model	Val. SSIM	Val. PSNR
E2E-VarNet	0.9432 (±0.0063)	39.99 (± 0.6144)



HUMUS-Net Experiments

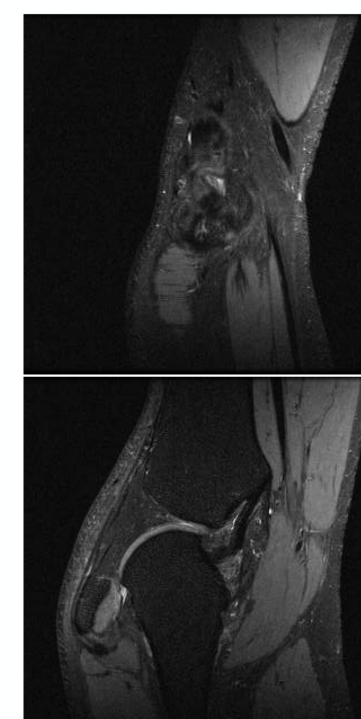
fastMRI multi-coil knee, 8x acceleration

Model	Test SSIM	Test PSNR
E2E-VarNet	0.8920	37.1
HUMUS-Net	**0.8951**	**37.4**



Stanford3D multi-coil knee, 8x acceleration

Model	Val. SSIM	Val. PSNR
E2E-VarNet	0.9432 (±0.0063)	39.99 (± 0.6144)
HUMUS-Net	**0.9453** (± 0.0065)	**40.35** (± 0.6460)

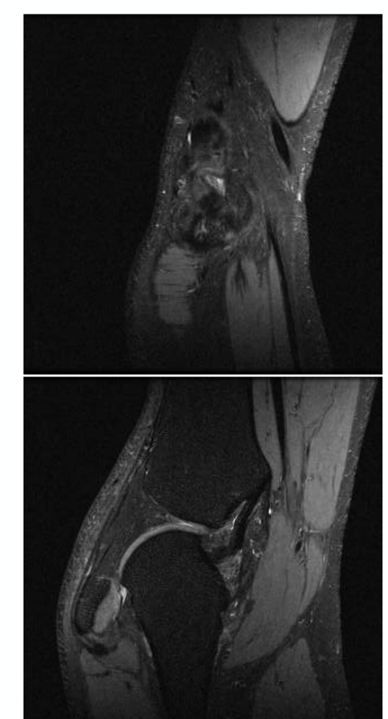


HUMUS-Net Experiments

fastMRI multi-coil knee, 8x acceleration

Model	Test SSIM	Test PSNR
E2E-VarNet	0.8920	37.1
HUMUS-Net	**0.8951**	**37.4**

HUMUS-Net 2/23/2022	8x	0.0086	0.8936
fastMRI Repo End-to-End VarNet 11/11/2020	8x	0.0085	0.8920
SubtleMR 6/23/2020	8x	0.0085	0.8919
Deneme4 10/7/2021	8x	0.0085	0.8919



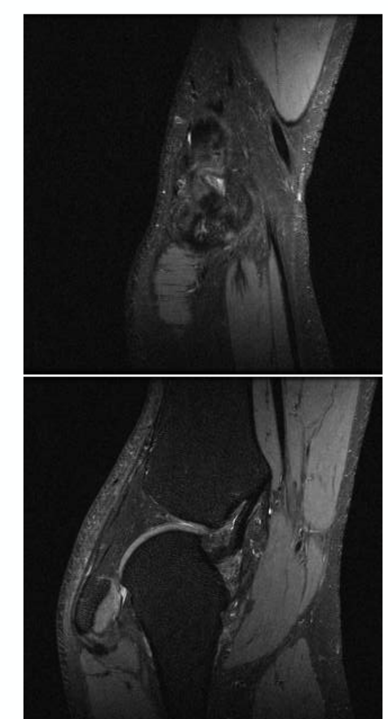
HUMUS-Net Experiments

fastMRI multi-coil knee, 8x acceleration

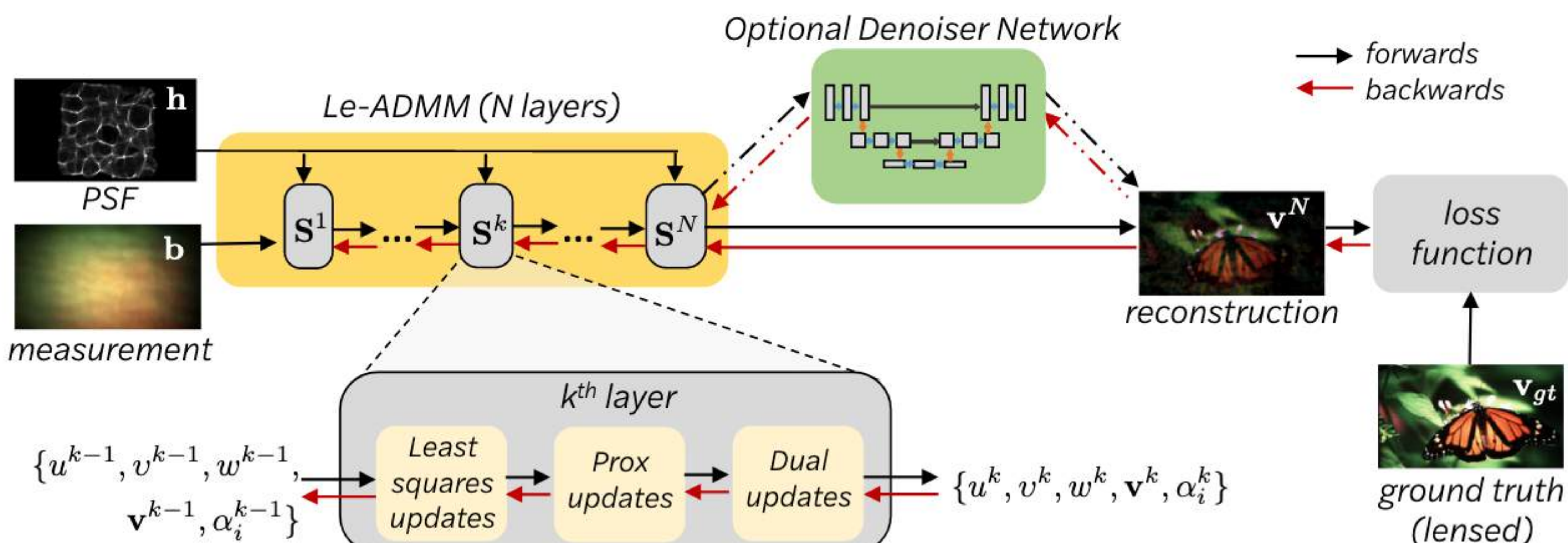
Model	Test SSIM	Test PSNR
E2E-VarNet	0.8920	37.1
HUMUS-Net	**0.8951**	**37.4**

Stanford3D multi-coil knee, 8x acceleration

Model	Val. SSIM	Val. PSNR
E2E-VarNet	0.9432 (±0.0063)	39.99 (± 0.6144)
HUMUS-Net	**0.9453** (± 0.0065)	**40.35** (± 0.6460)



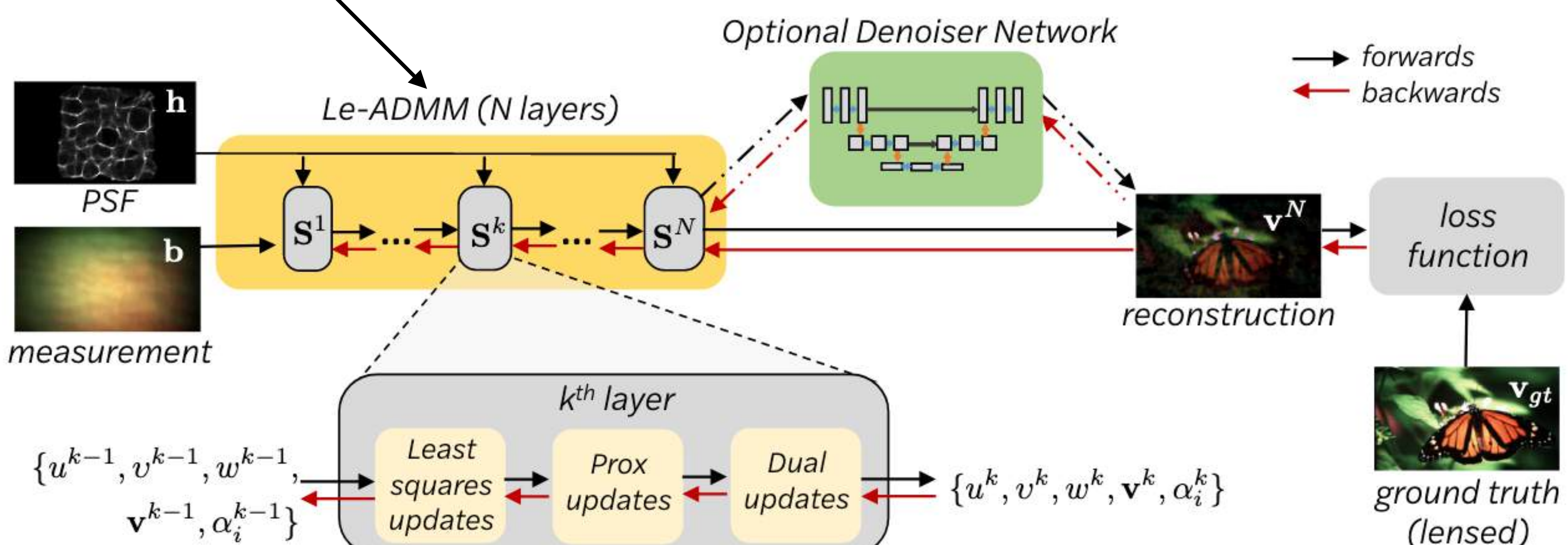
Deep Unrolling in lensless imaging



Monakhova, Kristina, et al. "Learned reconstructions for practical mask-based lensless imaging." *Optics express* 27.20 (2019): 28075-28090.

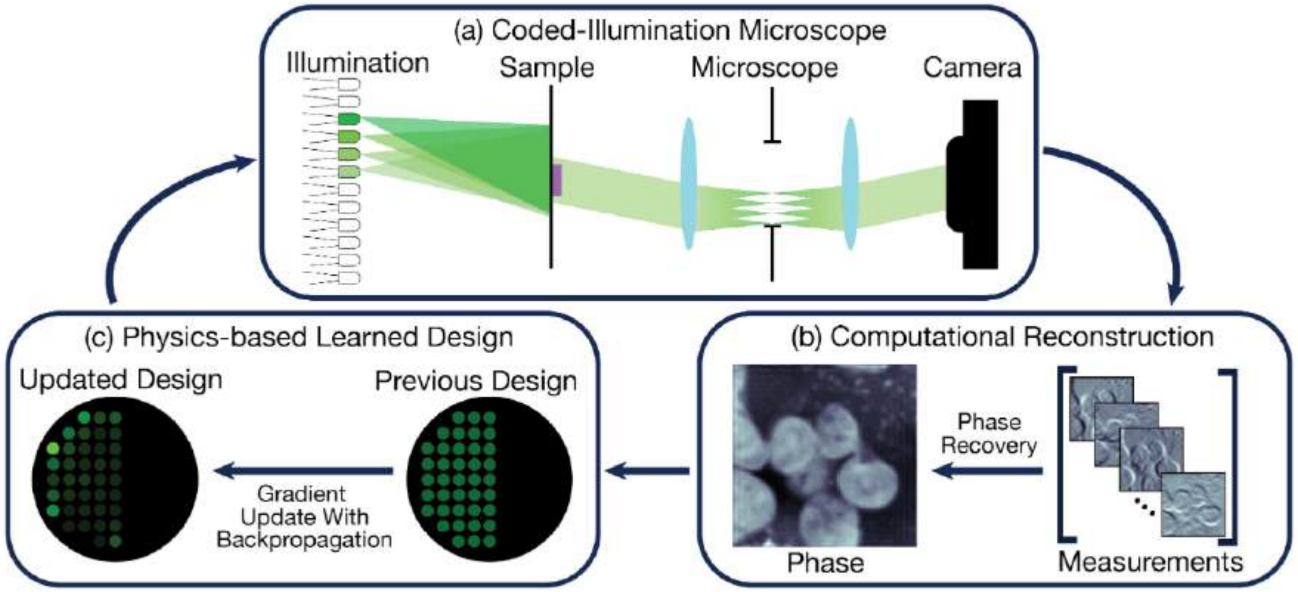
Deep Unrolling in lensless imaging

Learned (unrolled) ADMM



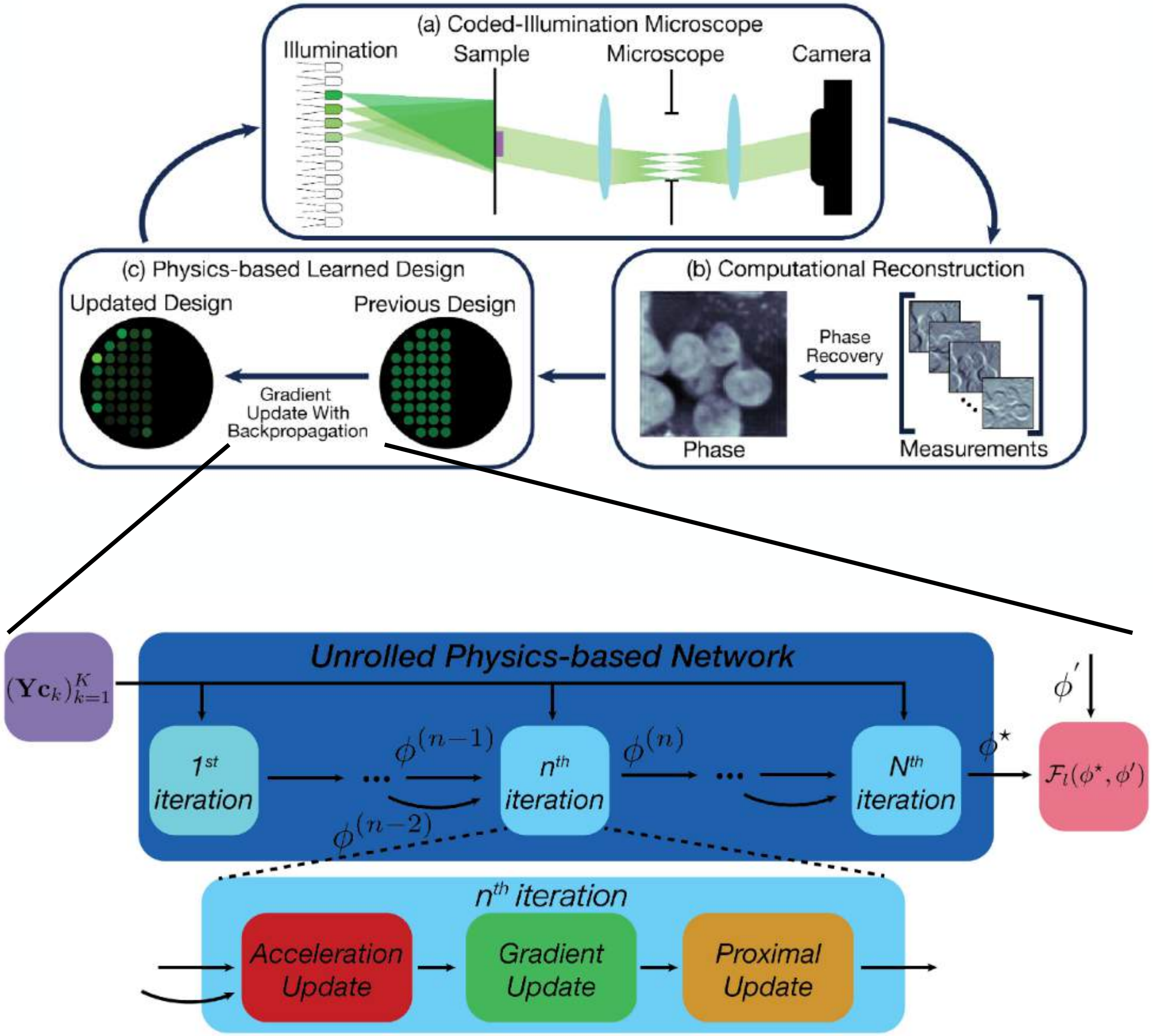
Monakhova, Kristina, et al. "Learned reconstructions for practical mask-based lensless imaging." *Optics express* 27.20 (2019): 28075-28090.

Deep unrolling in coded illumination pattern design



Kellman, M. R., Bostan, E., Repina, N. A., & Waller, L. (2019). Physics-based learned design: optimized coded-illumination for quantitative phase imaging. *IEEE Transactions on Computational Imaging*, *5*(3), 344-353.

Deep unrolling in coded illumination pattern design



Kellman, M. R., Bostan, E., Repina, N. A., & Waller, L. (2019). Physics-based learned design: optimized coded-illumination for quantitative phase imaging. *IEEE Transactions on Computational Imaging*, *5*(3), 344-353.

The Neumann Network

- Linear inverse problem formulation

$$\hat{x} = arg \min_x \|Ax - y\|^2 + \mathscr{R}(x)$$ and assume $$\mathscr{R}(x) = x^T R x$$

Gilton, D., Ongie, G. and Willett, R., 2019. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, *6*, pp.328-343.

The Neumann Network

- Linear inverse problem formulation

$$\hat{x} = arg\min_x \|Ax - y\|^2 + \mathscr{R}(x) \quad \text{and assume} \quad \mathscr{R}(x) = x^T R x$$

- Then the minimizer is

$$\hat{x} = (A^T A + R)^{-1} A^T y$$

Gilton, D., Ongie, G. and Willett, R., 2019. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, *6*, pp.328-343.

The Neumann Network

- Linear inverse problem formulation

$$\hat{x} = arg \min_x \|Ax - y\|^2 + \mathscr{R}(x)$$ and assume $$\mathscr{R}(x) = x^T R x$$

- Then the minimizer is

$$\hat{x} = (A^T A + R)^{-1} A^T y$$

- Neumann series expansion of inverse

$$\boldsymbol{B}^{-1} = \eta \sum_{k=0}^{\infty} (\boldsymbol{I} - \eta \boldsymbol{B})^k$$

Gilton, D., Ongie, G. and Willett, R., 2019. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, *6*, pp.328-343.

The Neumann Network

- Linear inverse problem formulation

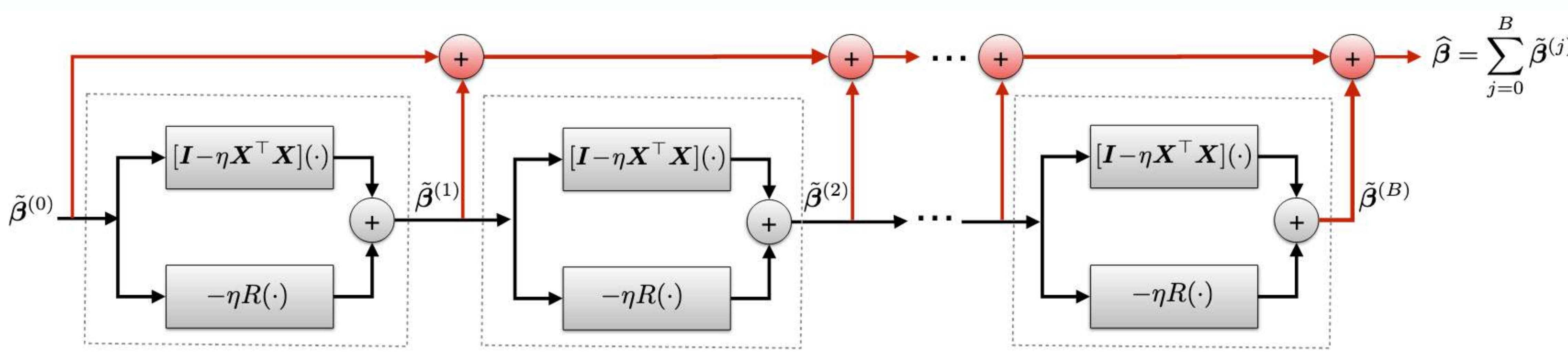
$$\hat{x} = arg \min_x \|Ax - y\|^2 + \mathscr{R}(x) \quad \text{and assume} \quad \mathscr{R}(x) = x^T R x$$

- Then the minimizer is

$$\hat{x} = (A^T A + R)^{-1} A^T y$$

- Neumann series expansion of inverse

$$\boldsymbol{B}^{-1} = \eta \sum_{k=0}^{\infty} (\boldsymbol{I} - \eta \boldsymbol{B})^k$$



Gilton, D., Ongie, G. and Willett, R., 2019. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, *6*, pp.328-343.

The Neumann Network

- Linear inverse problem formulation

$$\hat{x} = arg \min_x \|Ax - y\|^2 + \mathscr{R}(x) \quad \text{and assume} \quad \mathscr{R}(x) = x^T R x$$

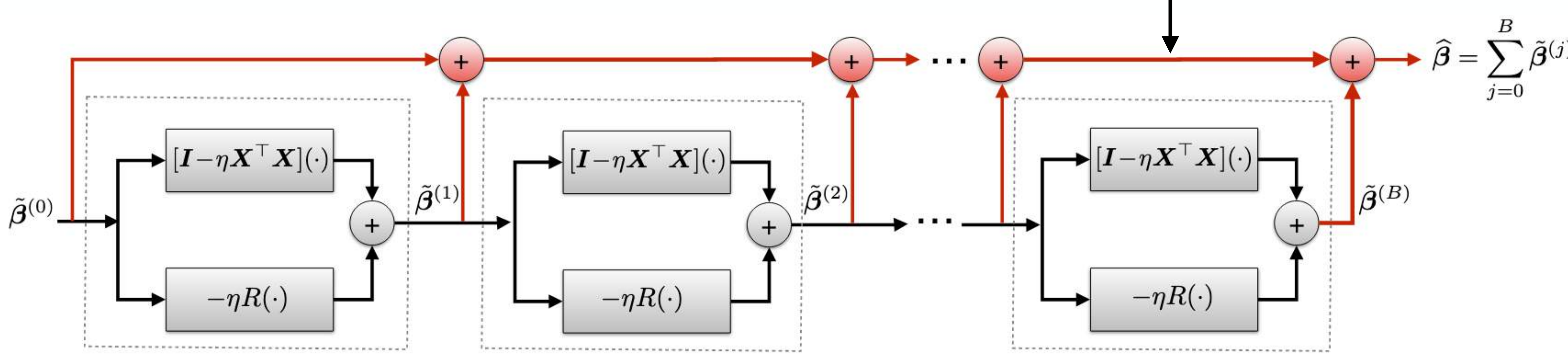
- Then the minimizer is

$$\hat{x} = (A^T A + R)^{-1} A^T y$$

- Neumann series expansion of inverse

$$\boldsymbol{B}^{-1} = \eta \sum_{k=0}^{\infty} (\boldsymbol{I} - \eta \boldsymbol{B})^k$$

skip connections arise naturally!



Gilton, D., Ongie, G. and Willett, R., 2019. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, *6*, pp.328-343.

Diffusion Models

(Many slides stolen from Arash Vahdat and collaborators see refs at the end)

Text-to-Image Models

Text-to-Image Models

DALL·E 2

OpenAI

Text-to-Image Models

Text-to-Image Models

DALL·E 2

OpenAI

Midjourney

Google

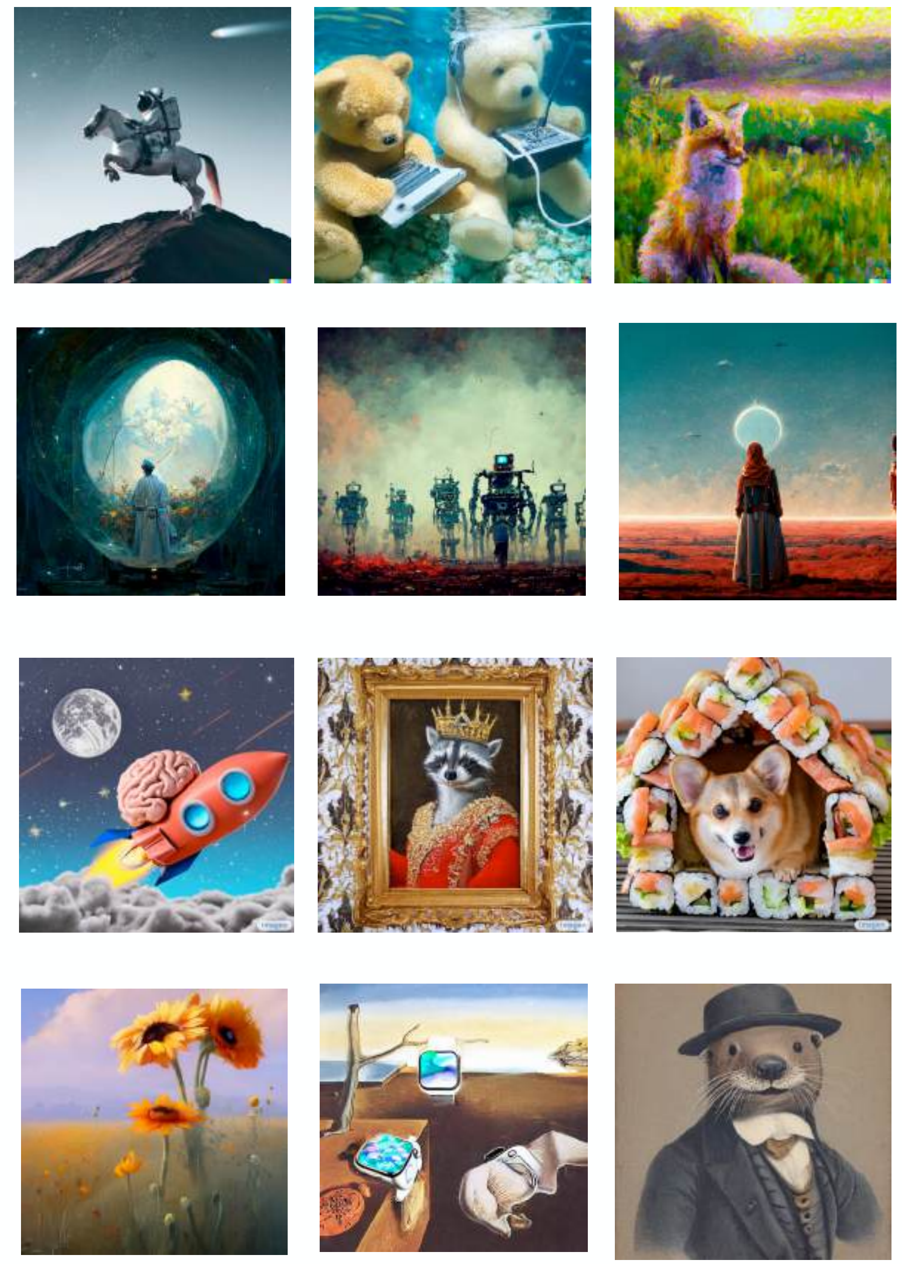
Text-to-Image Models

DALL·E 2

Midjourney

Imagen

Stable Diffusion



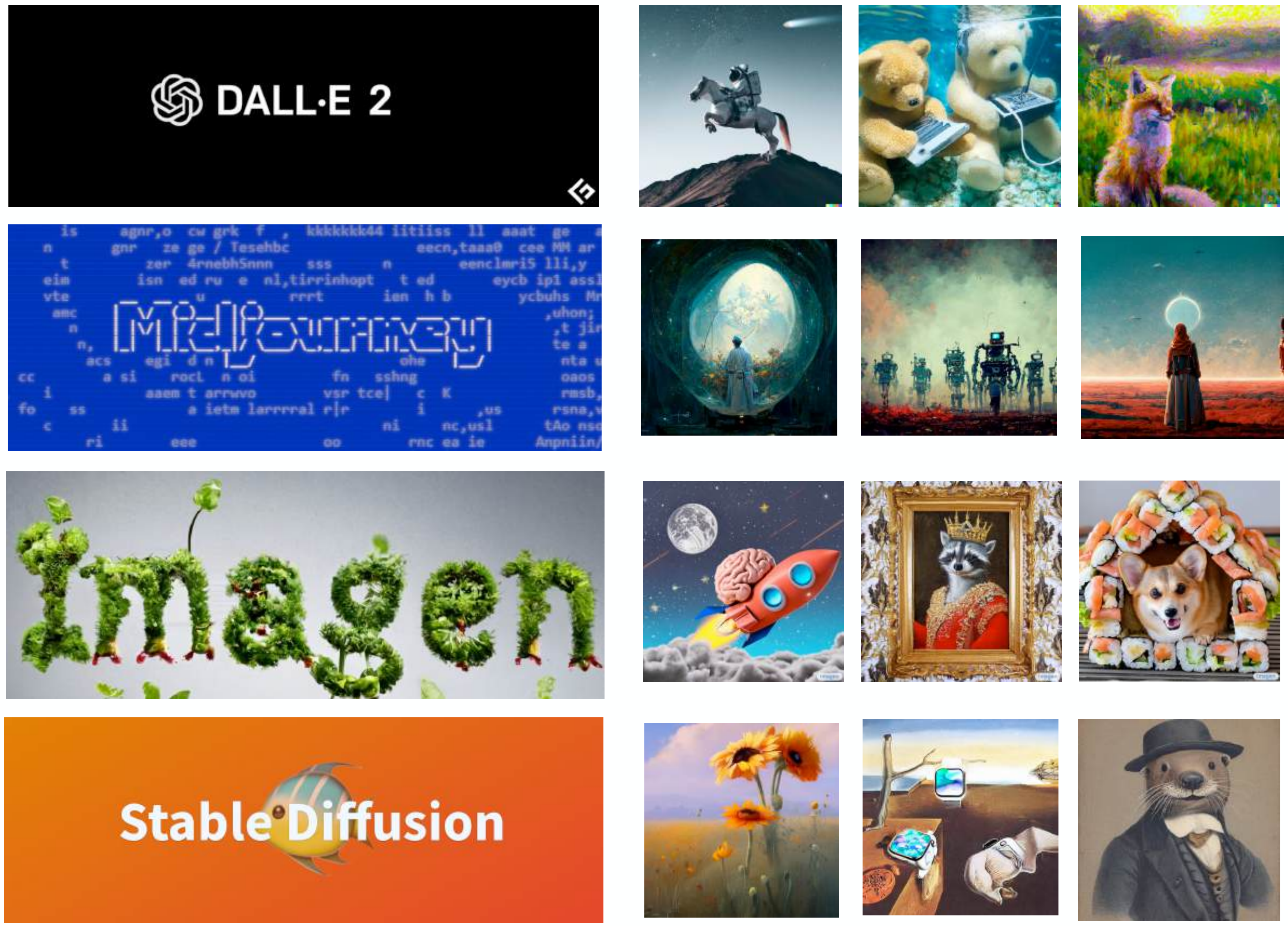
OpenAI

MIDJOURNEY AI

Google

stability.ai

Text-to-Image Models



OpenAI
MIDJOURNEY AI
Google
stability.ai

- What made this possible? Two key advances:
  - CLIP
  - Powerful generative models (diffusion)

Denoising Diffusion Probabilistic Models

$x_0 \sim q(x)$

real data distribution

Denoising Diffusion Probabilistic Models

$\cdots$

$x_0 \sim q(x)$

real data distribution

x_{t-1}

Denoising Diffusion Probabilistic Models

$x_0 \sim q(x)$

real data distribution

$\cdots$

x_{t-1}

x_t

Denoising Diffusion Probabilistic Models

$x_0 \sim q(x)$

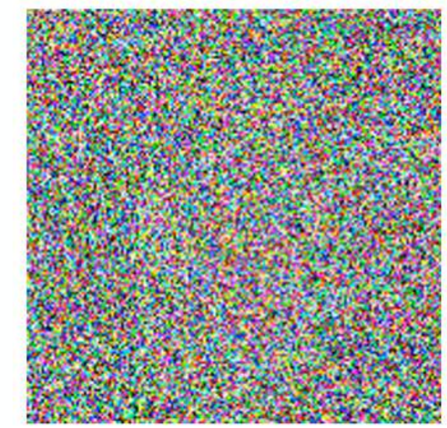
real data distribution

…

x_{t-1}

x_t

…



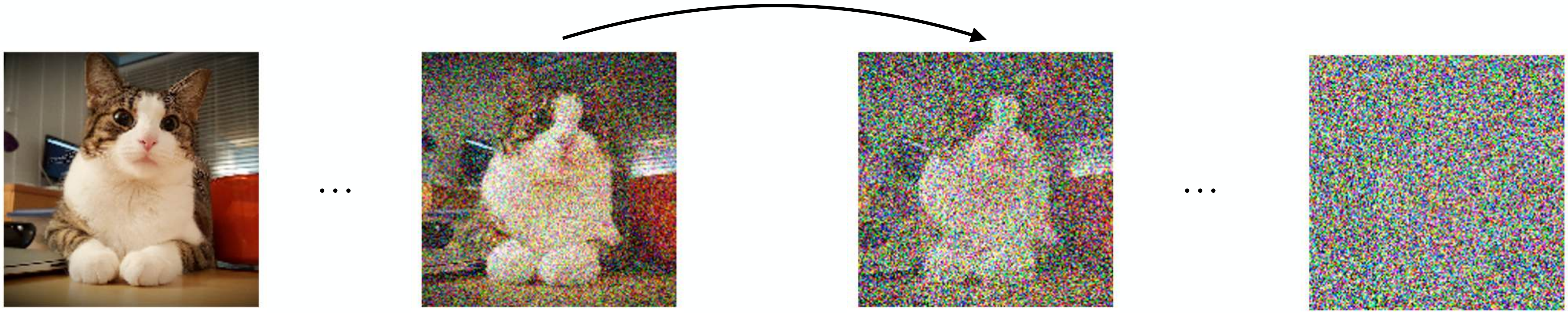
x_T

noise

Denoising Diffusion Probabilistic Models

forward diffusion process

$q(x_t | x_{t-1})$



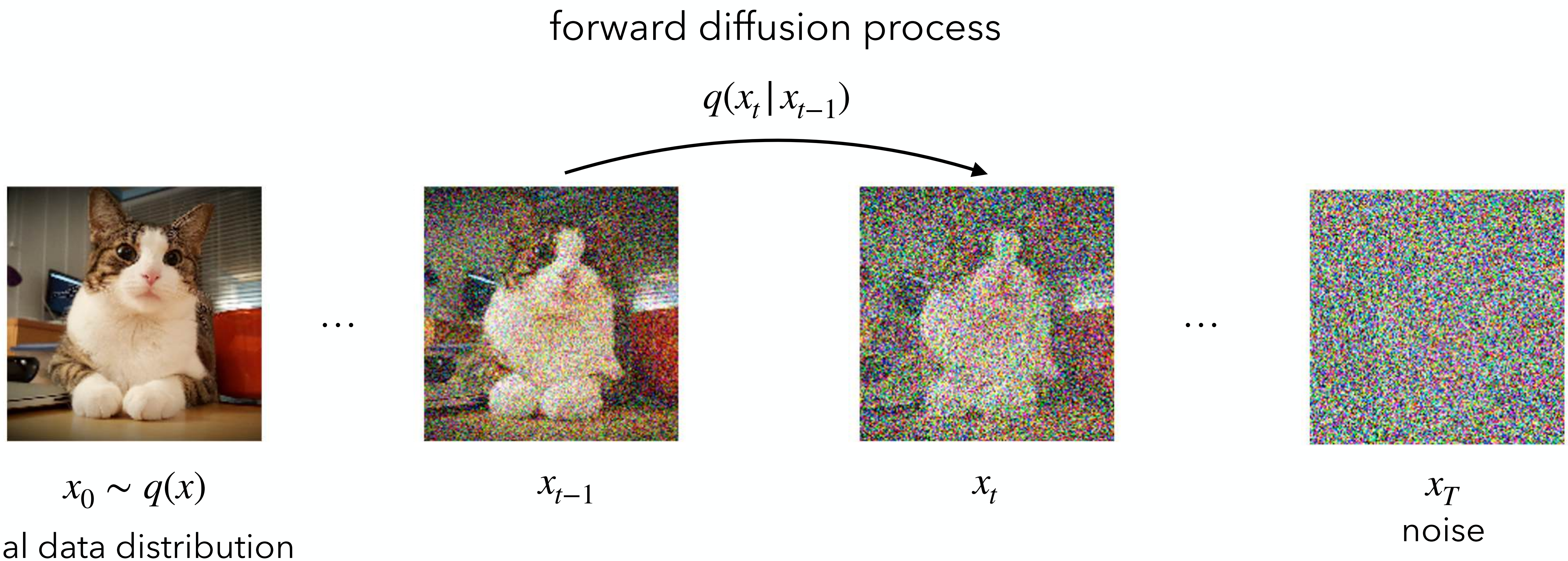
$x_0 \sim q(x)$
real data distribution

x_{t-1}

x_t

x_T
noise

Denoising Diffusion Probabilistic Models

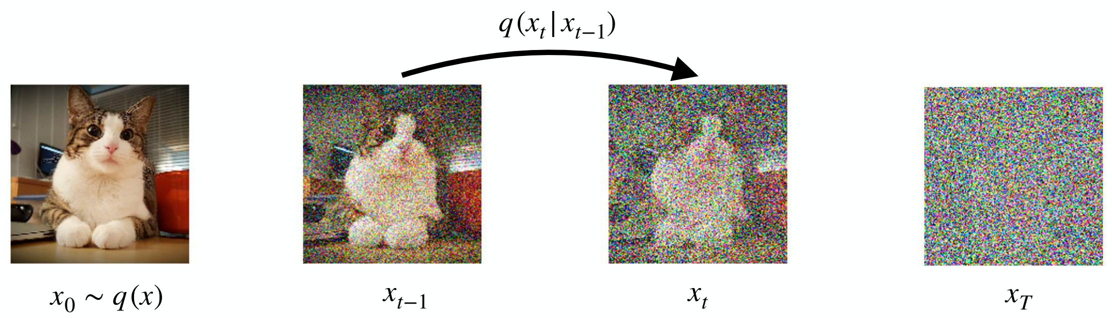


Forward diffusion process

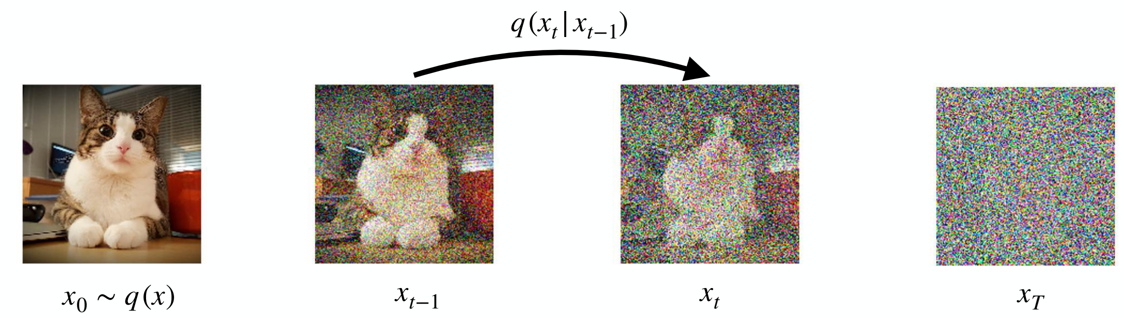
$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \qquad q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

variance schedule: $\{\beta_t \in (0,1)\}_{t=1}^{T}$

Forward sampling



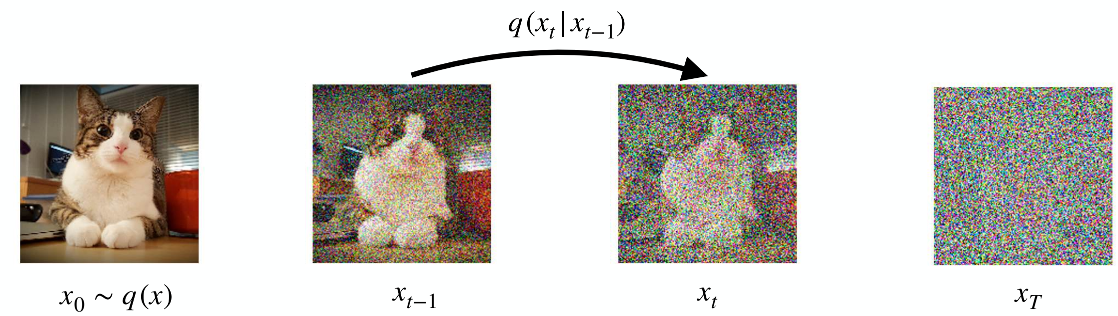
Forward sampling



- Reparametrization trick

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t\,; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \longrightarrow x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}, \;\; \epsilon_{t-1} \sim \mathcal{N}(0,I)$$

Forward sampling

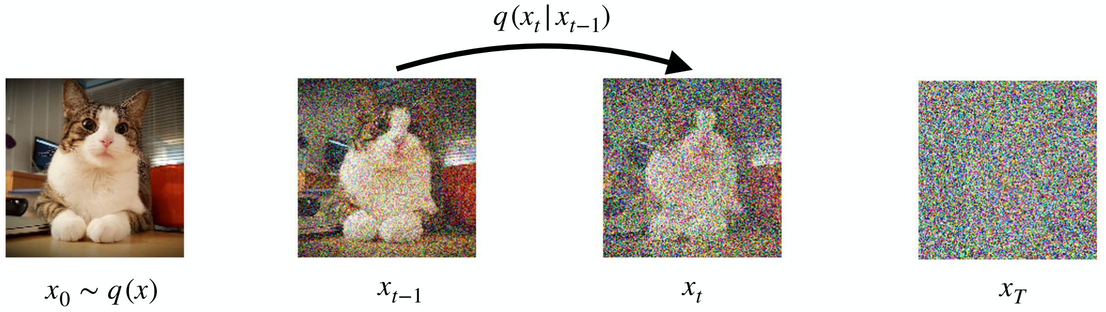


- Reparametrization trick

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t\,; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \longrightarrow x_t = \underbrace{\sqrt{1-\beta_t}}_{\text{deterministic variable}}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1},\ \ \epsilon_{t-1} \sim \mathcal{N}(0,I)$$

Forward sampling

- Reparametrization trick

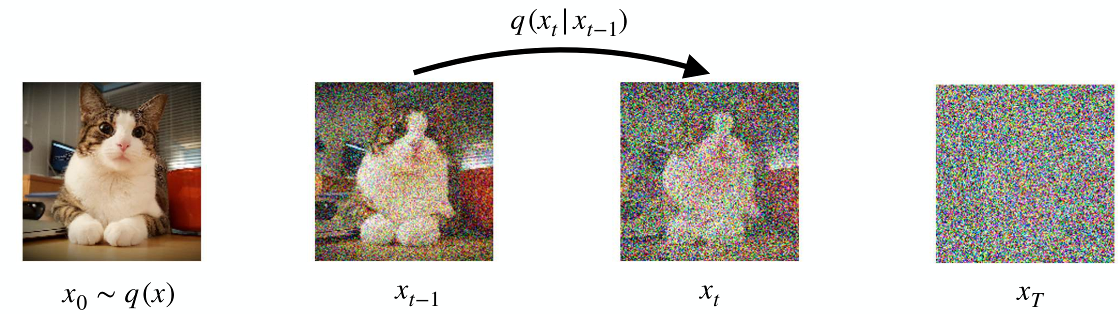


$$q(x_t|x_{t-1}) = \mathcal{N}(x_t\,; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \longrightarrow x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}, \;\; \epsilon_{t-1} \sim \mathcal{N}(0,I)$$

deterministic variable

aux. independent random variable

Forward sampling



- Reparametrization trick

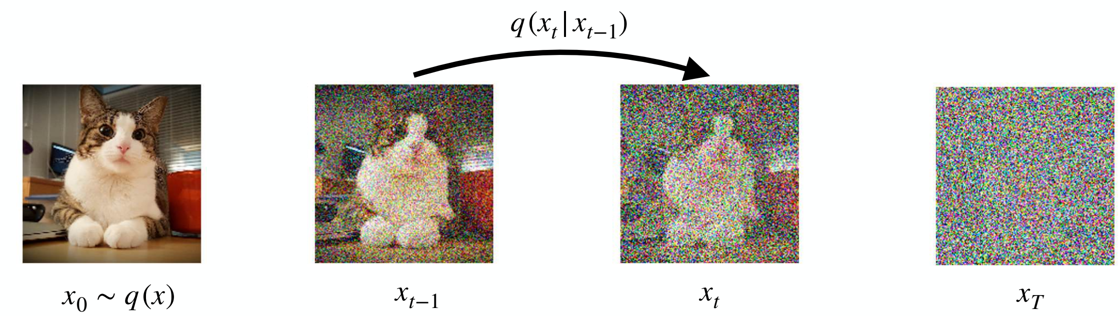
$$q(x_t|x_{t-1}) = \mathcal{N}(x_t\,; \sqrt{1-\beta_t}x_{t-1},\, \beta_t I) \longrightarrow x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}, \;\; \epsilon_{t-1} \sim \mathcal{N}(0, I)$$

deterministic variable

aux. independent random variable

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} := \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}$$

Forward sampling



- Reparametrization trick

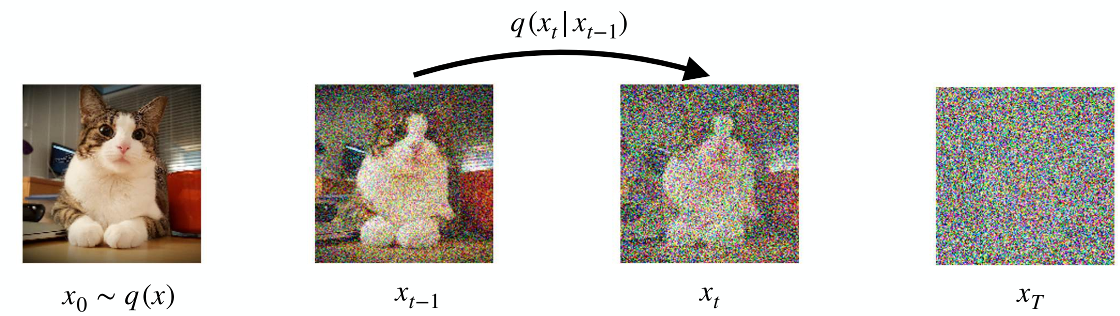
$$q(x_t | x_{t-1}) = \mathcal{N}(x_t ; \sqrt{1-\beta_t} x_{t-1}, \beta_t I) \longrightarrow x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t} \epsilon_{t-1}, \;\; \epsilon_{t-1} \sim \mathcal{N}(0, I)$$

deterministic variable

aux. independent random variable

$$x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t} \epsilon_{t-1} := \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t} \epsilon_{t-1}$$
$$= \sqrt{\alpha_t} \sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{\alpha_t} \sqrt{1-\alpha_{t-1}} \epsilon_{t-2} + \sqrt{1-\alpha_t} \epsilon_{t-1}$$

Forward sampling



- Reparametrization trick

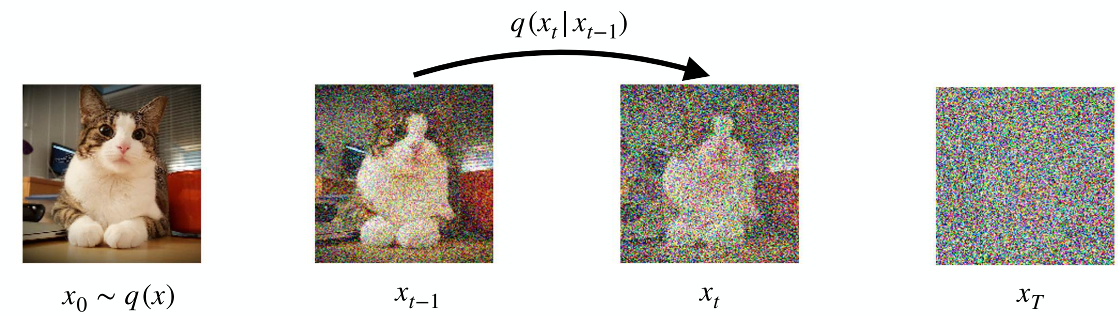
$$q(x_t|x_{t-1}) = \mathcal{N}(x_t\,;\sqrt{1-\beta_t}x_{t-1},\,\beta_t I) \longrightarrow x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1},\;\; \epsilon_{t-1} \sim \mathcal{N}(0,I)$$

deterministic variable

aux. independent random variable

$$\begin{aligned} x_t &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} := \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \end{aligned}$$

Forward sampling

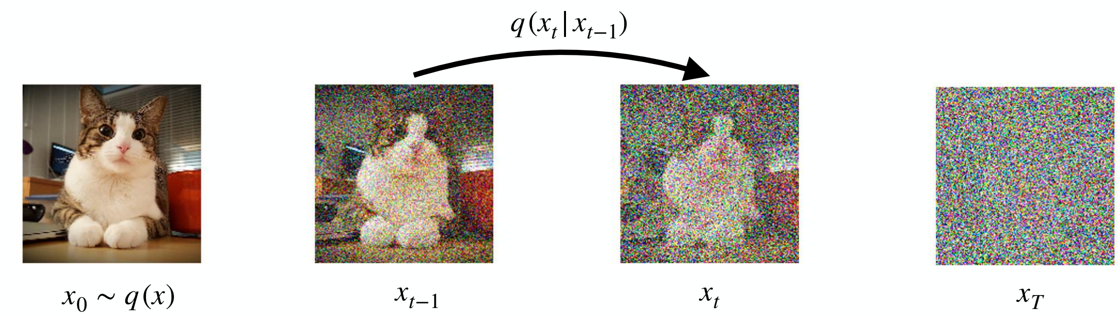


- Reparametrization trick

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t ; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \longrightarrow x_t = \underbrace{\sqrt{1-\beta_t}x_{t-1}}_{\text{deterministic variable}} + \underbrace{\sqrt{\beta_t}\epsilon_{t-1}}_{\text{aux. independent random variable}}, \;\; \epsilon_{t-1} \sim \mathcal{N}(0,I)$$

$$\begin{aligned} x_t &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} := \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \quad = \ldots = \sqrt{\alpha_t\alpha_{t-1}\ldots\alpha_1}x_0 + \sqrt{1-\alpha_t\alpha_{t-1}\ldots\alpha_1}\bar{\epsilon} \end{aligned}$$

Forward sampling



- Reparametrization trick

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t \,; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \longrightarrow x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}, \;\; \epsilon_{t-1} \sim \mathcal{N}(0,I)$$

deterministic variable

aux. independent random variable

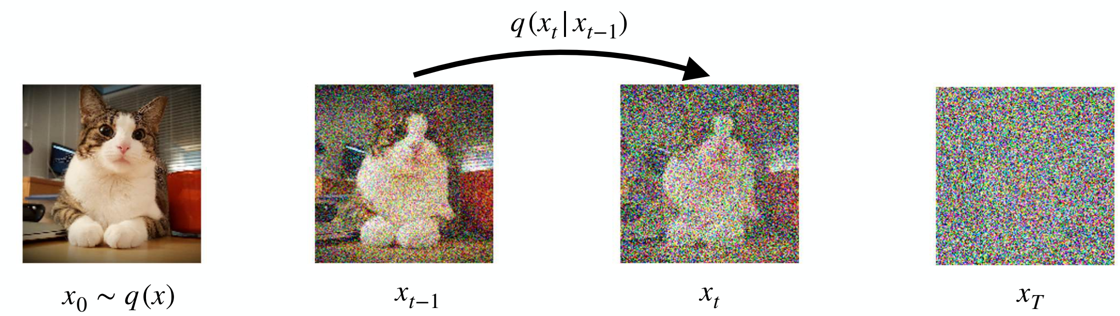
$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} := \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}$$

$$= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1}$$

$$= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \quad = \ldots = \sqrt{\alpha_t\alpha_{t-1}\ldots\alpha_1}x_0 + \sqrt{1-\alpha_t\alpha_{t-1}\ldots\alpha_1}\bar{\epsilon}$$

$$:= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0,I)$$

Forward sampling



- Reparametrization trick

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t\,; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \longrightarrow x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1},\ \ \epsilon_{t-1} \sim \mathcal{N}(0,I)$$

deterministic variable

aux. independent random variable

$$\begin{aligned} x_t &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1} := \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} \quad = \ldots = \sqrt{\alpha_t\alpha_{t-1}\ldots\alpha_1}x_0 + \sqrt{1-\alpha_t\alpha_{t-1}\ldots\alpha_1}\bar{\epsilon} \\ &:= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0,I) \end{aligned}$$

Sampling from forward process at any t :

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0,I)$$

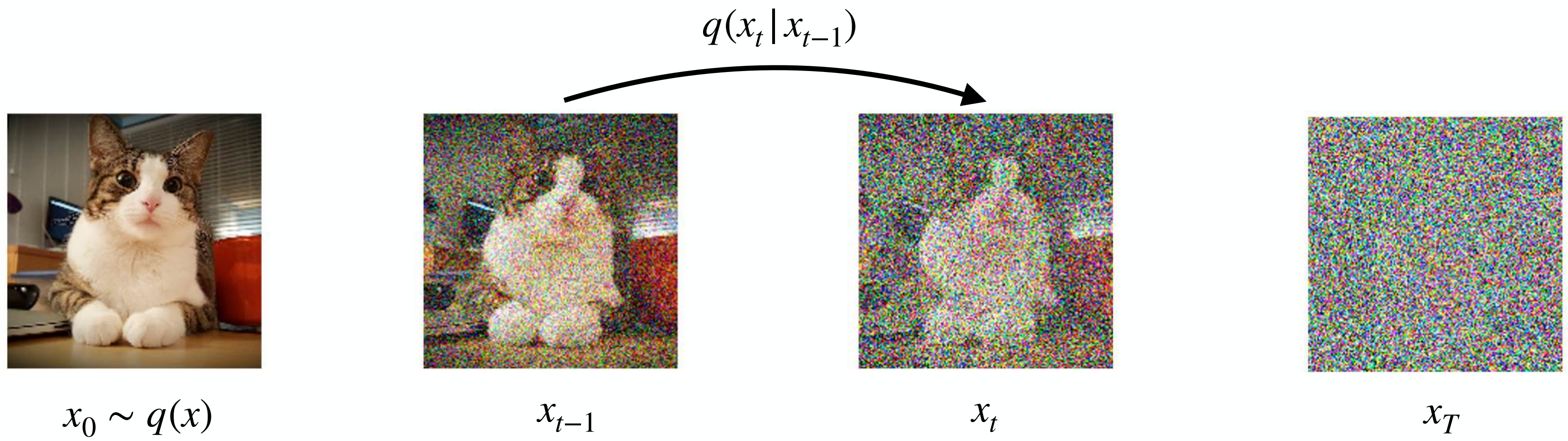
Reverse diffusion

"Creating noise from data is easy; creating data from noise is generative modeling."[1]

[1] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B., 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Reverse diffusion

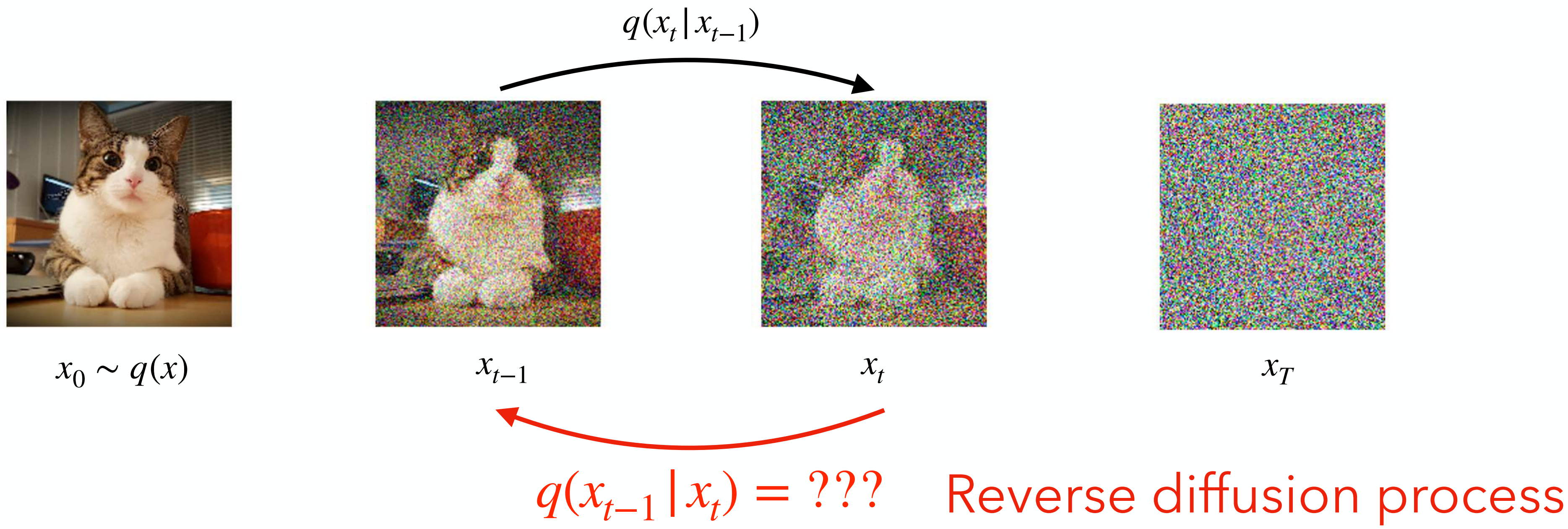
"Creating noise from data is easy; creating data from noise is generative modeling."[1]



[1] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B., 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Reverse diffusion

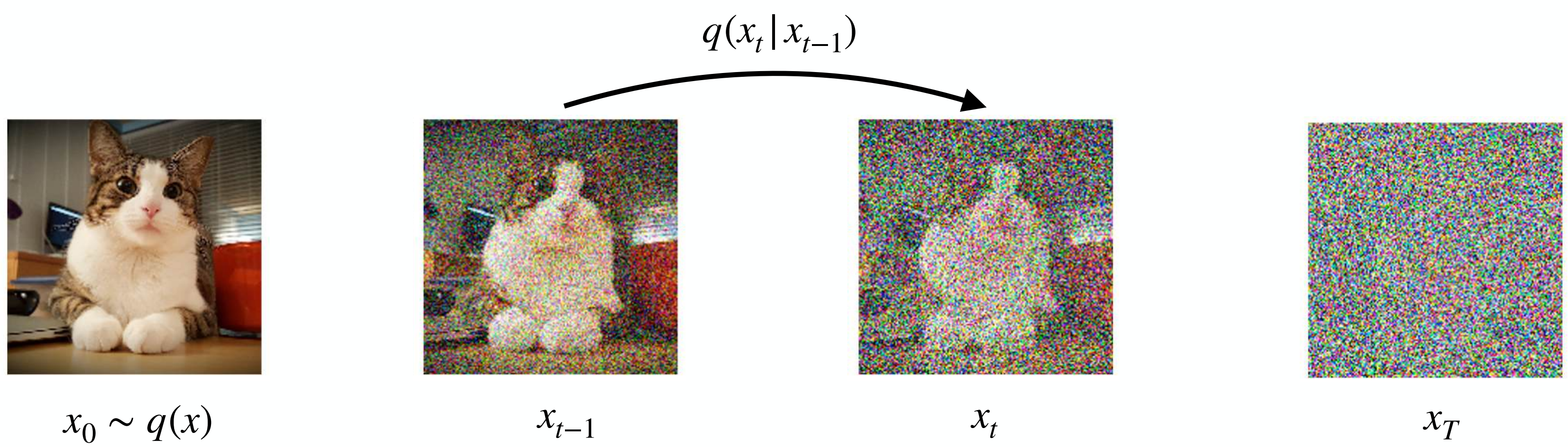
"Creating noise from data is easy; creating data from noise is generative modeling."[1]



[1] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B., 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Reverse diffusion

"Creating noise from data is easy; creating data from noise is generative modeling."[1]



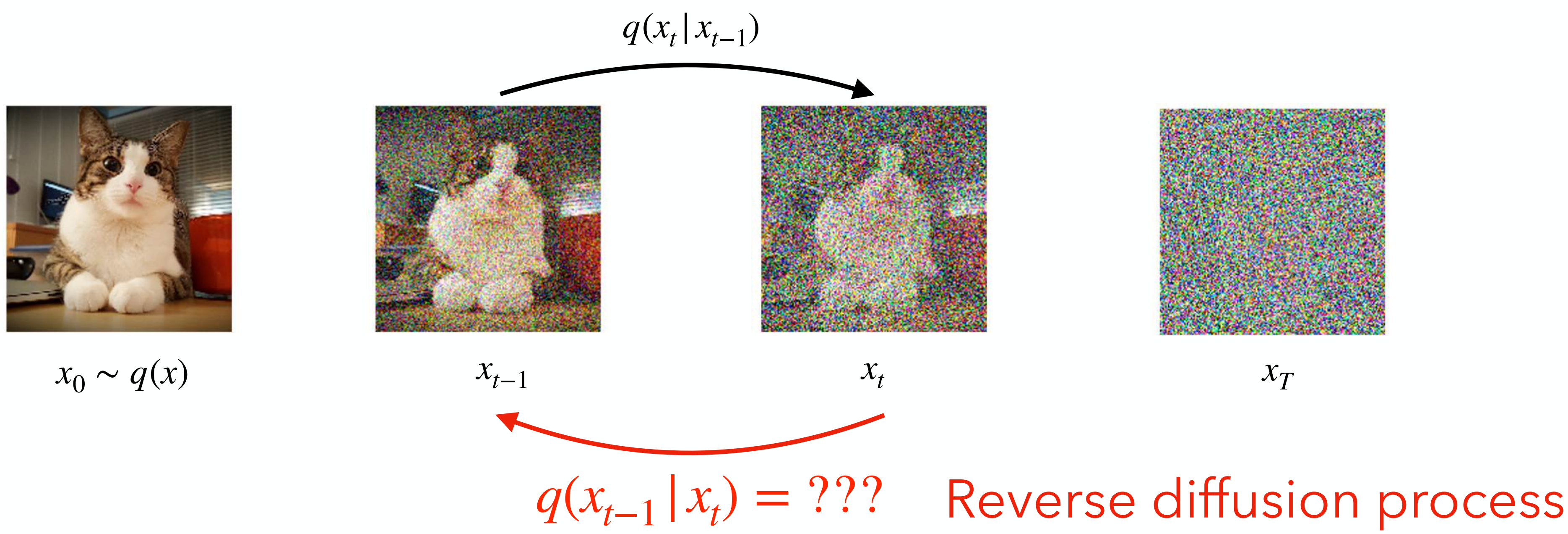
$q(x_{t-1}|x_t) = ???$ Reverse diffusion process

- If β_t is small, $q(x_{t-1}|x_t)$ is also Gaussian

[1] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B., 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Reverse diffusion

"Creating noise from data is easy; creating data from noise is generative modeling."[1]

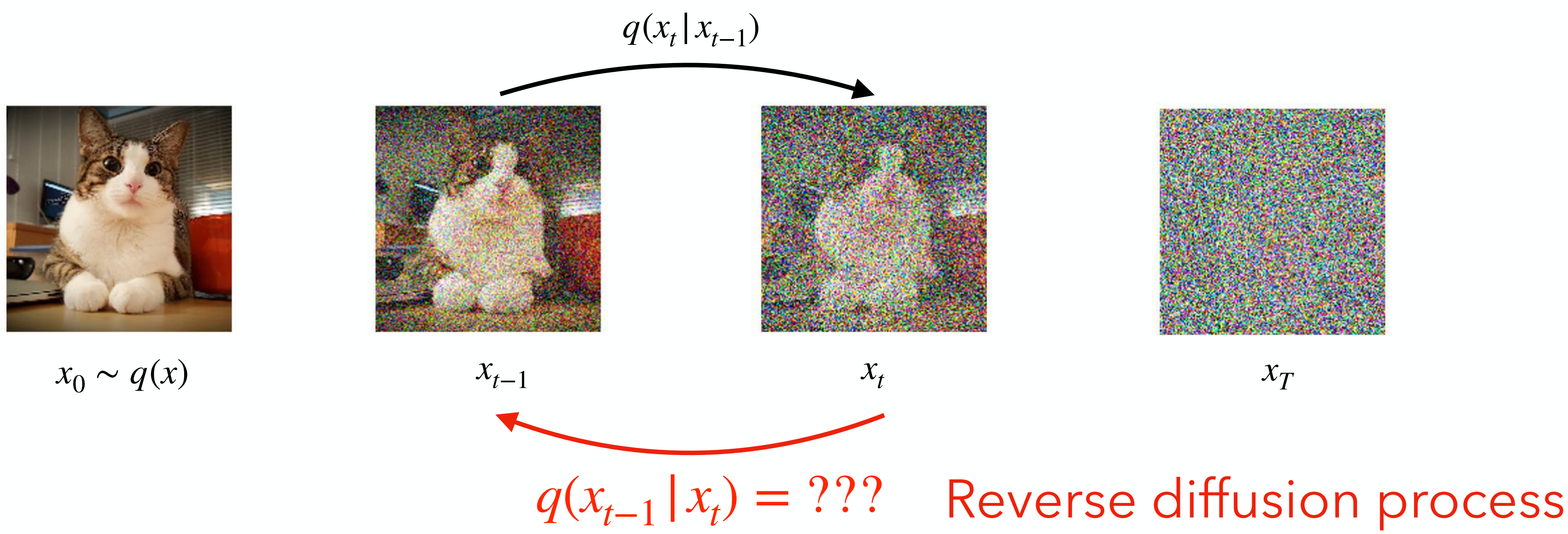


- If β_t is small, $q(x_{t-1} | x_t)$ is also Gaussian
- Estimating $q(x_{t-1} | x_t)$ is difficult, learn a model instead!

[1] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B., 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Reverse diffusion

"Creating noise from data is easy; creating data from noise is generative modeling."[1]



- If β_t is small, $q(x_{t-1} | x_t)$ is also Gaussian
- Estimating $q(x_{t-1} | x_t)$ is difficult, learn a model instead!

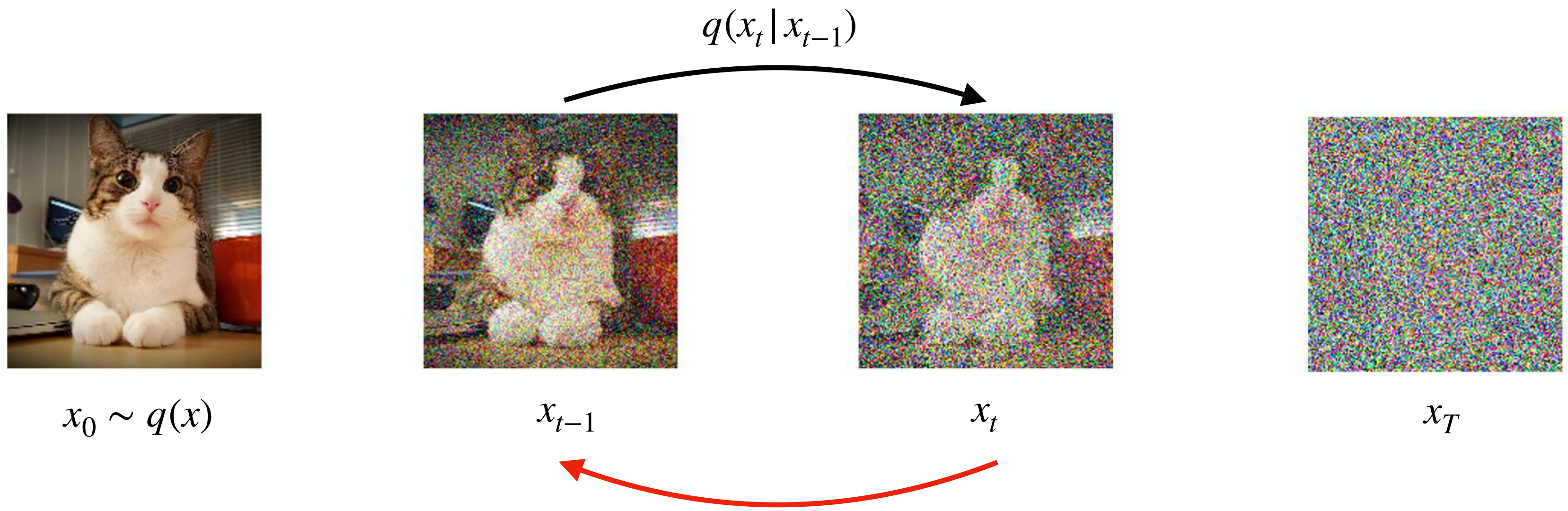
Reverse diffusion process

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t)) \qquad p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} | x_t)$$

[1] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B., 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Reverse diffusion

"Creating noise from data is easy; creating data from noise is generative modeling."[1]



$q(x_{t-1}|x_t) = ???$ Reverse diffusion process

- If β_t is small, $q(x_{t-1}|x_t)$ is also Gaussian
- Estimating $q(x_{t-1}|x_t)$ is difficult, learn a model instead!

Reverse diffusion process

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t)) \qquad p_\theta(x_{0:T}) = p_\theta(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

How to learn p_θ ?

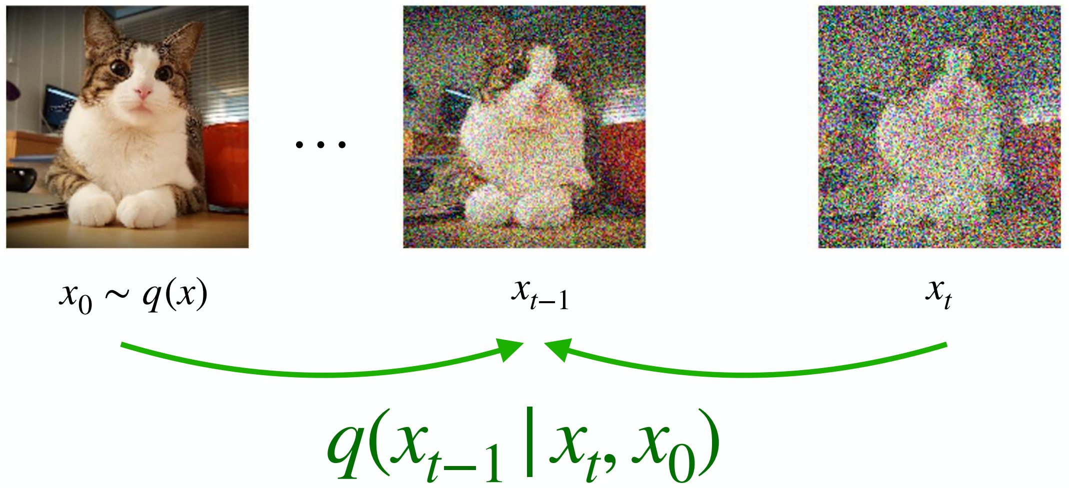
[1] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B., 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Reverse conditional probability

- Reverse conditional probability is tractable

Reverse conditional probability

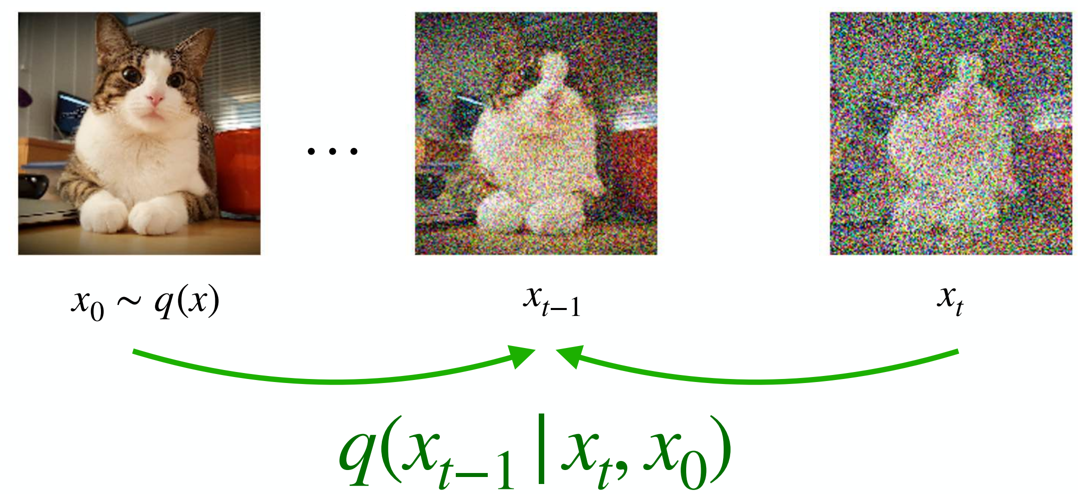
- Reverse conditional probability is tractable



Reverse conditional probability

- Reverse conditional probability is tractable

$$q(x_{t-1} | x_t, x_0) = \frac{q(x_t | x_{t-1}, x_0)\ q(x_{t-1} | x_0)}{q(x_t | x_0)}$$

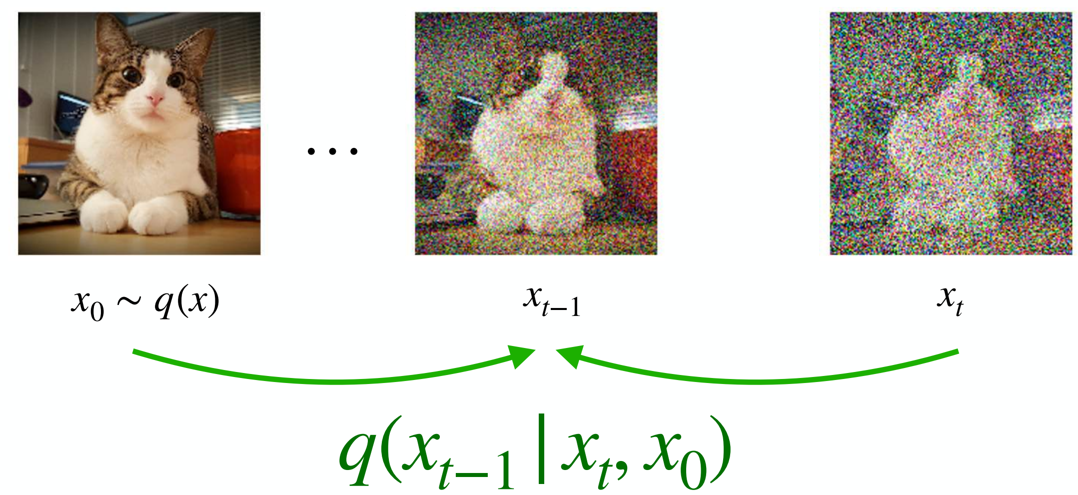


Reverse conditional probability

- Reverse conditional probability is tractable

$$q(x_{t-1} | x_t, x_0) = \frac{q(x_t | x_{t-1}, x_0)\ q(x_{t-1} | x_0)}{q(x_t | x_0)}$$

$$\dots$$



Reverse conditional probability

- Reverse conditional probability is tractable

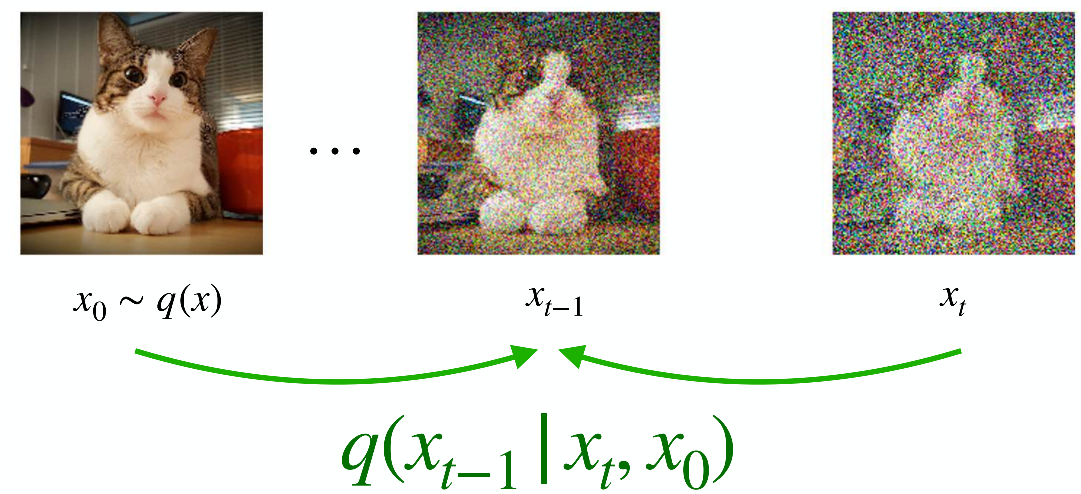
$$q(x_{t-1} | x_t, x_0) = \frac{q(x_t | x_{t-1}, x_0)\ q(x_{t-1} | x_0)}{q(x_t | x_0)}$$

$$\cdots$$

$$q(x_{t-1} | x_t, x_0) = \mathcal{N}(x_{t-1};\ \tilde{\mu}(x_t, x_0),\ \tilde{\beta}_t I)$$

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t \right)$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$$



Reverse conditional probability

- Reverse conditional probability is tractable

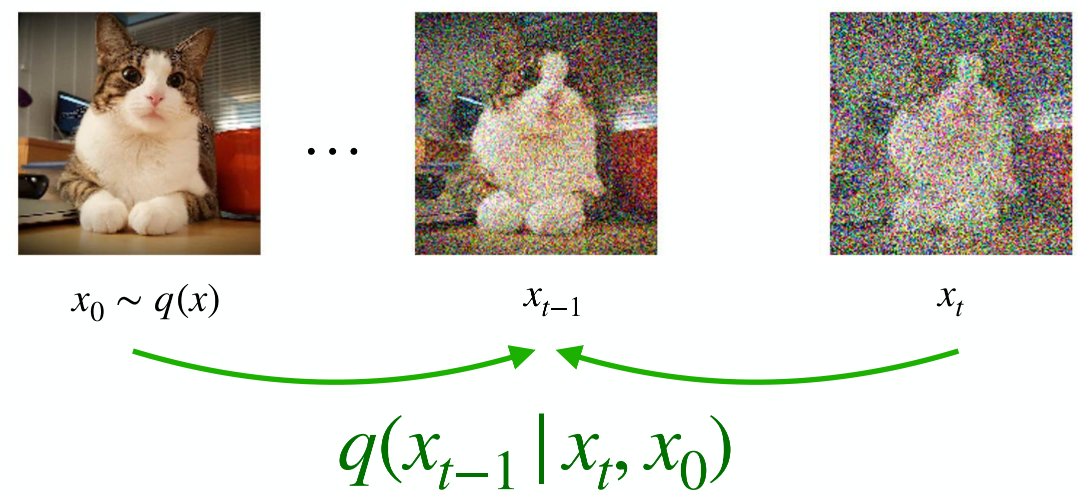
$$q(x_{t-1} | x_t, x_0) = \frac{q(x_t | x_{t-1}, x_0)\ q(x_{t-1} | x_0)}{q(x_t | x_0)}$$

$$\ldots$$

$$q(x_{t-1} | x_t, x_0) = \mathcal{N}(x_{t-1};\ \tilde{\mu}(x_t, x_0),\ \tilde{\beta}_t I)$$

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_t \right)$$

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$$



- Reverse process:

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$$

Reverse conditional probability

- Reverse conditional probability is tractable

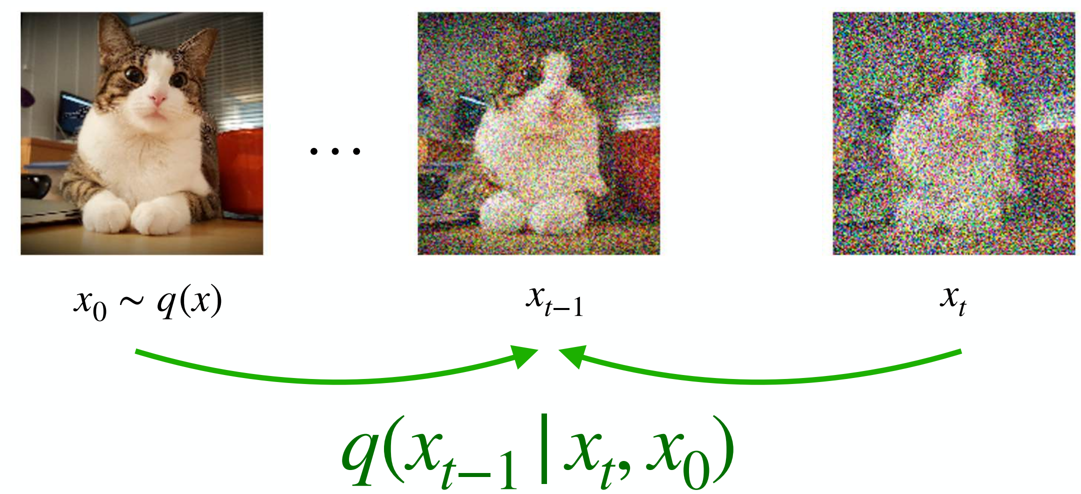
$$q(x_{t-1} | x_t, x_0) = \frac{q(x_t | x_{t-1}, x_0)\ q(x_{t-1} | x_0)}{q(x_t | x_0)}$$

$$\dots$$

$$q(x_{t-1} | x_t, x_0) = \mathcal{N}(x_{t-1};\ \tilde{\mu}(x_t, x_0),\ \tilde{\beta}_t I)$$

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right)$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$



- Reverse process:

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$$

$$\mu_\theta(x_t, t) := \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$

Putting it all together

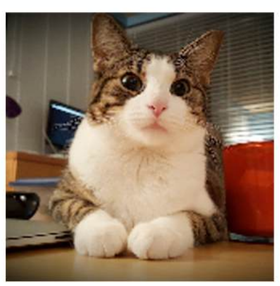
- Training

$$L_{simple} = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_o + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\|^2\right]$$

Putting it all together

- Training

$$L_{simple} = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_o + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\|^2\right]$$

$x_0 \sim q(x)$ 

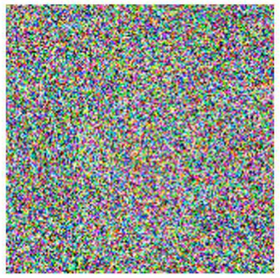
Putting it all together

- Training

$$L_{simple} = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_o + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\|^2\right]$$

$x_0 \sim q(x)$

$\epsilon \sim \mathcal{N}(0,I)$



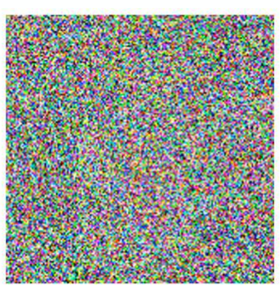
Putting it all together

- Training

$$L_{simple} = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_o + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\|^2\right]$$

$x_0 \sim q(x)$

$\epsilon \sim \mathcal{N}(0,I)$



$t \sim \mathrm{U}[1,T]$

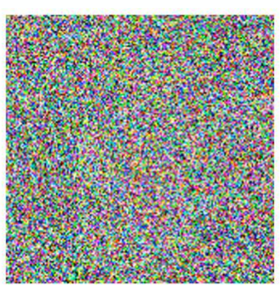
t

Putting it all together

- Training

$$L_{simple} = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_o + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\|^2\right]$$

$x_0 \sim q(x)$ 

$\epsilon \sim \mathcal{N}(0,I)$ 

$t \sim \mathrm{U}[1,T]$ t

forward diffusion

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

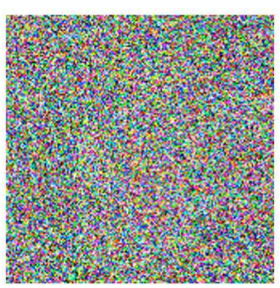
Putting it all together

- Training

$$L_{simple} = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_o + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\|^2\right]$$

$x_0 \sim q(x)$

$\epsilon \sim \mathcal{N}(0,I)$

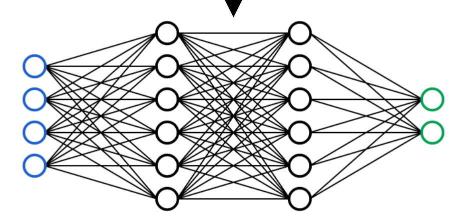


$t \sim \mathrm{U}[1,T]$ $\quad t$

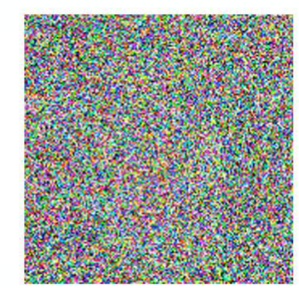
forward diffusion

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

t



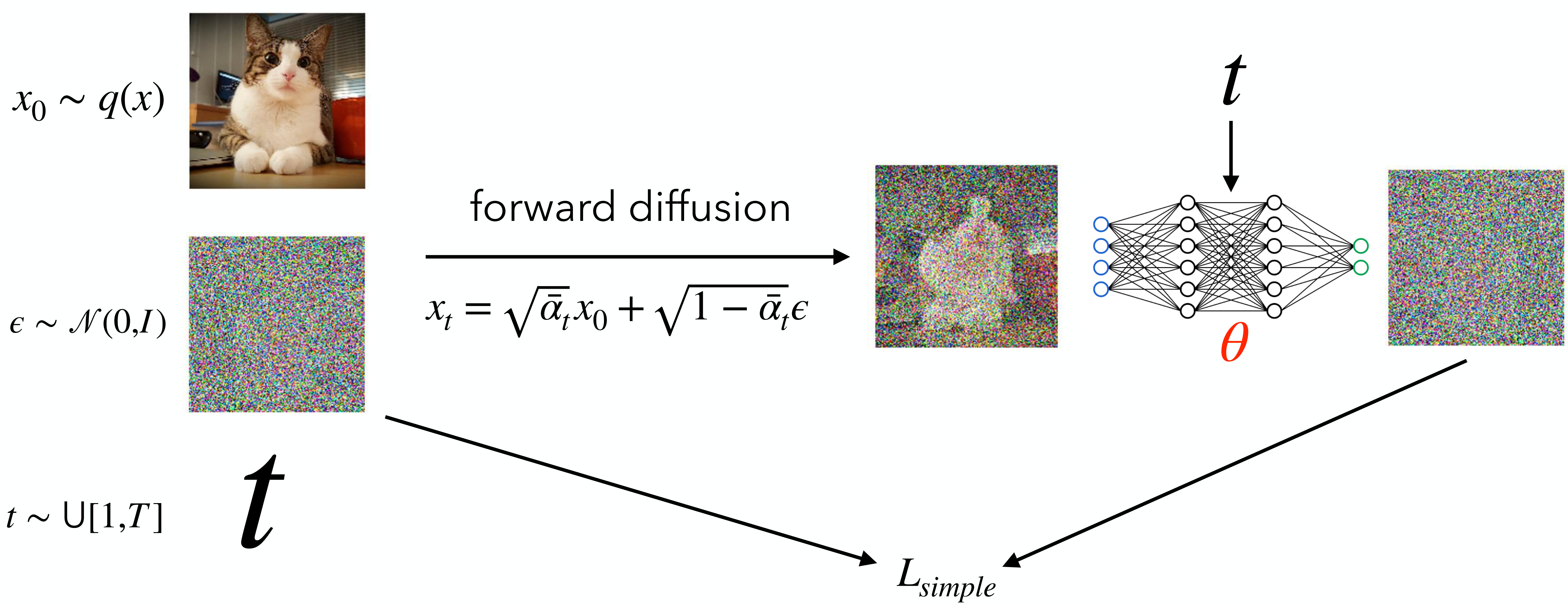
θ



Putting it all together

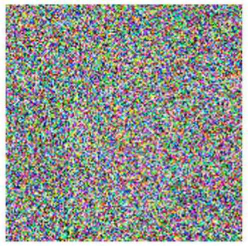
- Training

$$L_{simple} = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_o + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t\right)\|^2\right]$$



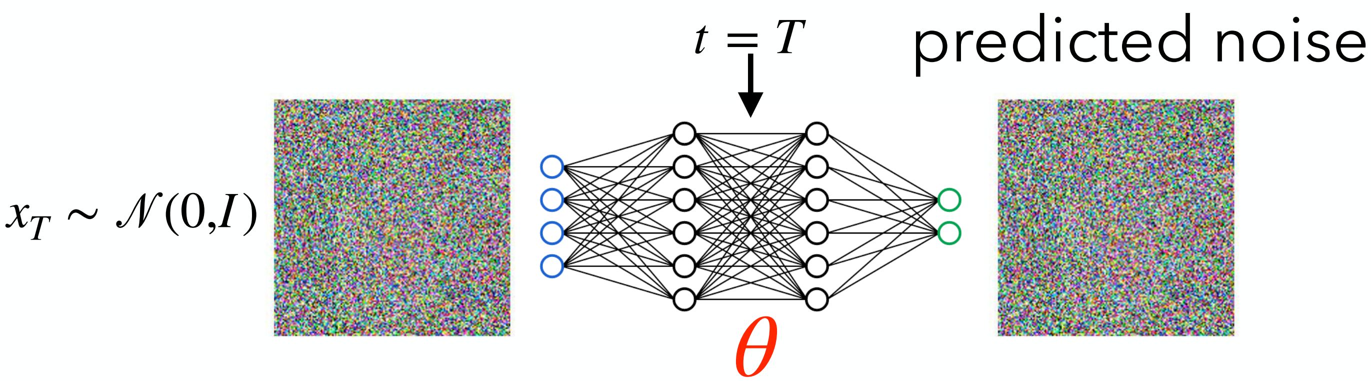
Putting it all together

- Sampling (naive)

$x_T \sim \mathcal{N}(0, I)$ 

Putting it all together

- Sampling (naive)



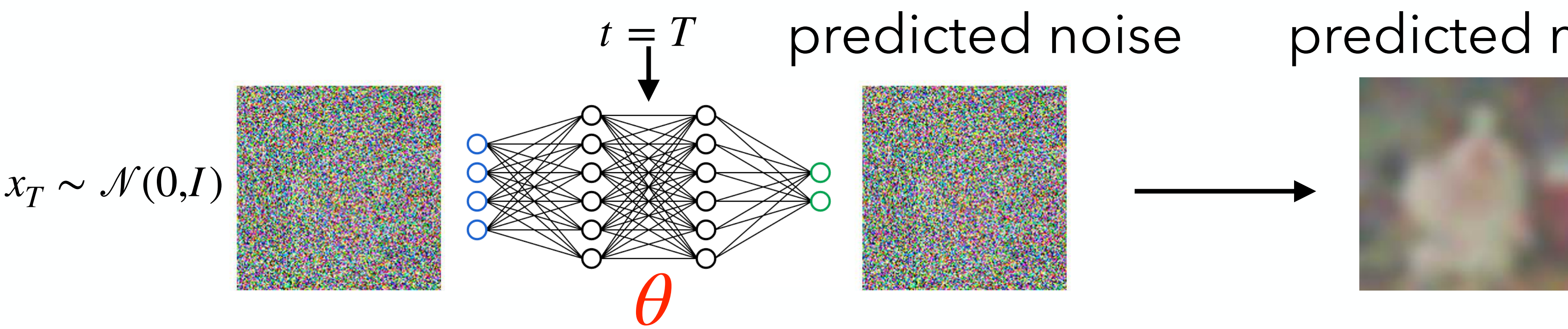
Putting it all together

- Sampling (naive)

Reverse diffusion process parameterization

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$$

$$\mu_\theta(x_t, t) := \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$



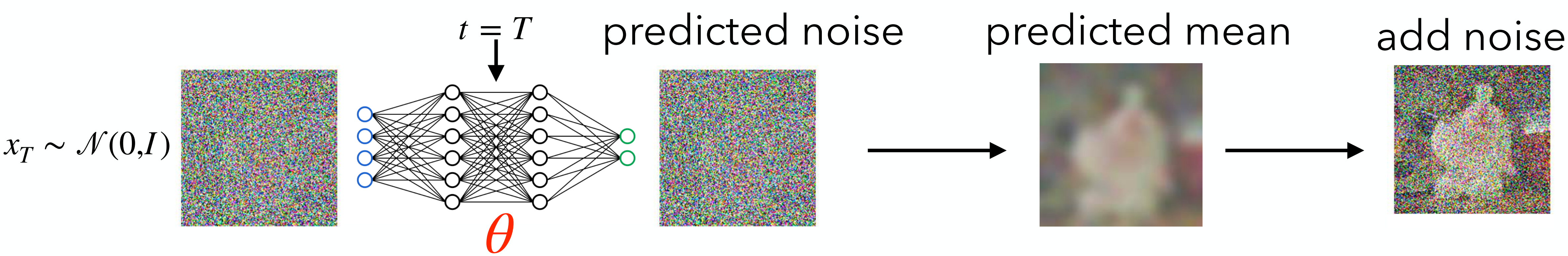
Putting it all together

- Sampling (naive)

Reverse diffusion process parameterization

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$$

$$\mu_\theta(x_t, t) := \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$



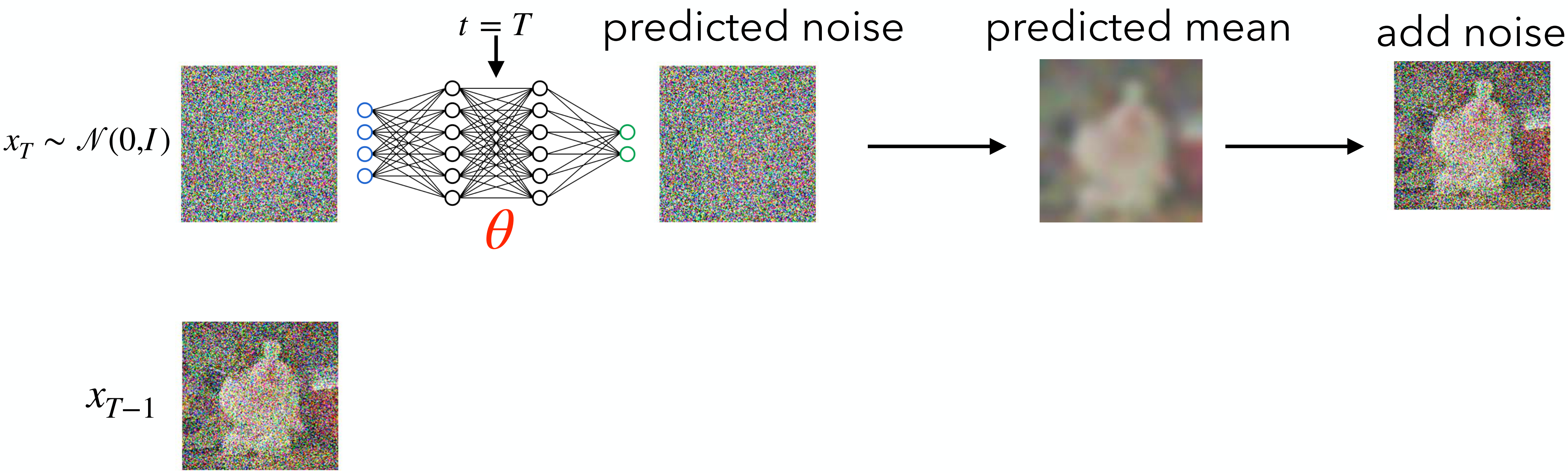
Putting it all together

- Sampling (naive)

Reverse diffusion process parameterization

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$$

$$\mu_\theta(x_t, t) := \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$



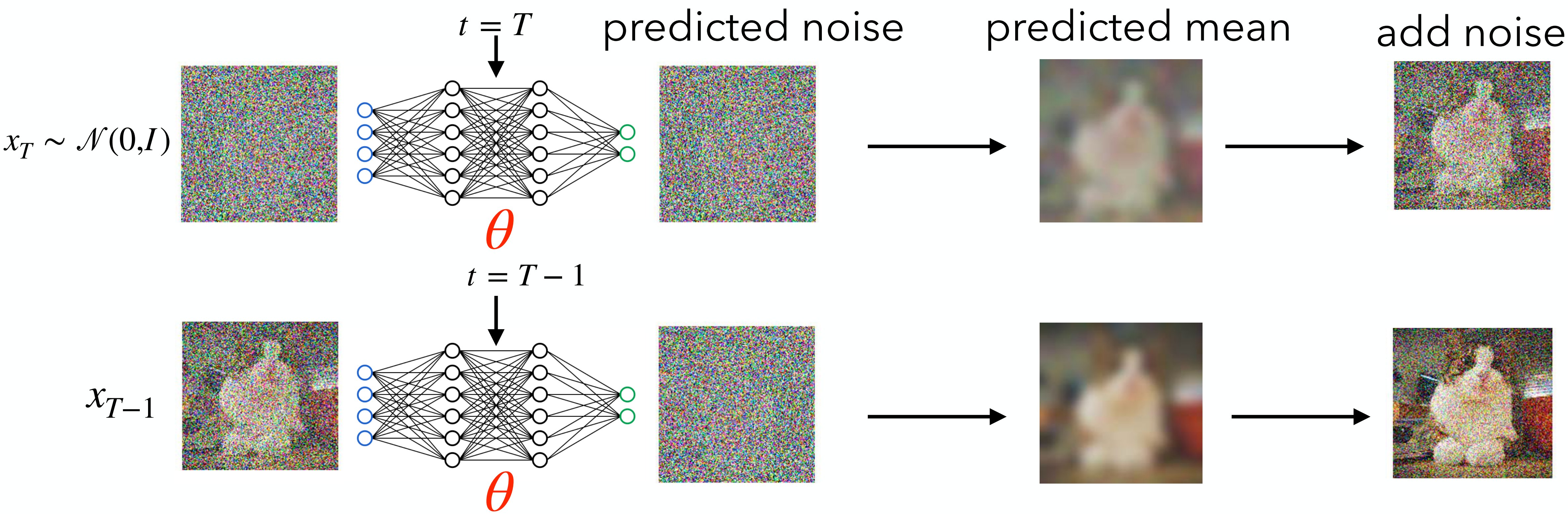
Putting it all together

- Sampling (naive)

Reverse diffusion process parameterization

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$$

$$\mu_\theta(x_t, t) := \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$



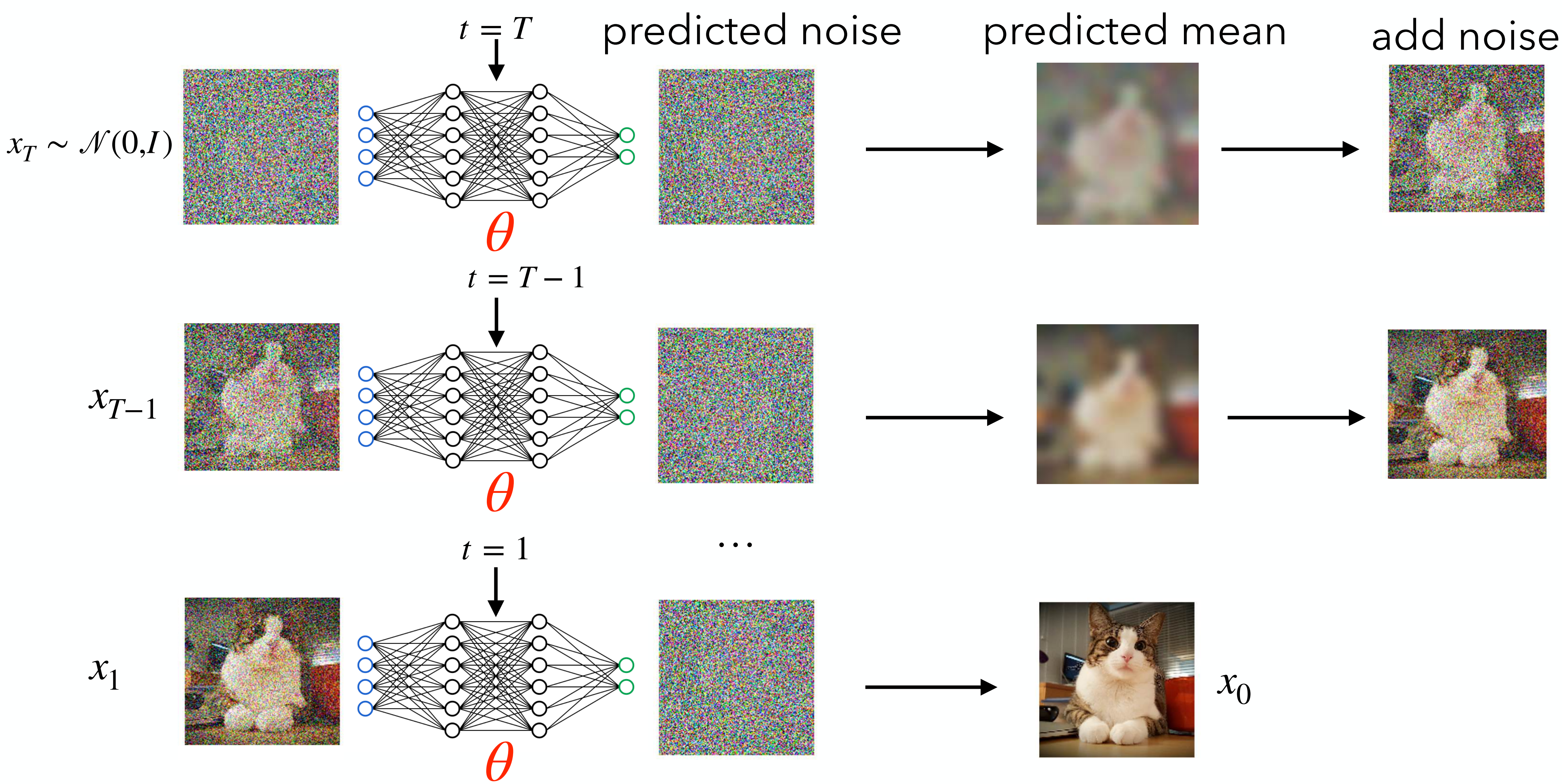
Putting it all together

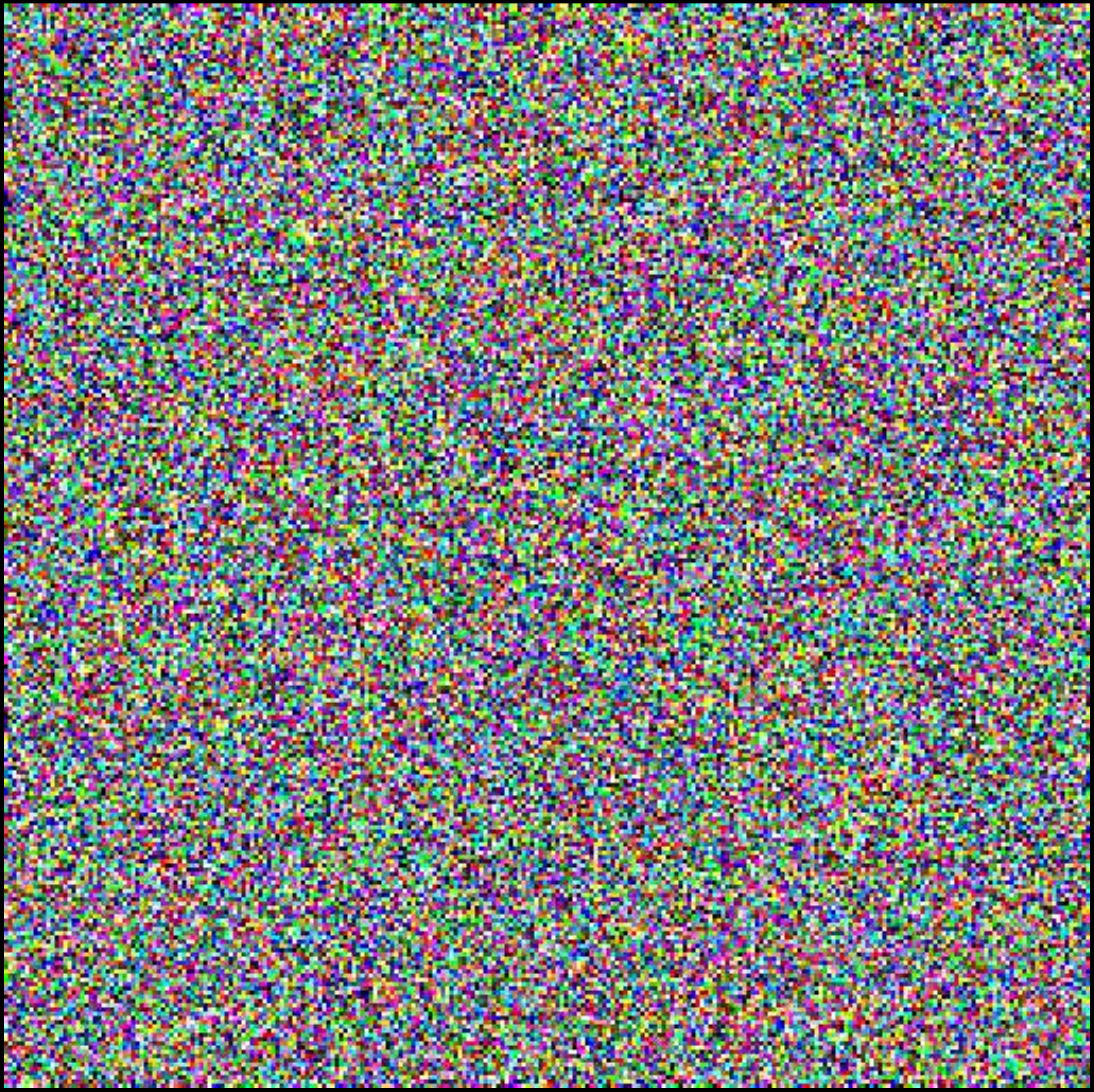
- Sampling (naive)

Reverse diffusion process parameterization

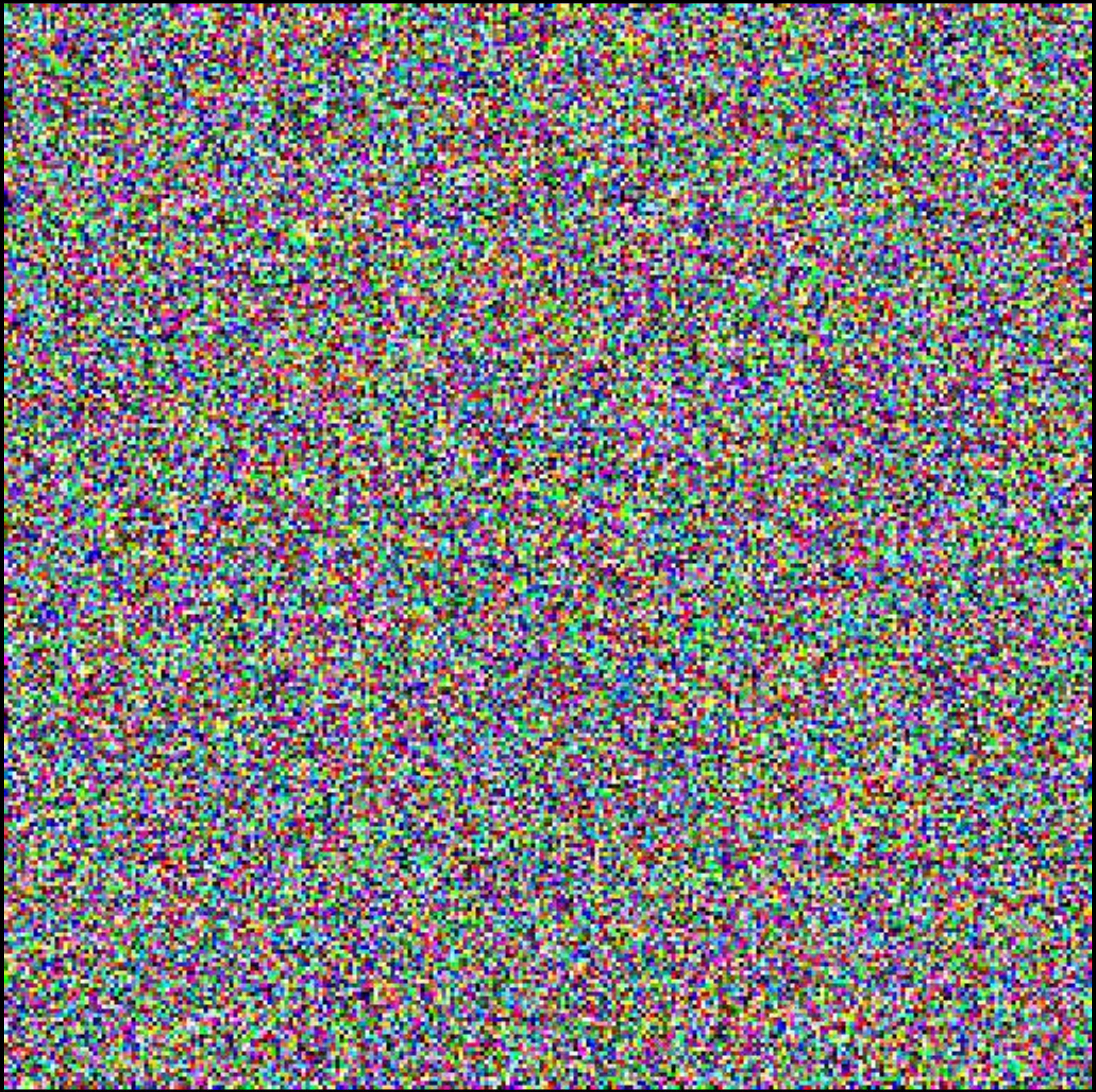
$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1};\ \mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$$

$$\mu_\theta(x_t, t) := \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$



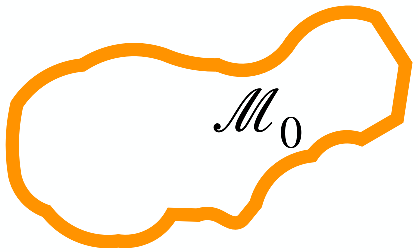


credit: Alex Nichol

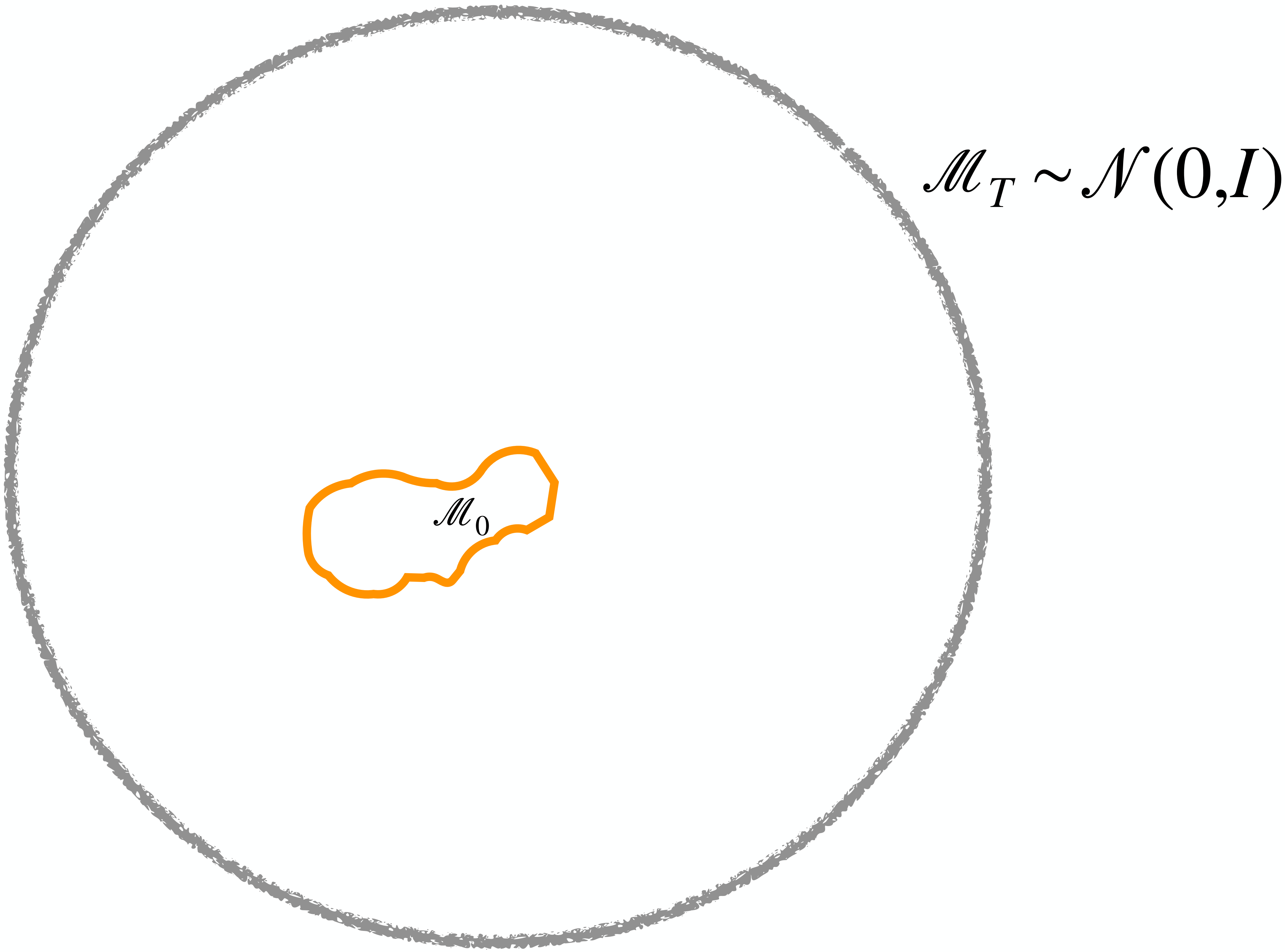


credit: Alex Nichol

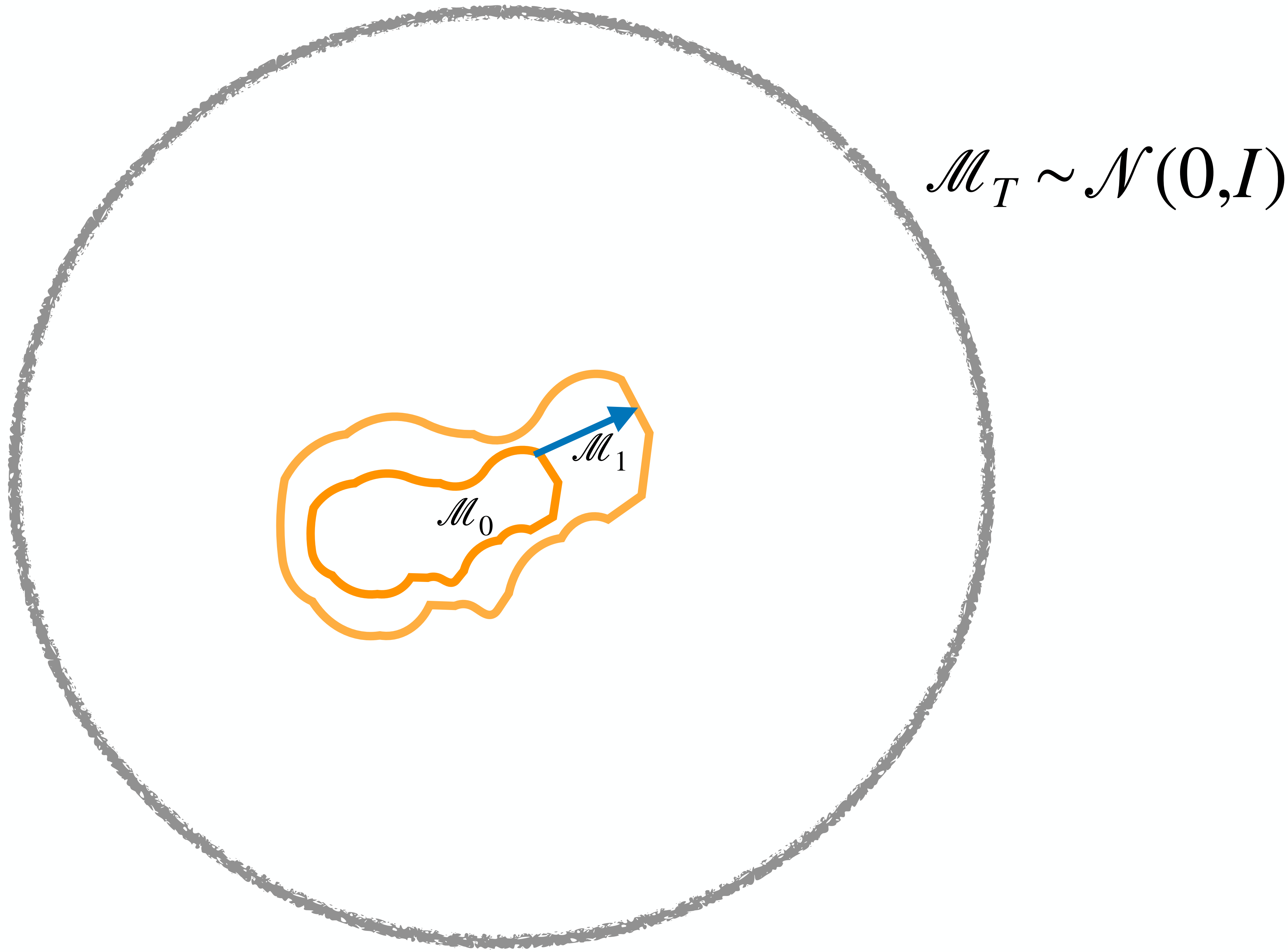
Geometric interpretation



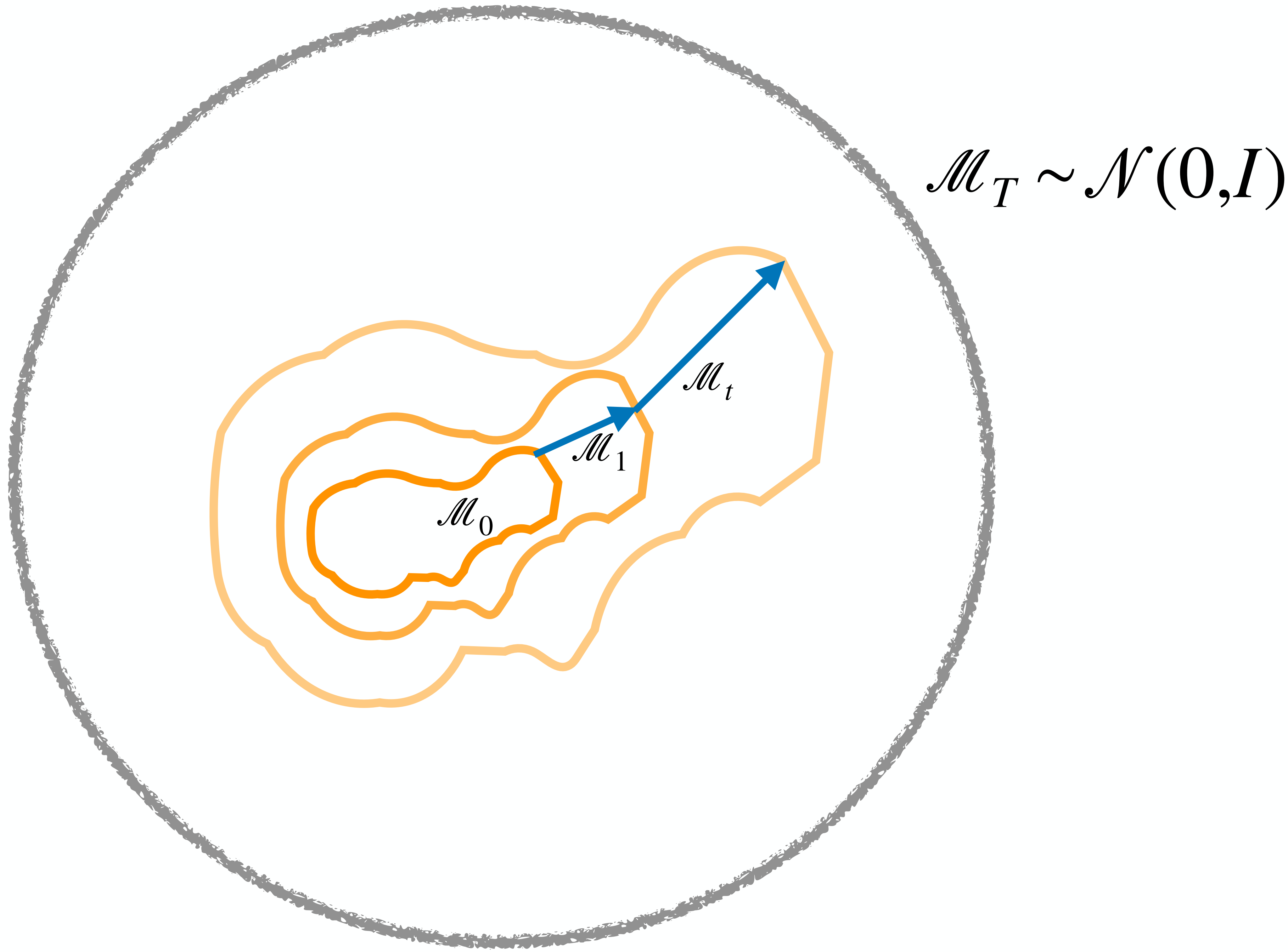
Geometric interpretation



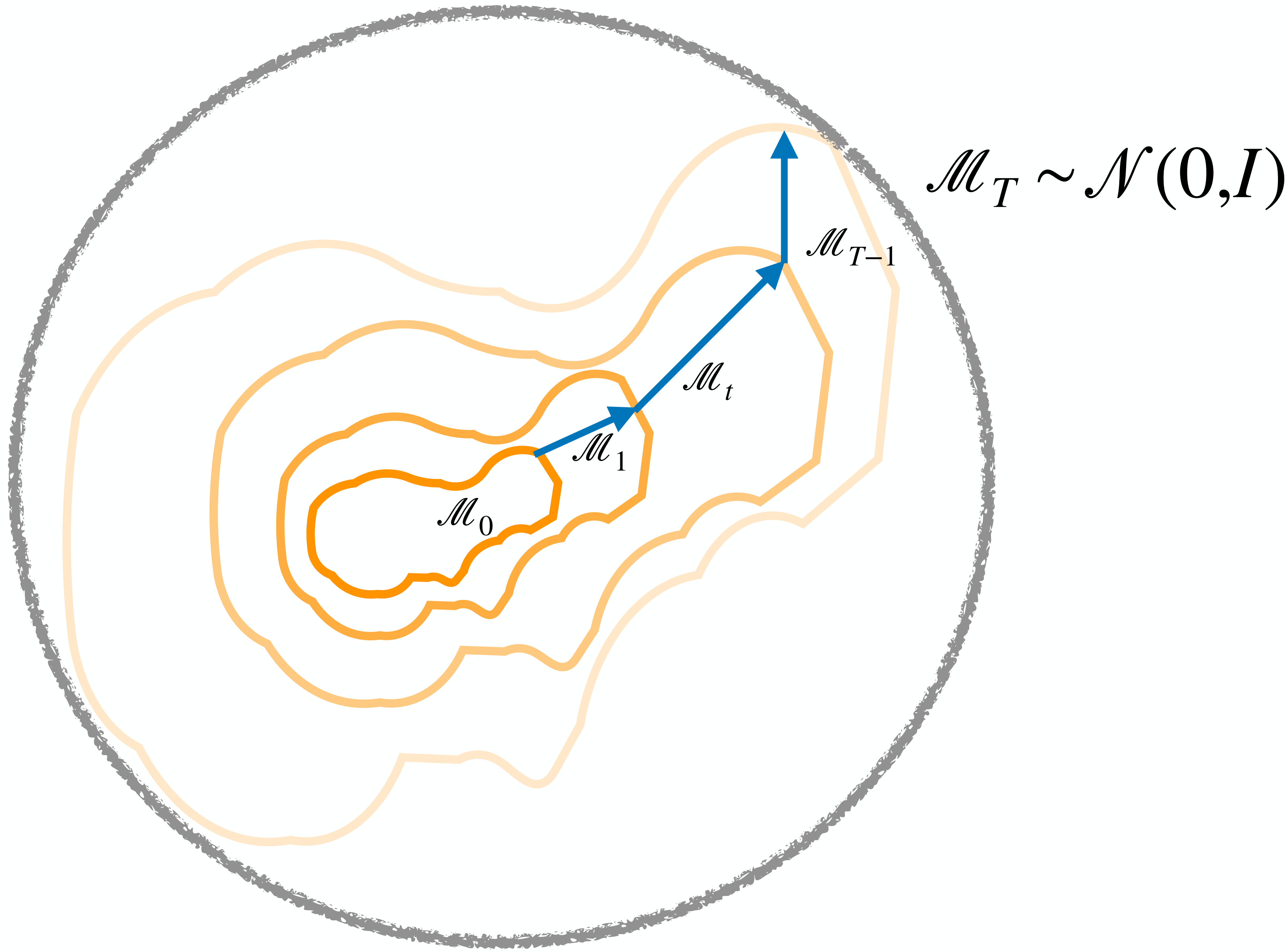
Geometric interpretation



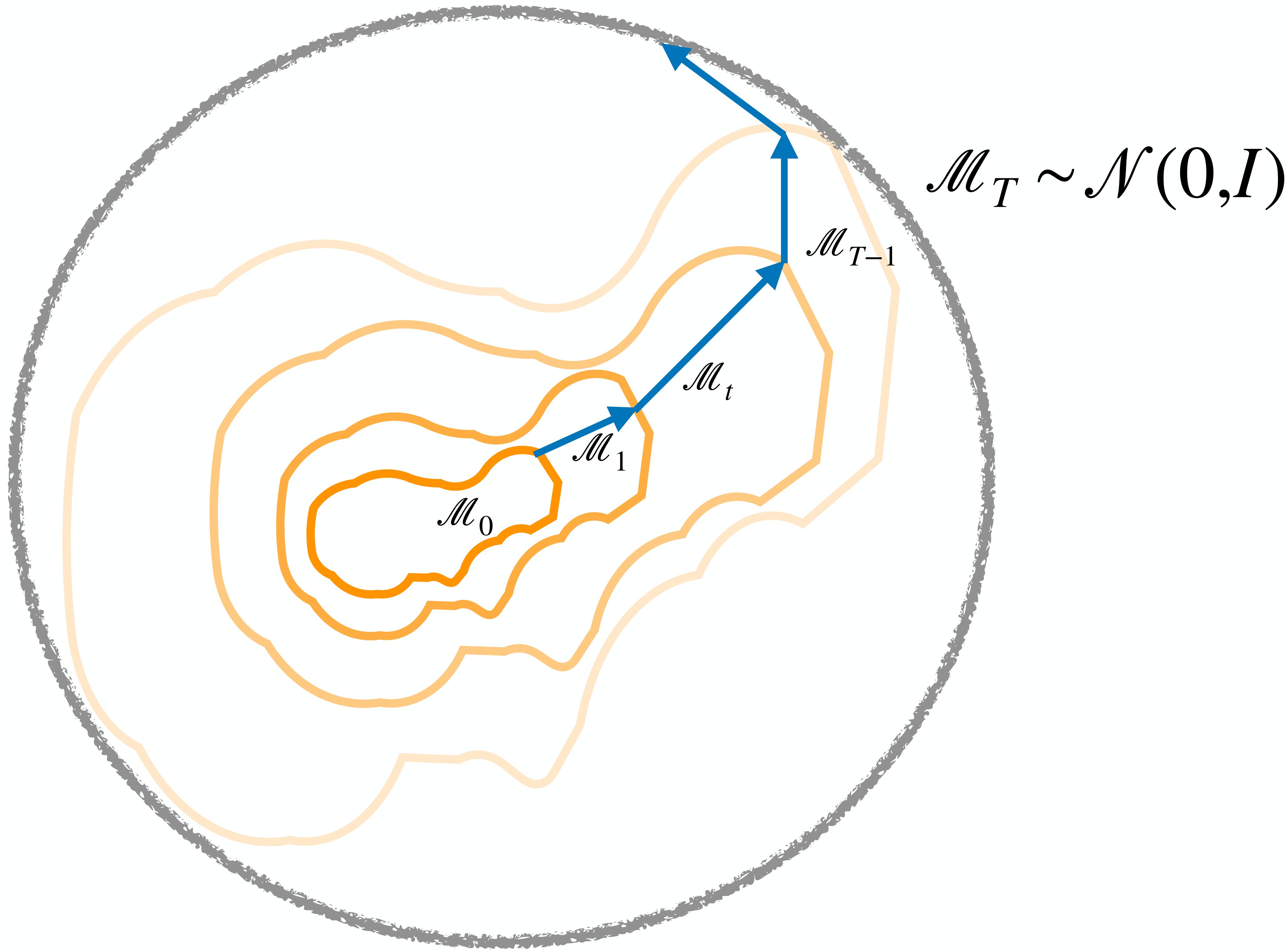
Geometric interpretation



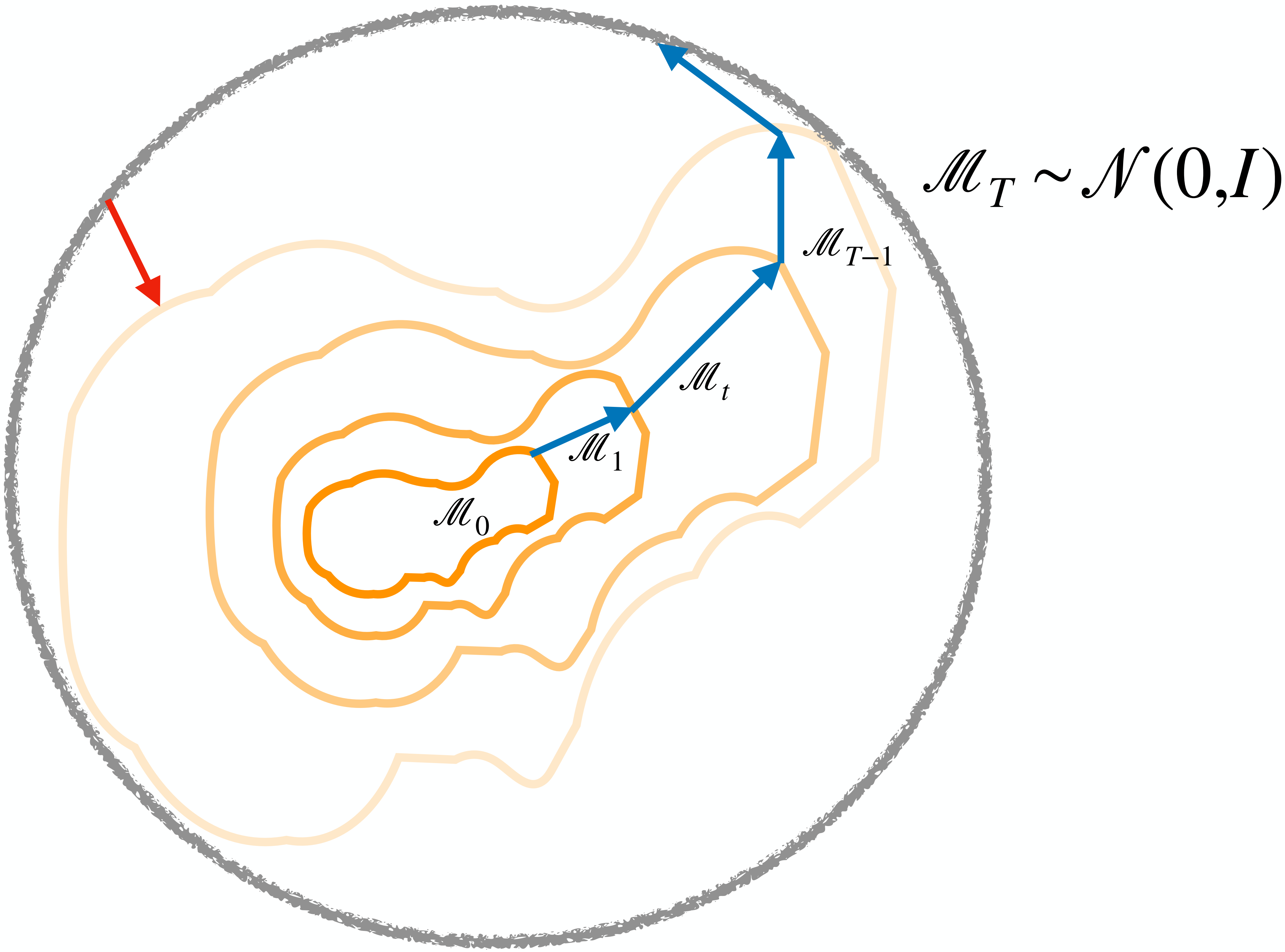
Geometric interpretation



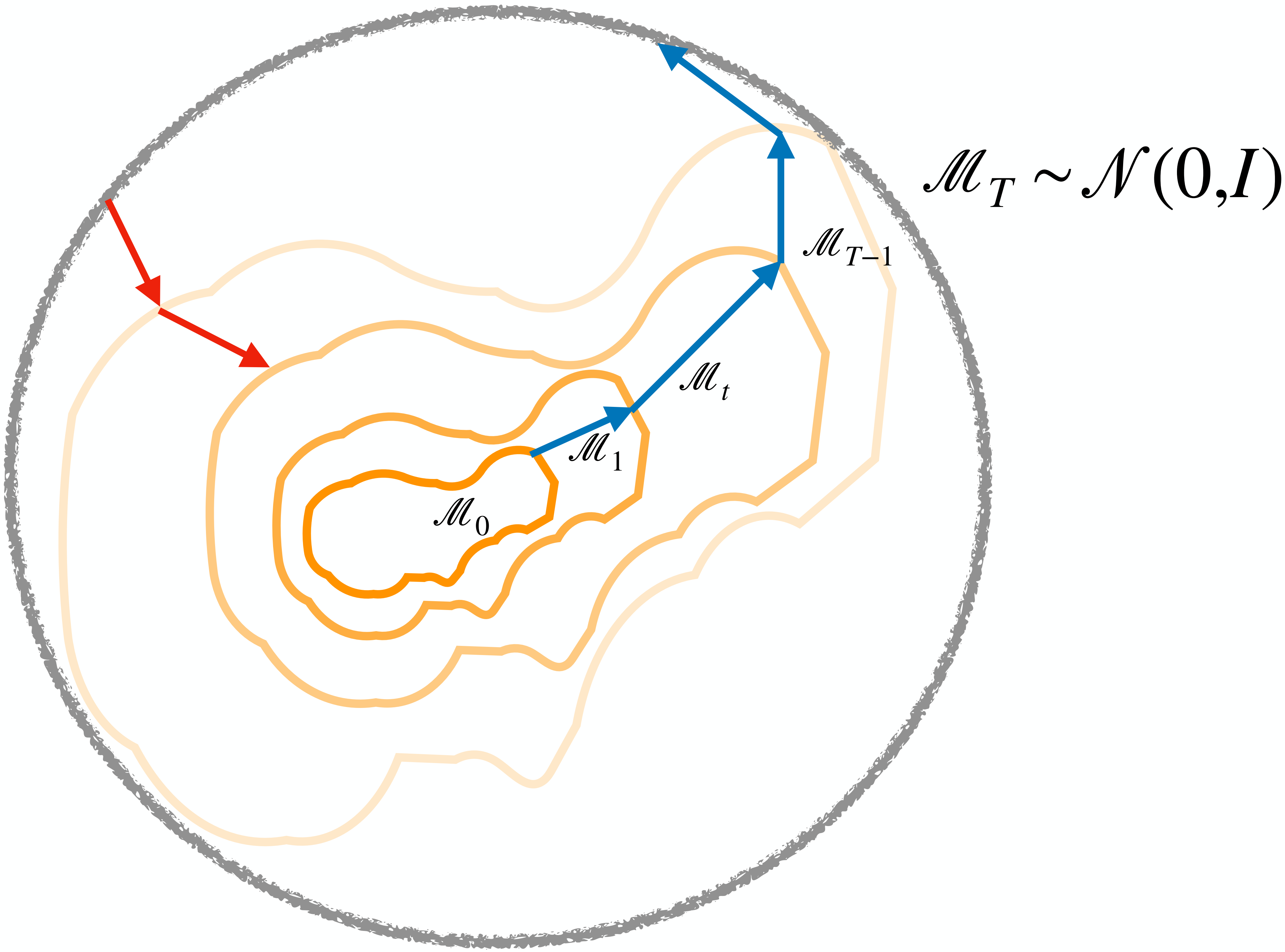
Geometric interpretation



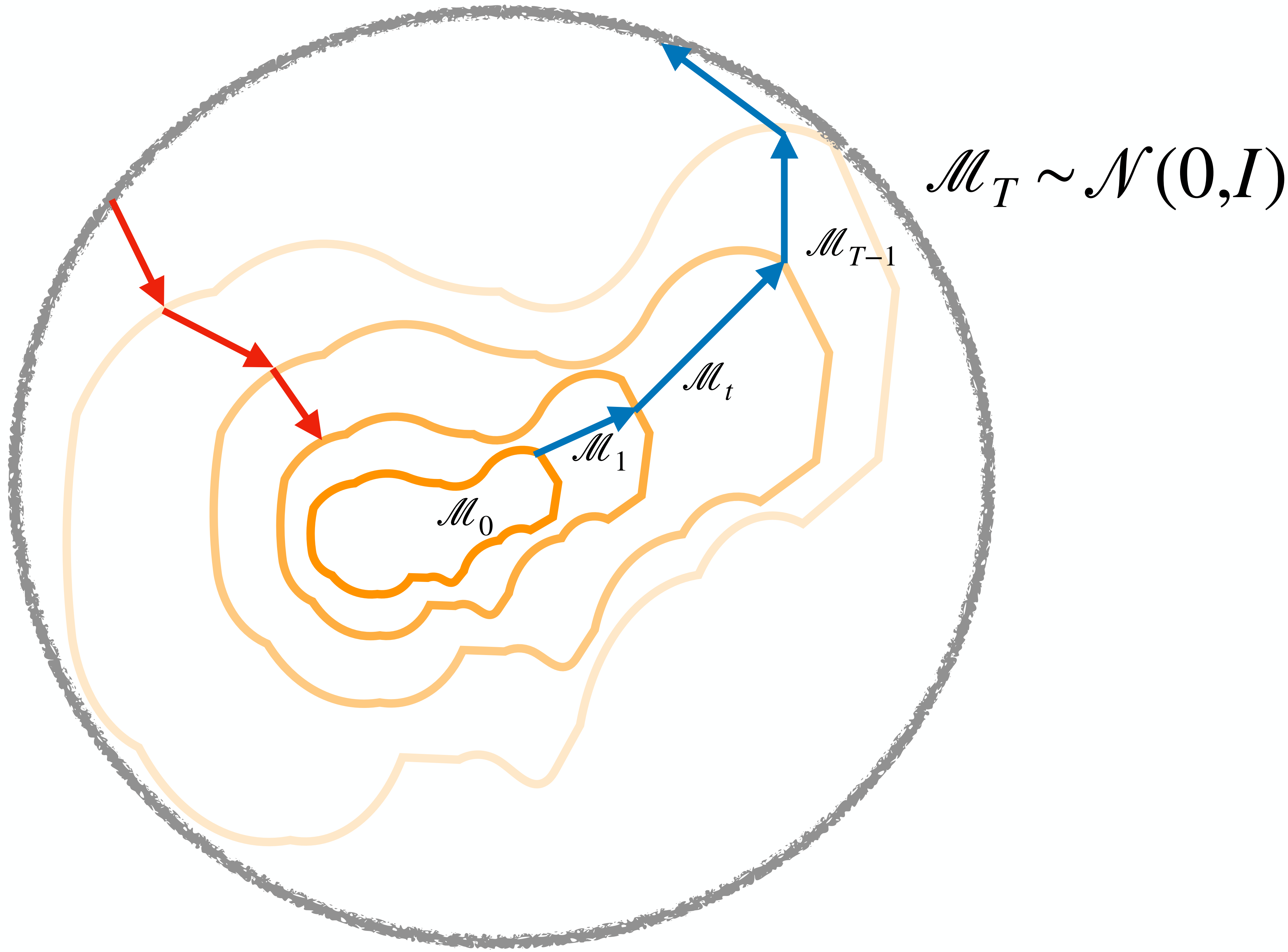
Geometric interpretation



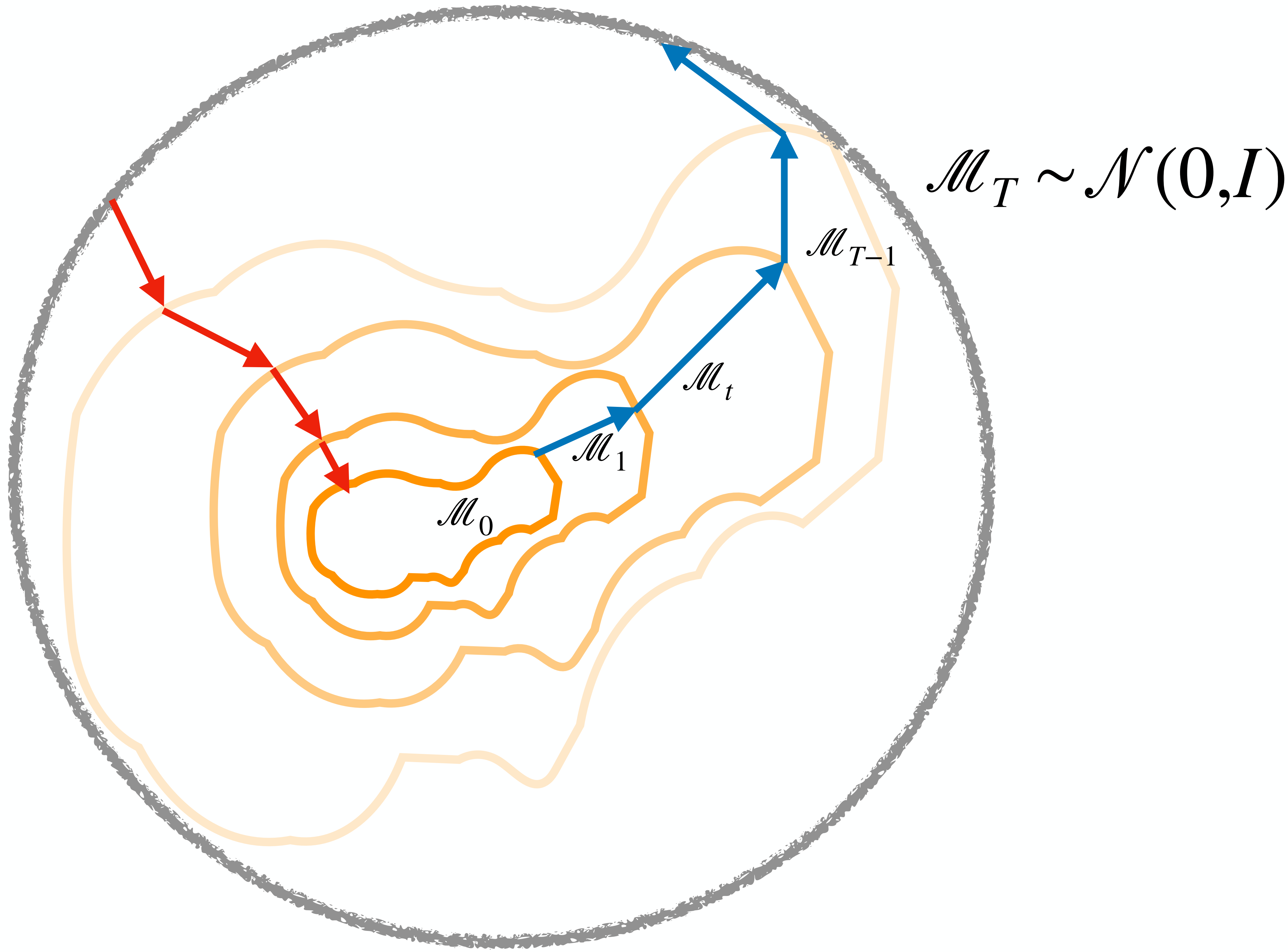
Geometric interpretation



Geometric interpretation



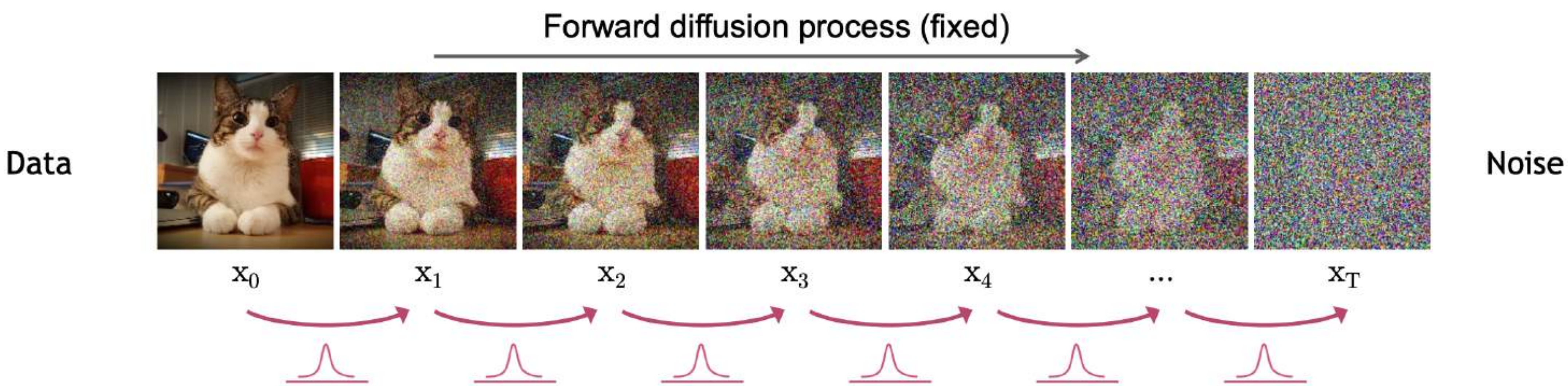
Geometric interpretation



Diffusion Models as Score Matching

Forward Diffusion

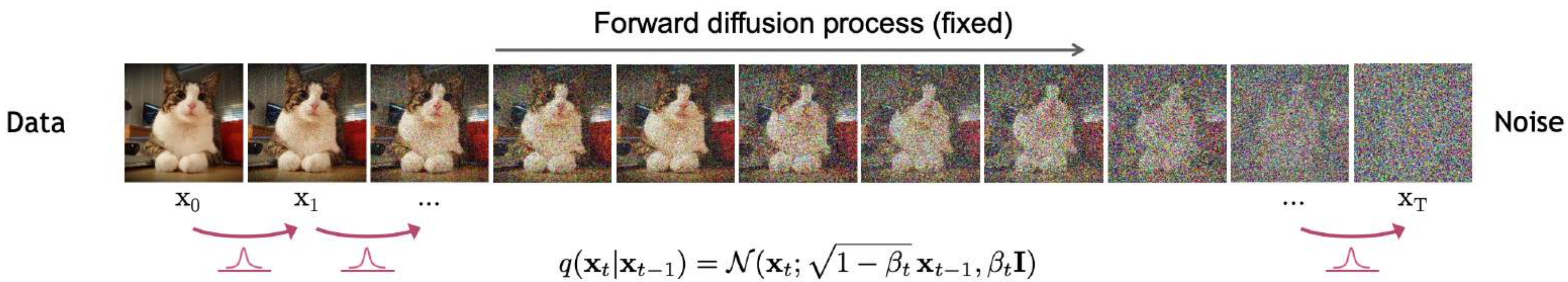
Consider the forward diffusion process again:



$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Forward Diffusion

Consider the limit of **many small steps:**



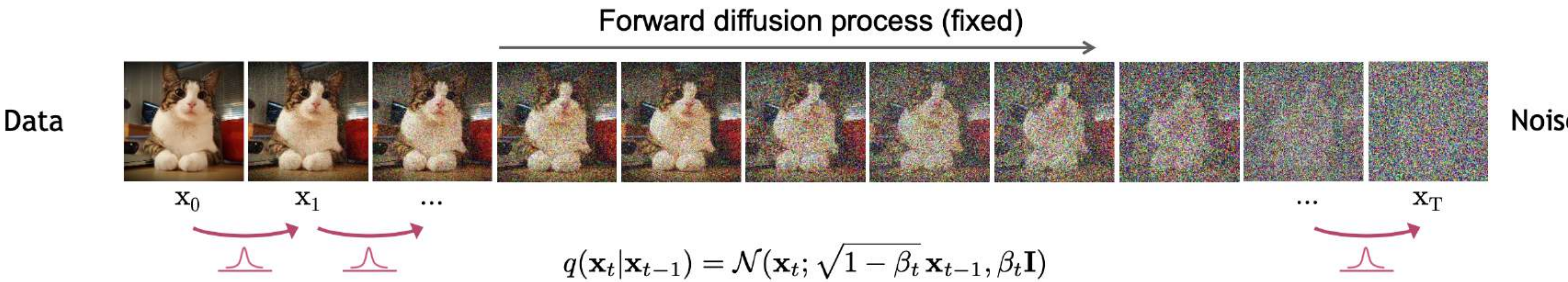
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

➡ $$\mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0},\mathbf{I})$$

Song et al., "Score-Based Generative Modeling through Stochastic Differential Equations", *ICLR*, 2021

Forward Diffusion

Consider the limit of **many small steps:**



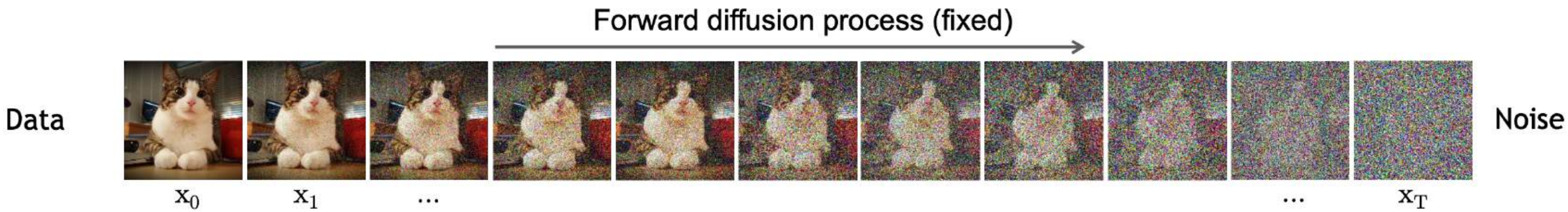
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$\begin{aligned} \mathbf{x}_t &= \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \\ \Rightarrow \quad &= \sqrt{1-\beta(t)\Delta t}\,\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \qquad (\beta_t := \beta(t)\Delta t) \end{aligned}$$

Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021

Forward Diffusion

Consider the limit of **many small steps:**



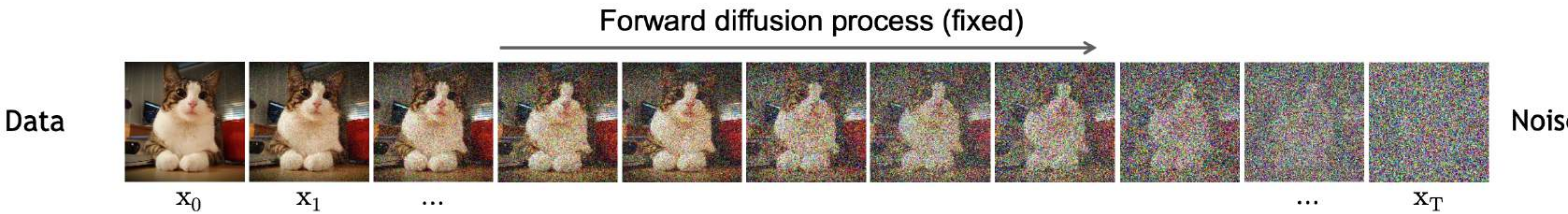
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \\
&= \sqrt{1-\beta(t)\Delta t}\,\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && (\beta_t := \beta(t)\Delta t) \\
&\approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && \text{(Taylor expansion)}
\end{aligned}$$

[Song et al., "Score-Based Generative Modeling through Stochastic Differential Equations", *ICLR*, 2021](#)

Forward Diffusion

Consider the limit of **many small steps:**



$$\mathbf{x}_t \approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I})$$

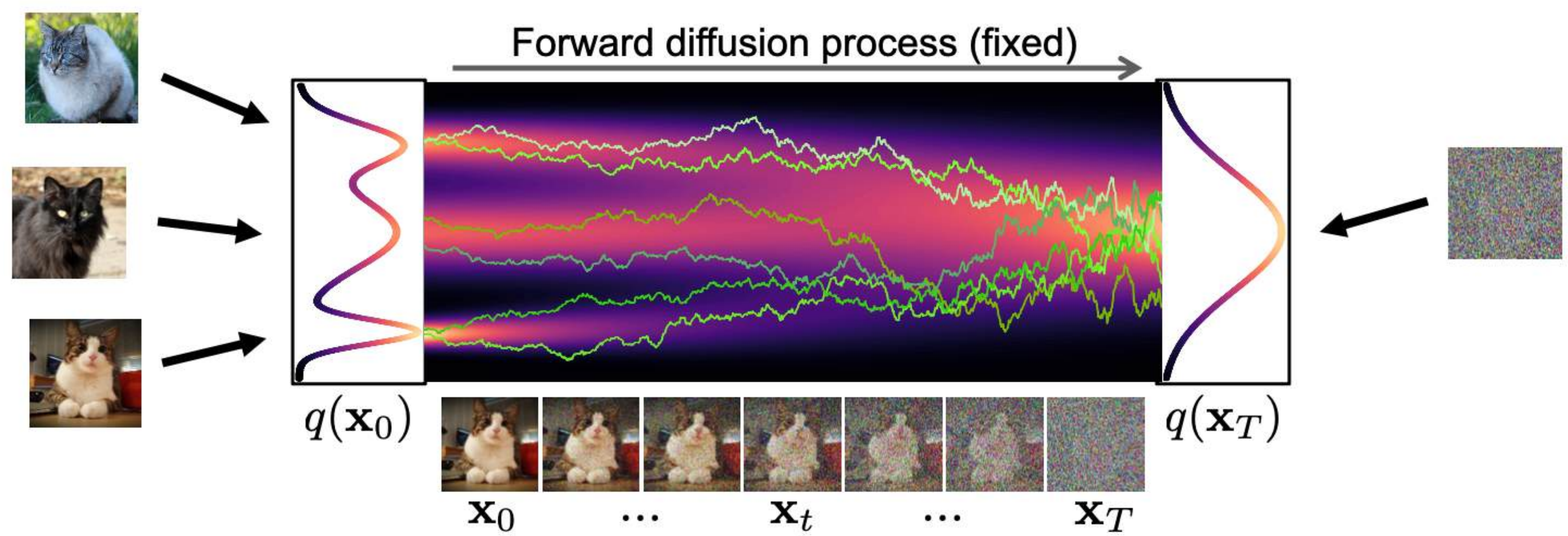
$$\Downarrow$$

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Stochastic Differential Equation (SDE)
describing the diffusion in infinitesimal limit

Song et al., "Score-Based Generative Modeling through Stochastic Differential Equations", *ICLR*, 2021

The Generative Reverse Stochastic Differential Equation



Forward Diffusion SDE: $$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Reverse Generative Diffusion SDE: $$\mathrm{d}\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}^{\text{drift term}}\mathrm{d}t + \overbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

➡ **Simulate reverse diffusion process: Data generation from random noise!**

[Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021]

Diffusion Models in Inverse Problems

General framework

$$y = \mathscr{A}(x_0) + z$$

Unconditional sampling

$$\hat{\mathbf{x}}_{t_{i-1}} = \boldsymbol{h}(\hat{\mathbf{x}}_{t_i}, \mathbf{z}_i, \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_{t_i}, t_i))$$

Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

General framework

$$y = \mathscr{A}(x_0) + z$$

Unconditional sampling

$$\hat{\mathbf{x}}_{t_{i-1}} = \boldsymbol{h}(\hat{\mathbf{x}}_{t_i}, \mathbf{z}_i, \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_{t_i}, t_i))$$

Measurement conditioned sampling

1) Data consistency step

$$\hat{\mathbf{x}}'_{t_i} = \boldsymbol{k}(\hat{\mathbf{x}}_{t_i}, \hat{\mathbf{y}}_{t_i}, \lambda)$$

2) Unconditional diffusion step

$$\hat{\mathbf{x}}_{t_{i-1}} = \boldsymbol{h}(\hat{\mathbf{x}}'_{t_i}, \mathbf{z}_i, \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_{t_i}, t_i)), \quad i = N, N-1, \cdots, 1$$

Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

General framework

$$y = \mathscr{A}(x_0) + z$$

Unconditional sampling

$$\hat{\mathbf{x}}_{t_{i-1}} = \boldsymbol{h}(\hat{\mathbf{x}}_{t_i}, \mathbf{z}_i, \boldsymbol{s}_{\boldsymbol{\theta}^*}(\hat{\mathbf{x}}_{t_i}, t_i))$$

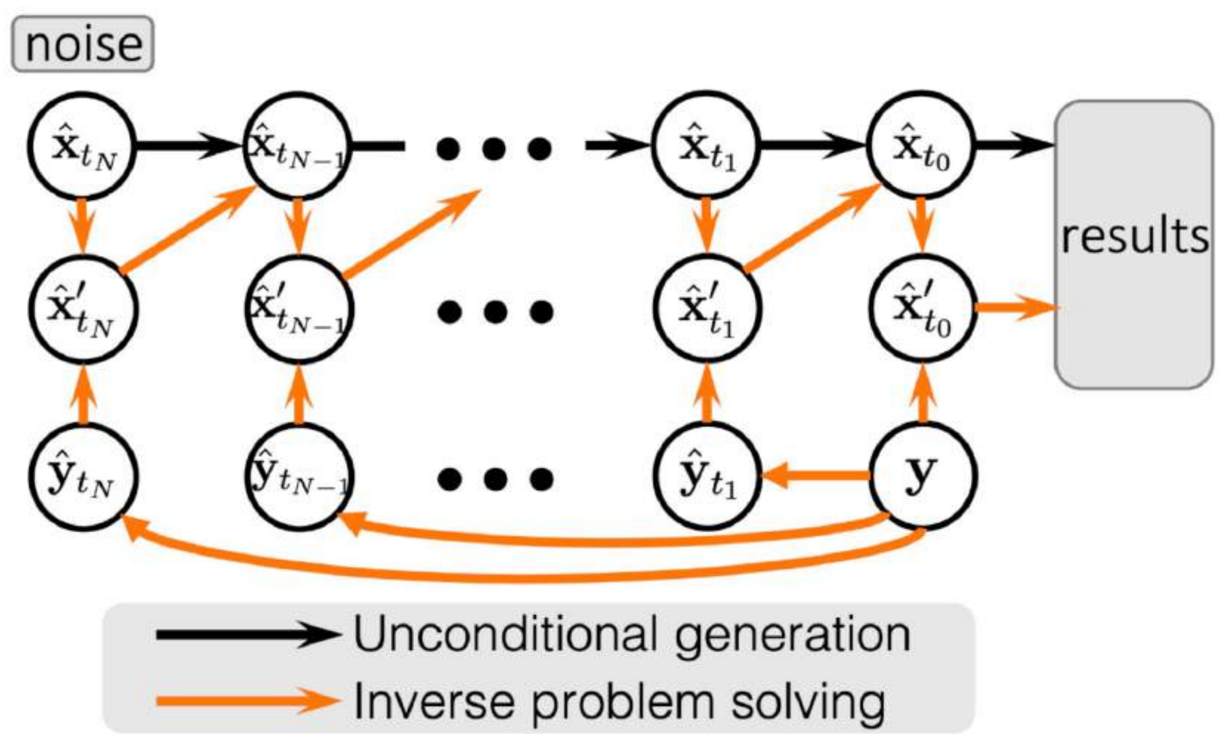
Measurement conditioned sampling

1) Data consistency step

$$\hat{\mathbf{x}}'_{t_i} = \boldsymbol{k}(\hat{\mathbf{x}}_{t_i}, \hat{\mathbf{y}}_{t_i}, \lambda)$$

2) Unconditional diffusion step

$$\hat{\mathbf{x}}_{t_{i-1}} = \boldsymbol{h}(\hat{\mathbf{x}}'_{t_i}, \mathbf{z}_i, \boldsymbol{s}_{\boldsymbol{\theta}^*}(\hat{\mathbf{x}}_{t_i}, t_i)), \quad i = N, N-1, \cdots, 1$$



Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

Medical imaging example

Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

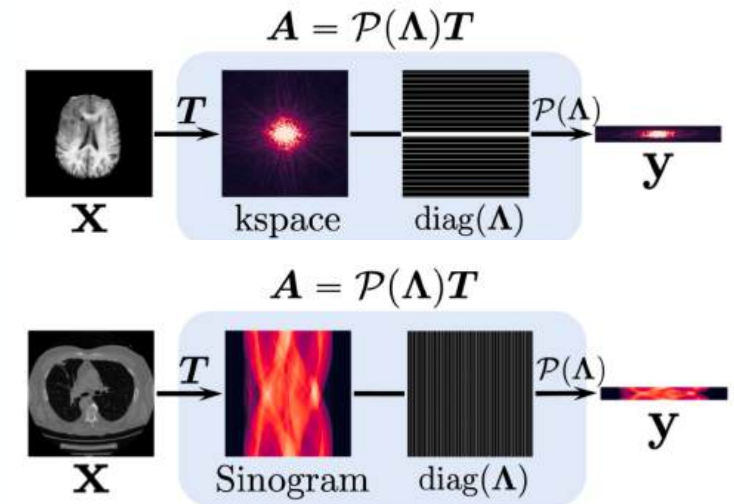
Medical imaging example

Assume that the forward model has the form:

$$\mathscr{A}(x_0) = \mathscr{P}(\Lambda)Tx_0$$

subsampling

invertible



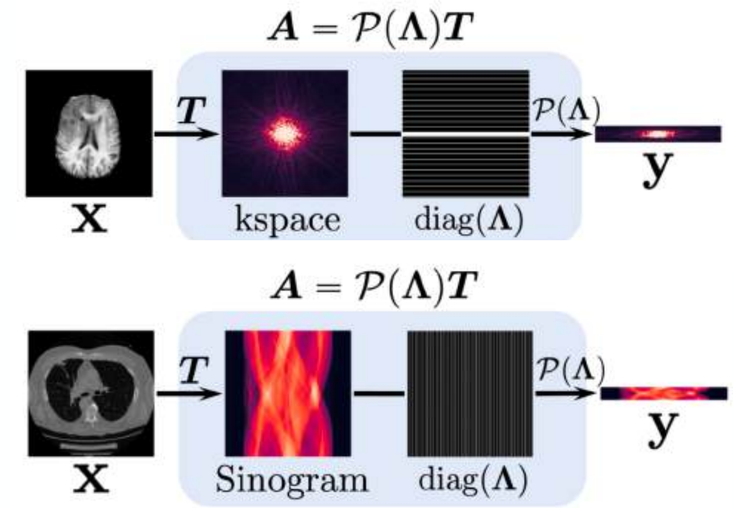
Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

Medical imaging example

Assume that the forward model has the form:

$$\mathscr{A}(x_0) = \mathscr{P}(\Lambda)Tx_0$$

subsampling invertible



Measurement conditioned sampling

1) Data consistency

$$\hat{\mathbf{x}}'_{t_i} = \boldsymbol{k}(\hat{\mathbf{x}}_{t_i}, \hat{\mathbf{y}}_{t_i}, \lambda)$$

2) Unconditional diffusion

$$\hat{\mathbf{x}}_{t_{i-1}} = \boldsymbol{h}(\hat{\mathbf{x}}'_{t_i}, \mathbf{z}_i, \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_{t_i}, t_i)), \quad i = N, N-1, \cdots, 1$$

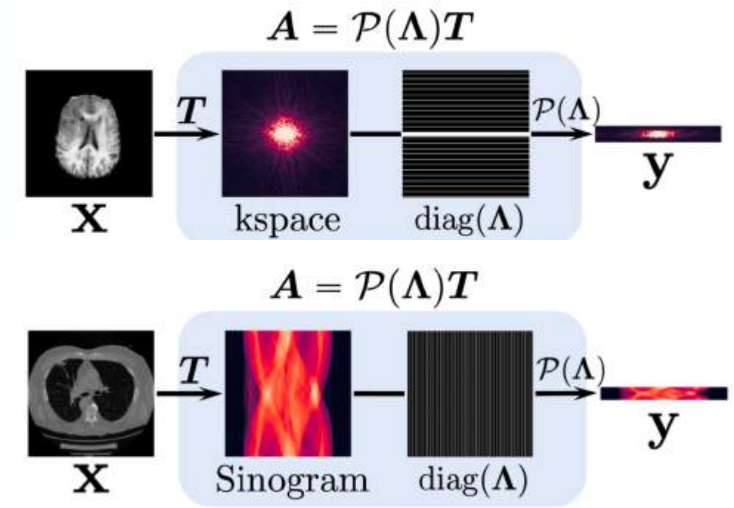
Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

Medical imaging example

Assume that the forward model has the form:

$$\mathscr{A}(x_0) = \mathscr{P}(\Lambda) T x_0$$

subsampling invertible



Measurement conditioned sampling

1) Data consistency

$$\hat{\mathbf{x}}'_{t_i} = \boldsymbol{k}(\hat{\mathbf{x}}_{t_i}, \hat{\mathbf{y}}_{t_i}, \lambda)$$

2) Unconditional diffusion

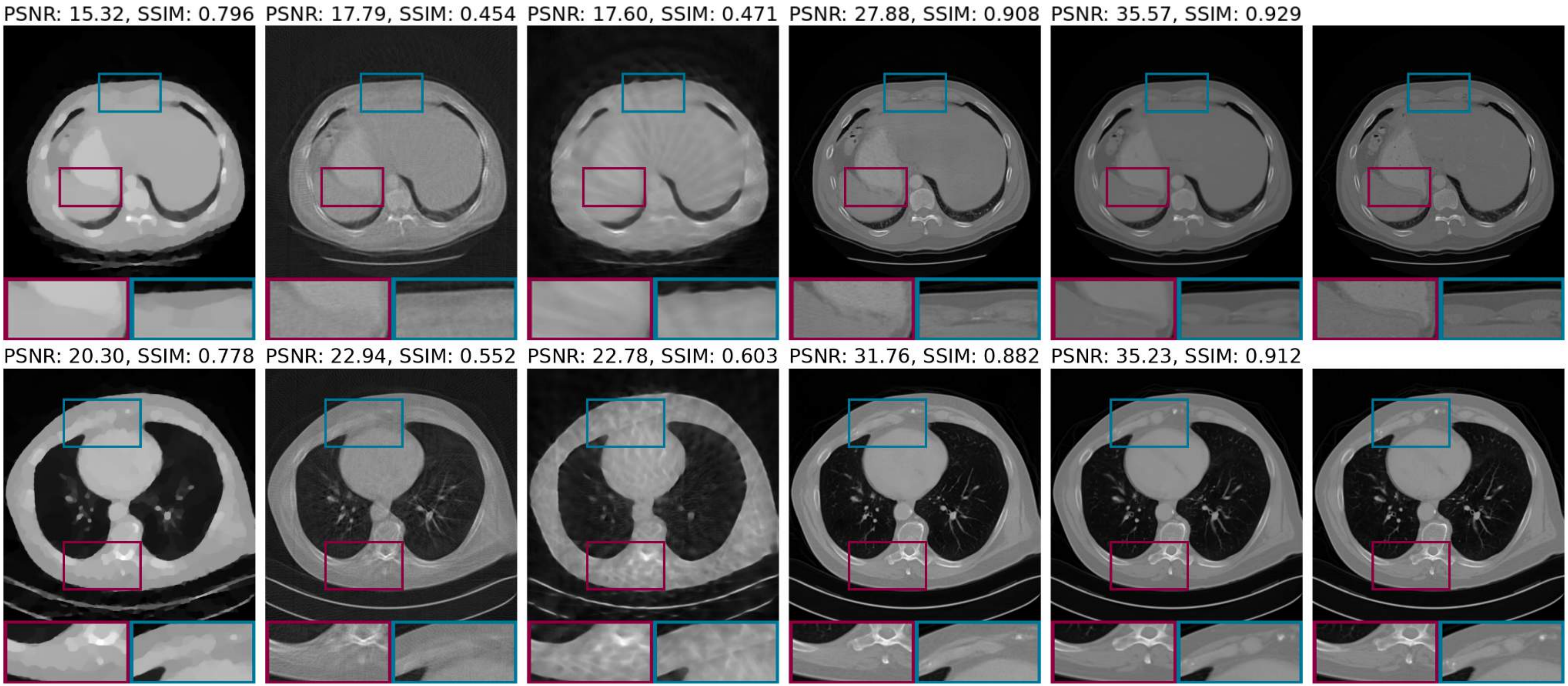
$$\hat{\mathbf{x}}_{t_{i-1}} = \boldsymbol{h}(\hat{\mathbf{x}}'_{t_i}, \mathbf{z}_i, \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_{t_i}, t_i)), \quad i = N, N-1, \cdots, 1$$

Data consistency:

$$\hat{\mathbf{x}}'_{t_i} = \underset{\boldsymbol{z}\in\mathbb{R}^n}{\arg\min}\{(1-\lambda)\,\|\boldsymbol{z} - \hat{\mathbf{x}}_{t_i}\|^2_{\boldsymbol{T}} + \min_{\boldsymbol{u}\in\mathbb{R}^n} \lambda\,\|\boldsymbol{z} - \boldsymbol{u}\|^2_{\boldsymbol{T}}\} \quad s.t. \quad \boldsymbol{A}\boldsymbol{u} = \hat{\mathbf{y}}_{t_i}$$

Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

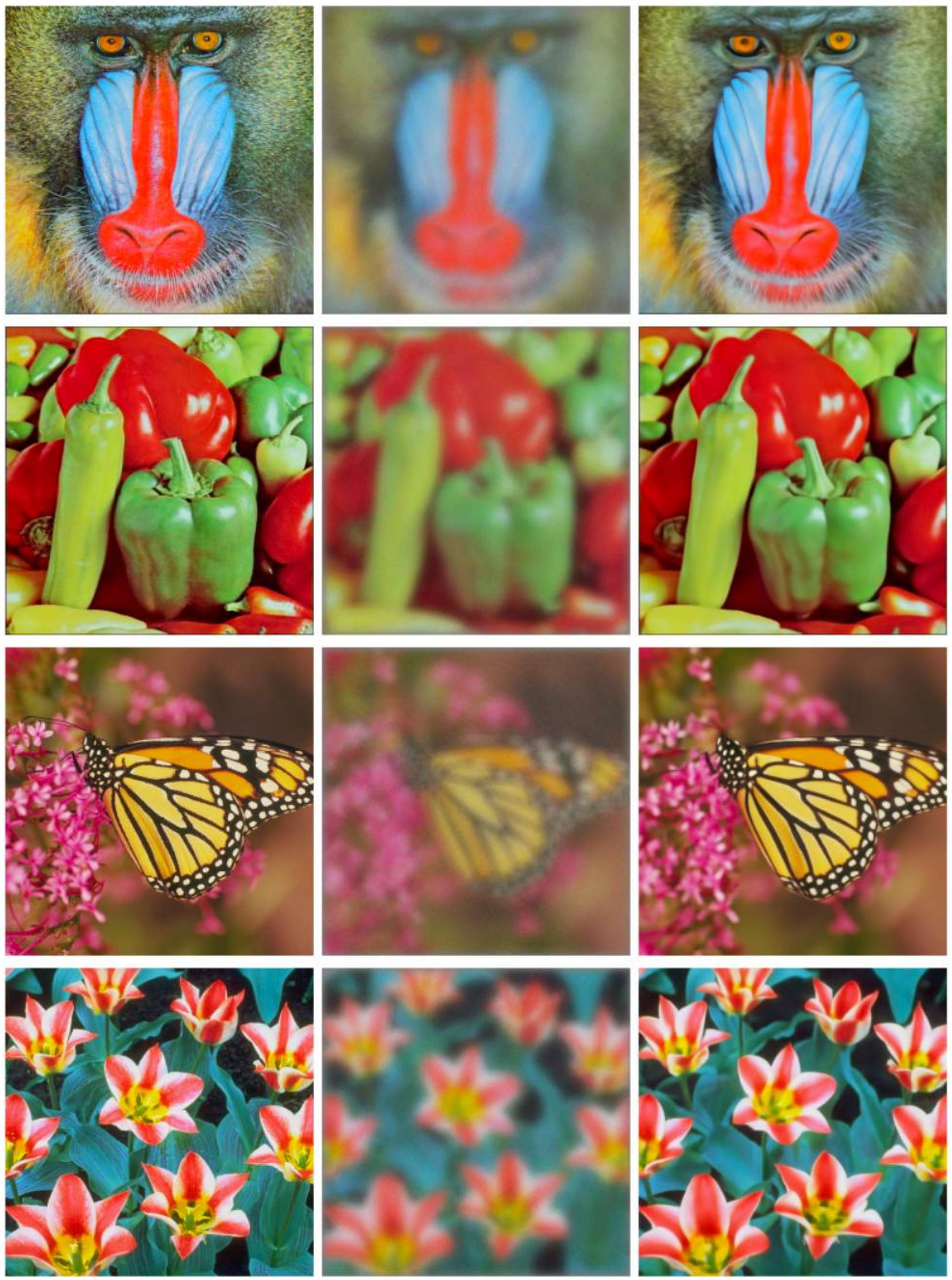
Reconstruction results



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

Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

Deblurring and inpainting results

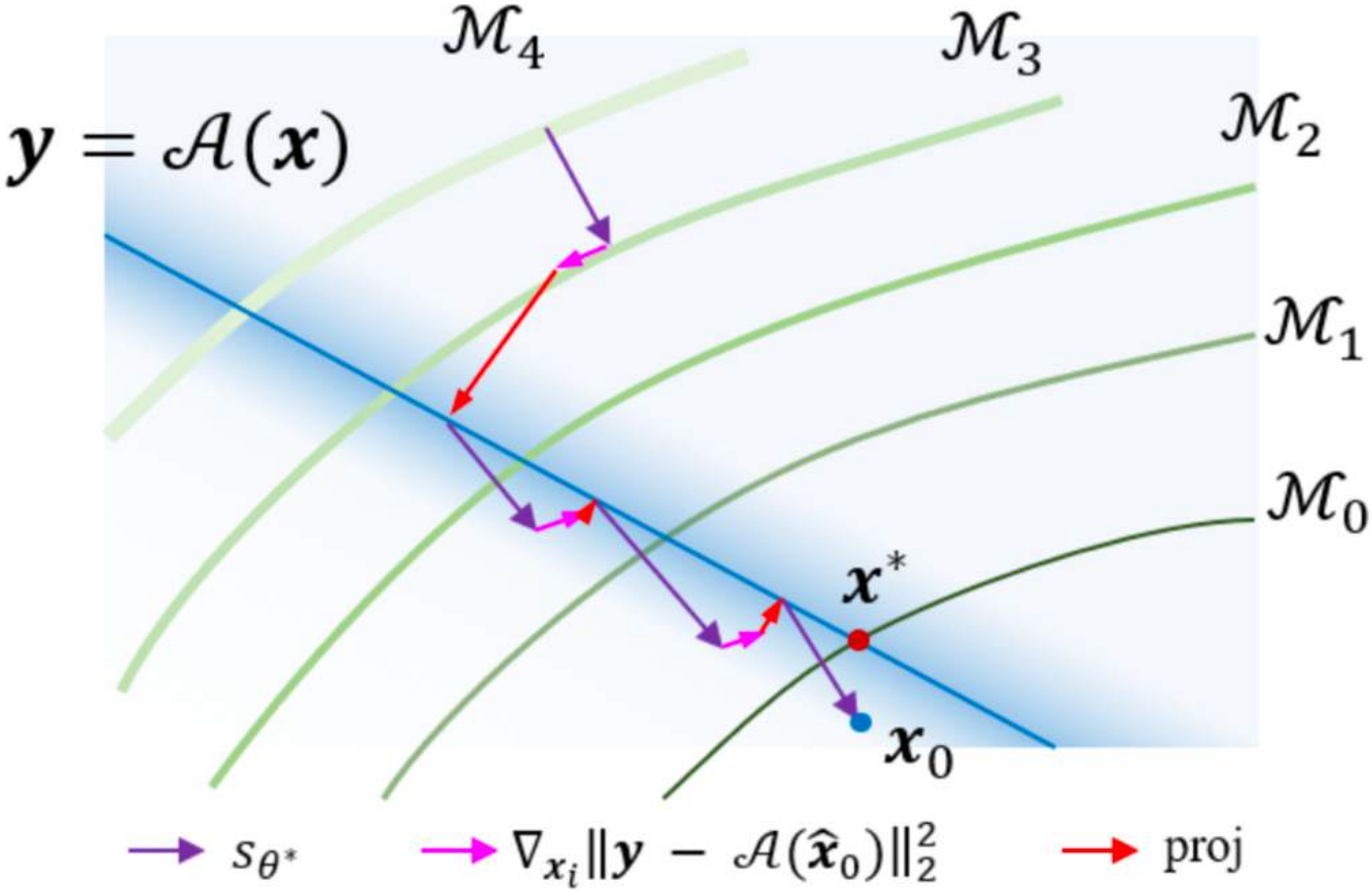


Original Blurred DDRM (20)

Original Occluded DDRM (20)

Kawar, Bahjat, et al. "Denoising diffusion restoration models." *arXiv preprint arXiv:2201.11793* (2022).

Intuition



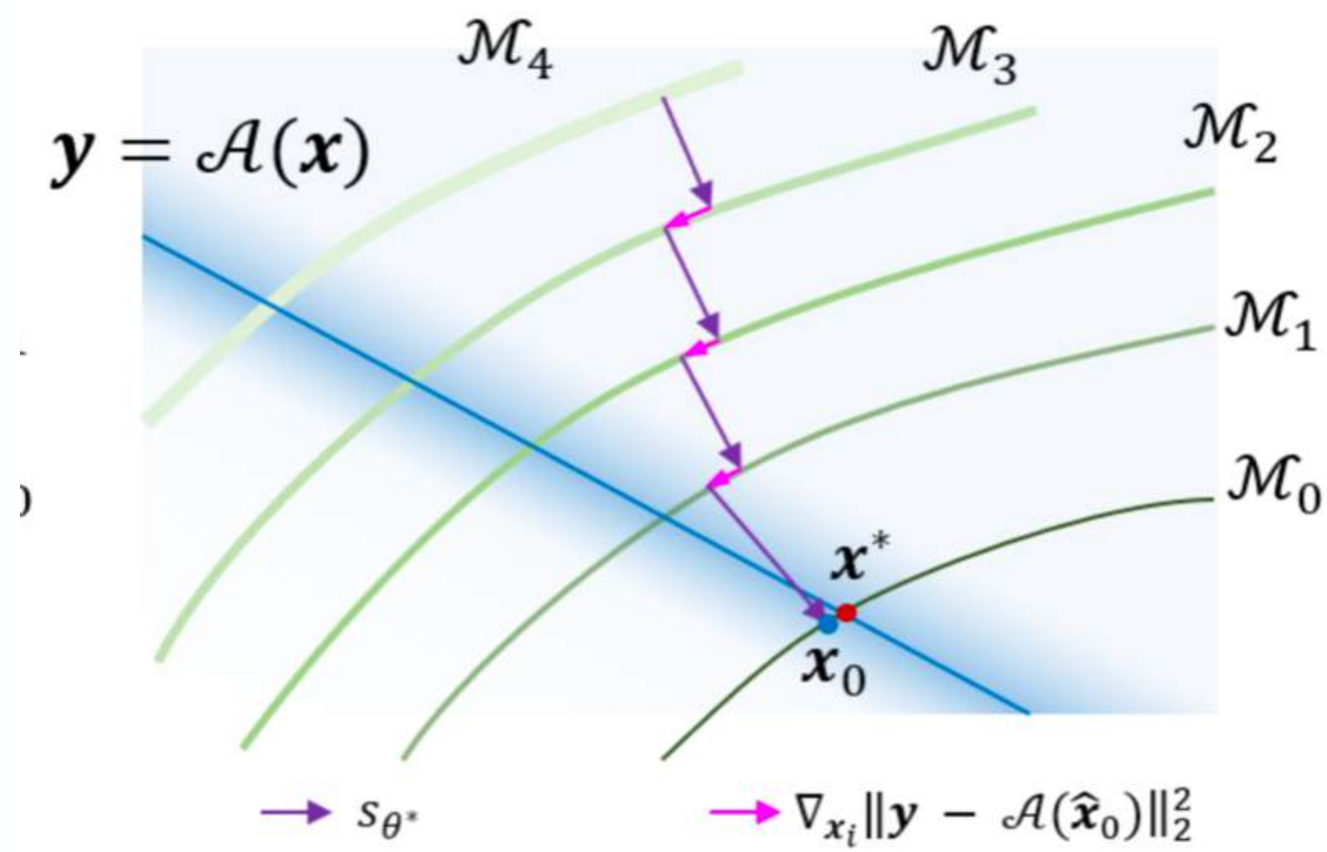
Song, Yang, et al. "Solving inverse problems in medical imaging with score-based generative models." *arXiv preprint arXiv:2111.08005* (2021).

Diffusion Posterior Sampling

- **Bayesian framework**

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{n} \qquad\qquad p_\theta(\boldsymbol{x}|\boldsymbol{y}) \propto p_\theta(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})$$

solve inverse problem = sample from posterior



Diffusion Posterior Sampling

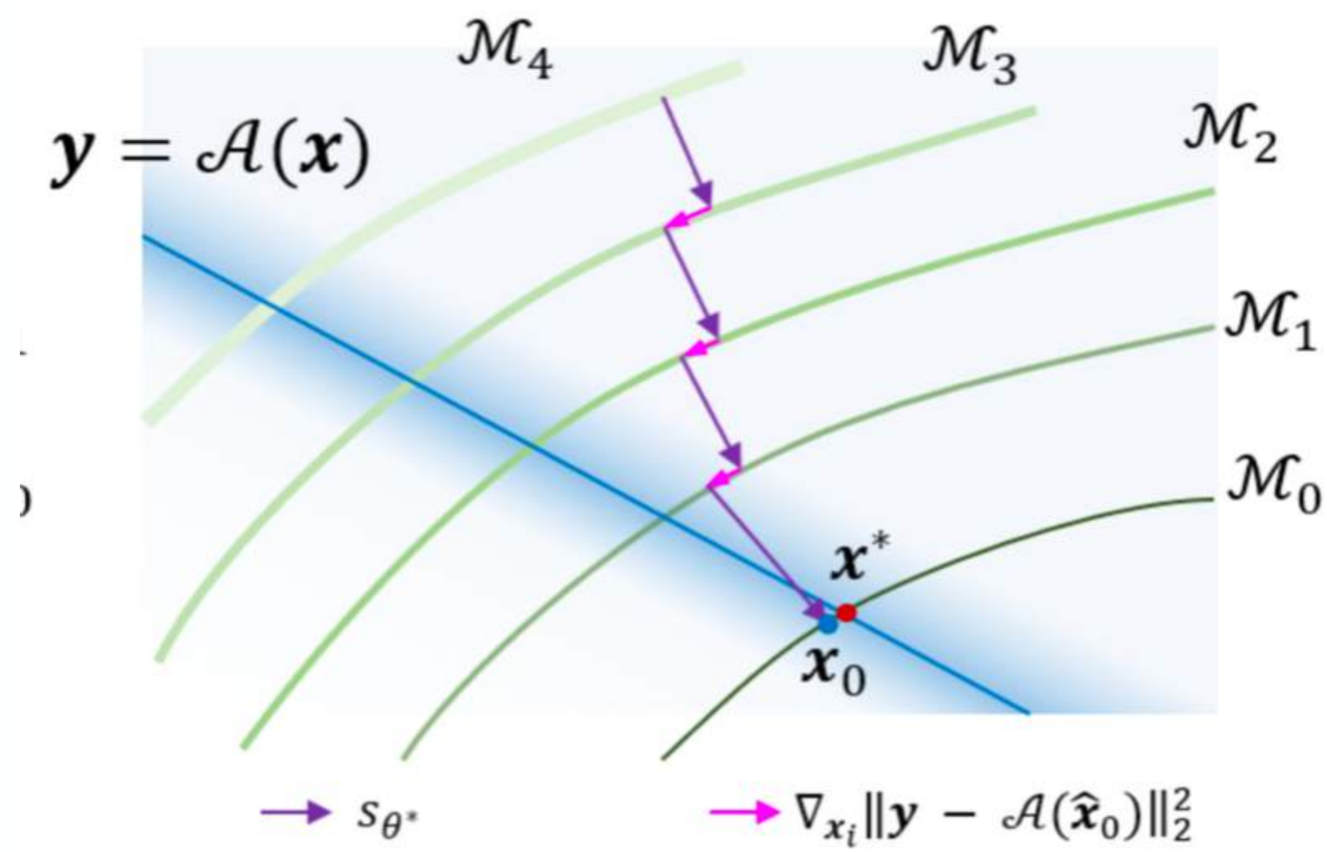
- **Bayesian framework**

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{n} \qquad\qquad p_\theta(\boldsymbol{x}|\boldsymbol{y}) \propto p_\theta(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})$$

solve inverse problem = sample from posterior

- **Hijacking the diffusion process**

$$\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}|\boldsymbol{y}) = \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\boldsymbol{x})$$



Diffusion Posterior Sampling

- **Bayesian framework**

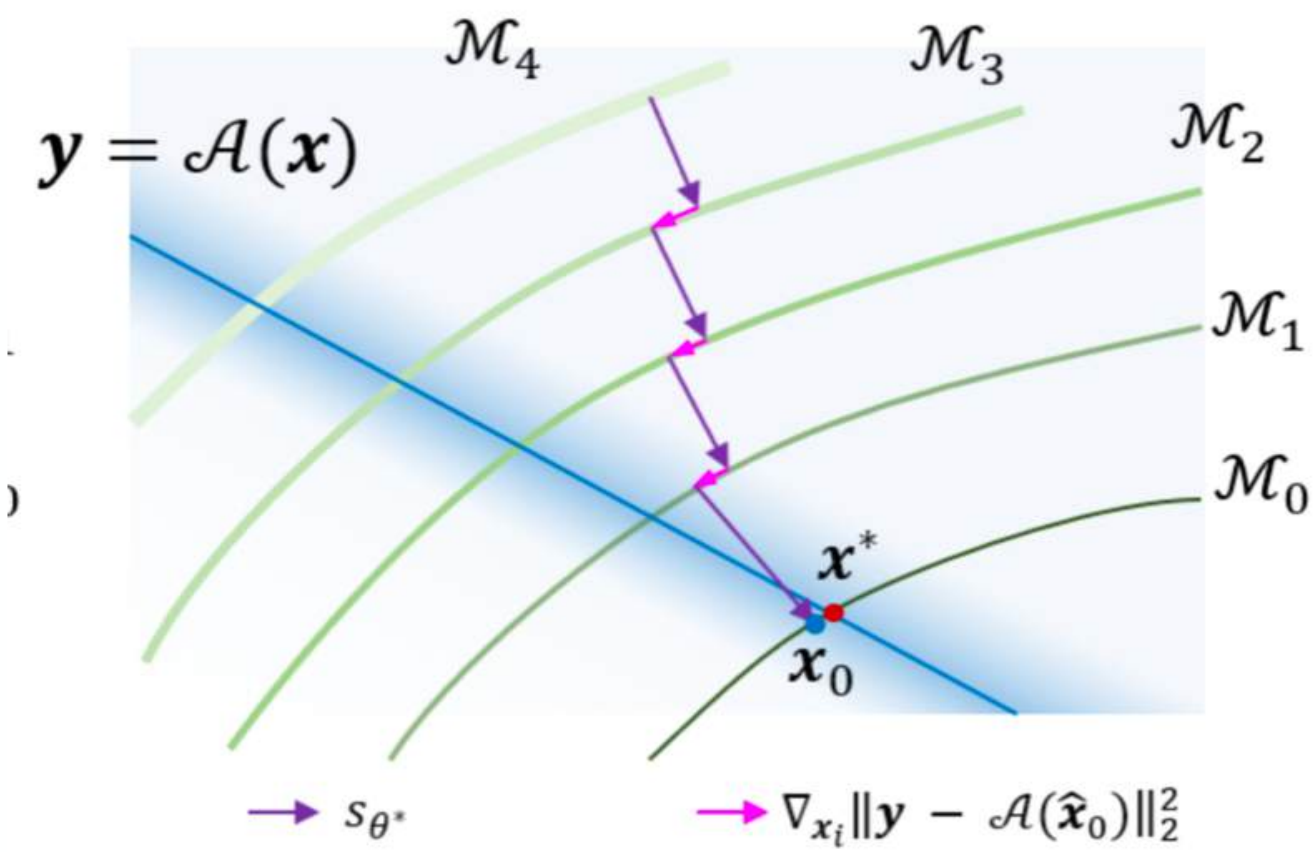
$$y = \mathcal{A}(x) + n \qquad\qquad p_\theta(x|y) \propto p_\theta(x)p(y|x)$$

solve inverse problem = sample from posterior

- **Hijacking the diffusion process**

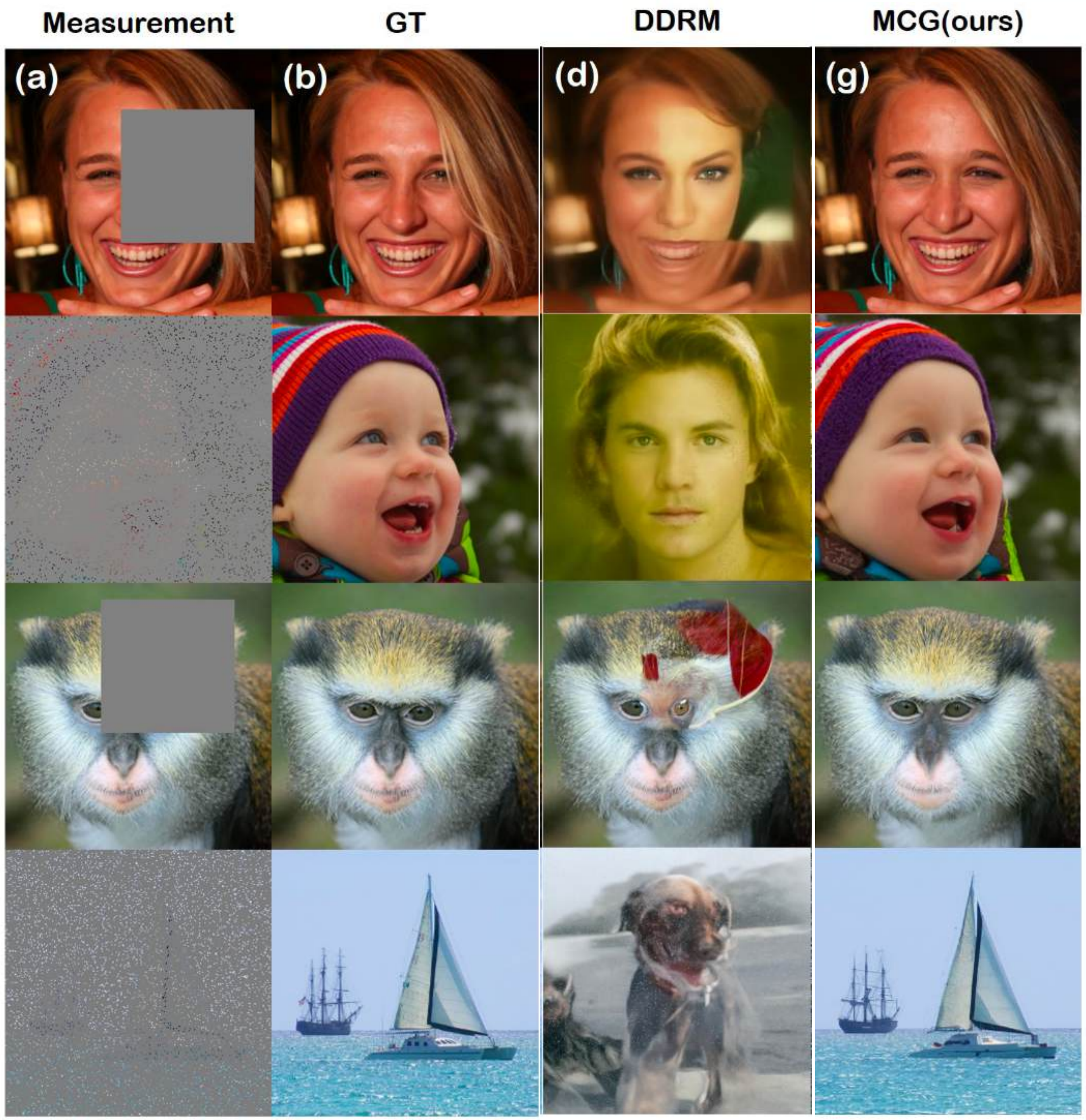
$$\nabla_x \log p_\theta(x|y) = \nabla_x \log p_\theta(x) + \nabla_x \log p(y|x)$$

prior = pre-trained diffusion model



Chung, Hyungjin, et al. "Diffusion posterior sampling for general noisy inverse problems." *arXiv preprint arXiv:2209.14687* (2022).

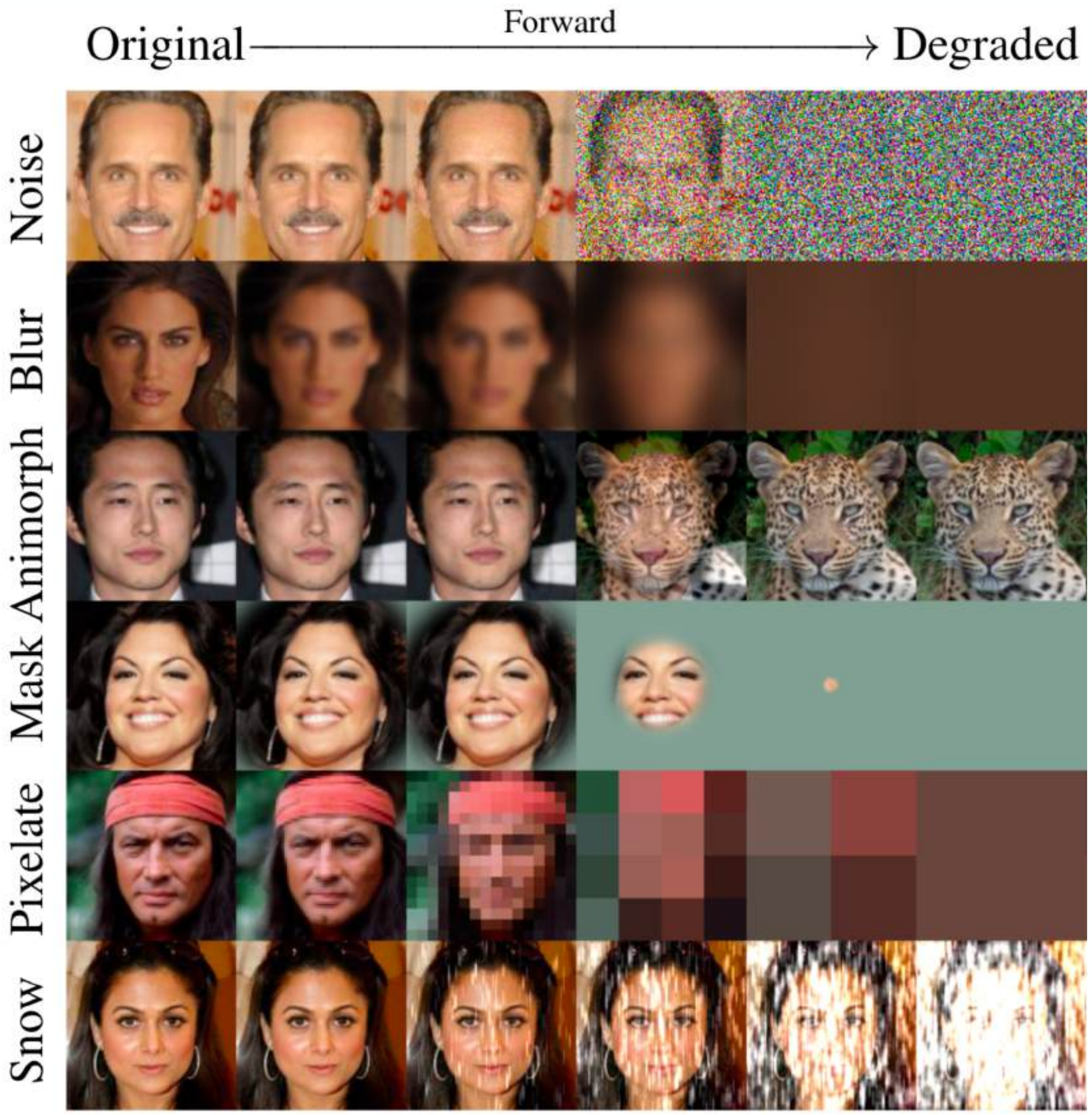
Diffusion posterior sampling



Chung, H., Sim, B., Ryu, D. and Ye, J.C., 2022. Improving Diffusion Models for Inverse Problems using Manifold Constraints. *arXiv preprint arXiv:2206.00941*.

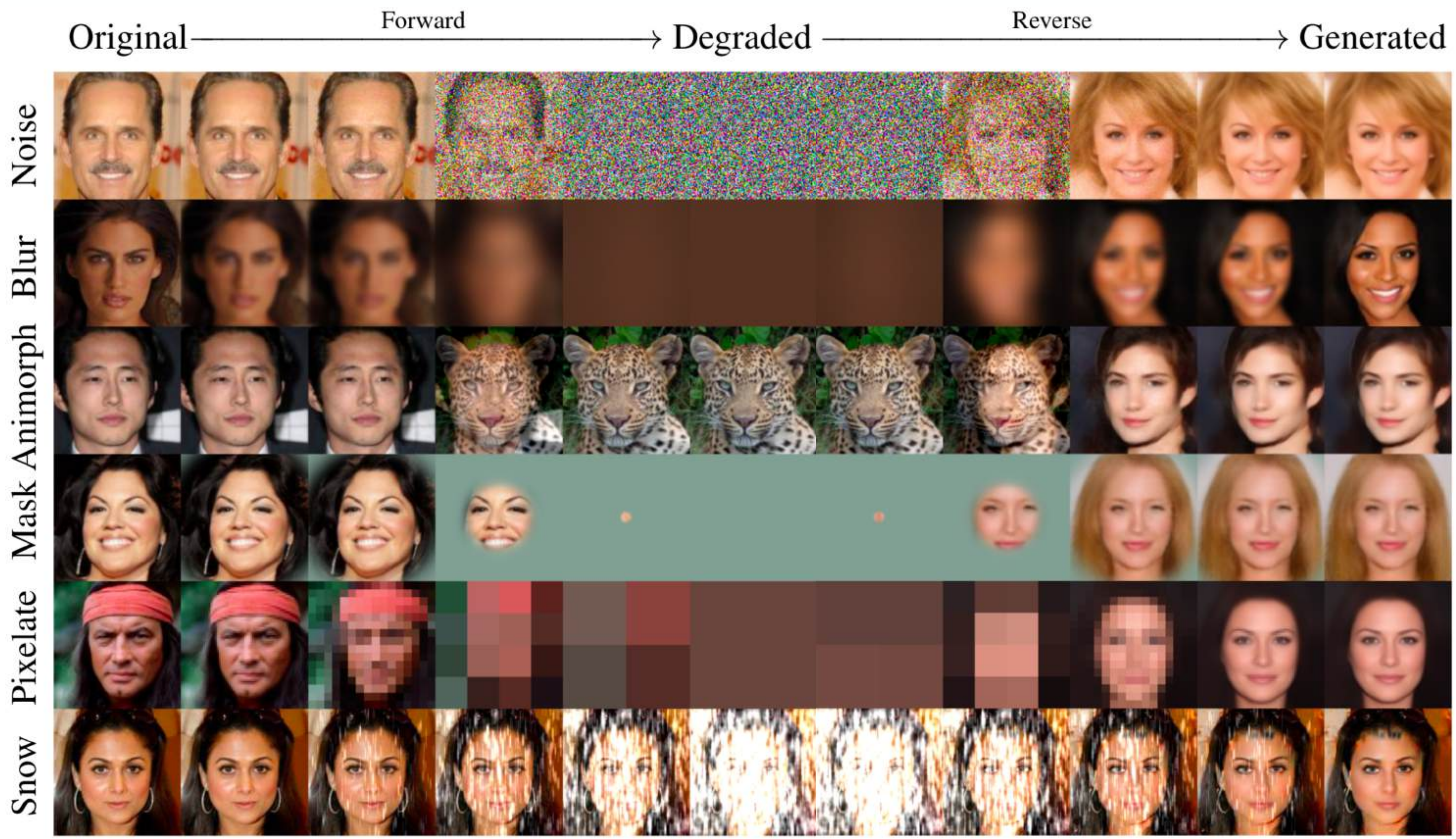
Dirac Diffusion

Cold diffusion



Bansal, Arpit, et al. "Cold diffusion: Inverting arbitrary image transforms without noise." *arXiv preprint arXiv:2208.09392* (2022).

Cold diffusion



Bansal, Arpit, et al. "Cold diffusion: Inverting arbitrary image transforms without noise." *arXiv preprint arXiv:2208.09392* (2022).

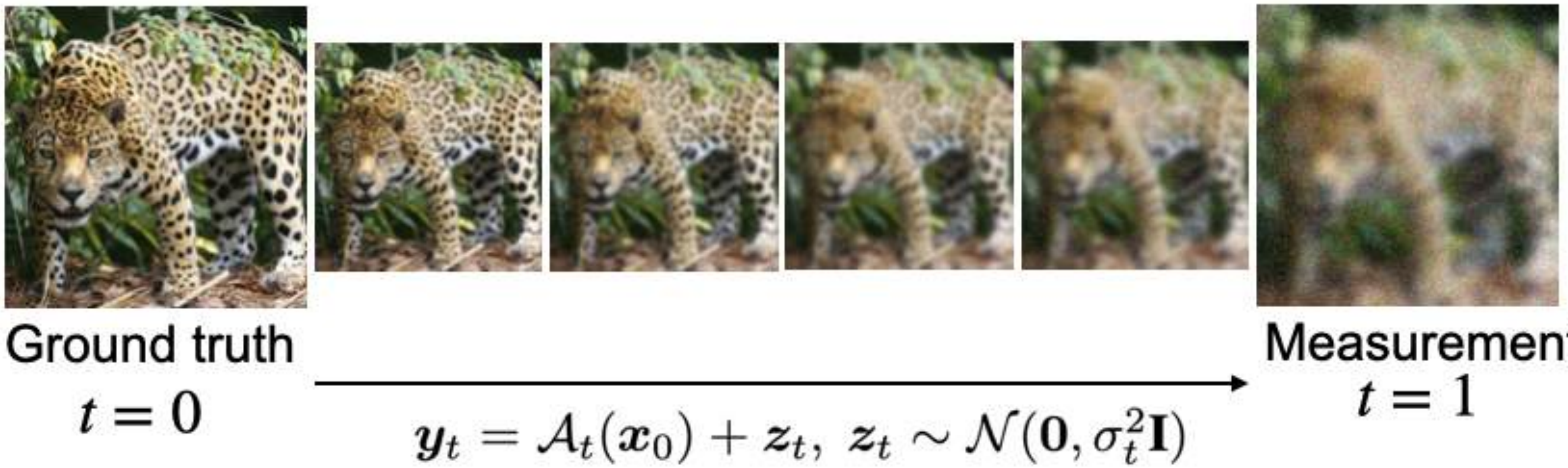
Stochastic Degradation Process (SDP)

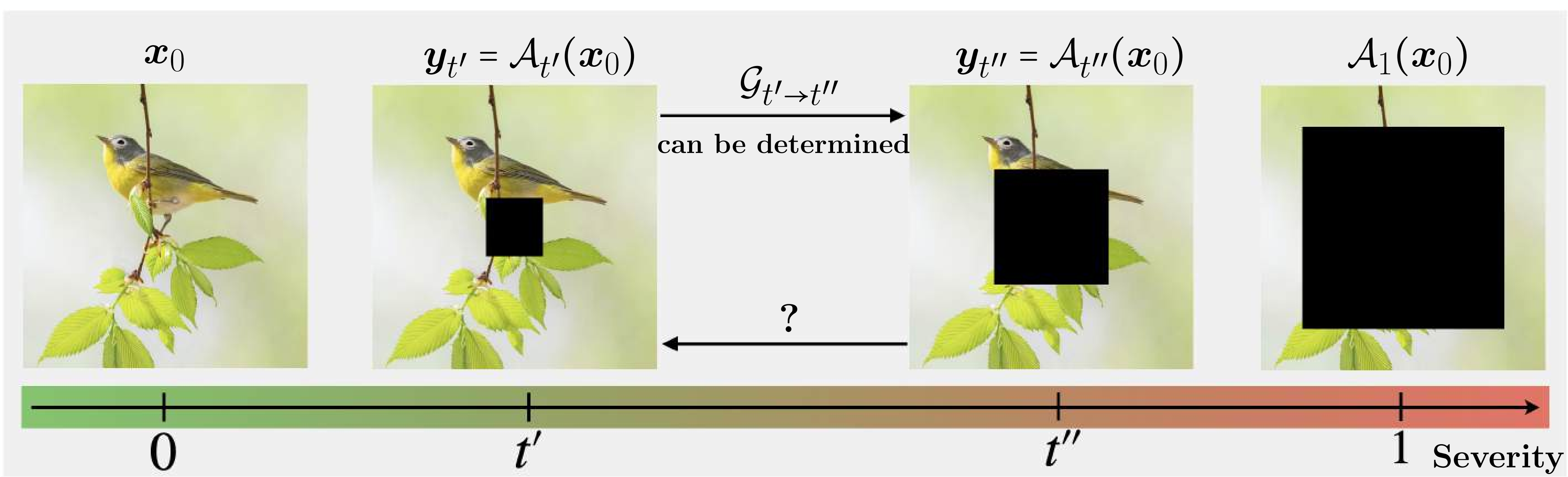
Ground truth $t = 0$

Measurement $t = 1$

$$\boldsymbol{y}_t = \mathcal{A}_t(\boldsymbol{x}_0) + \boldsymbol{z}_t,\ \boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I})$$

Degradation Severity



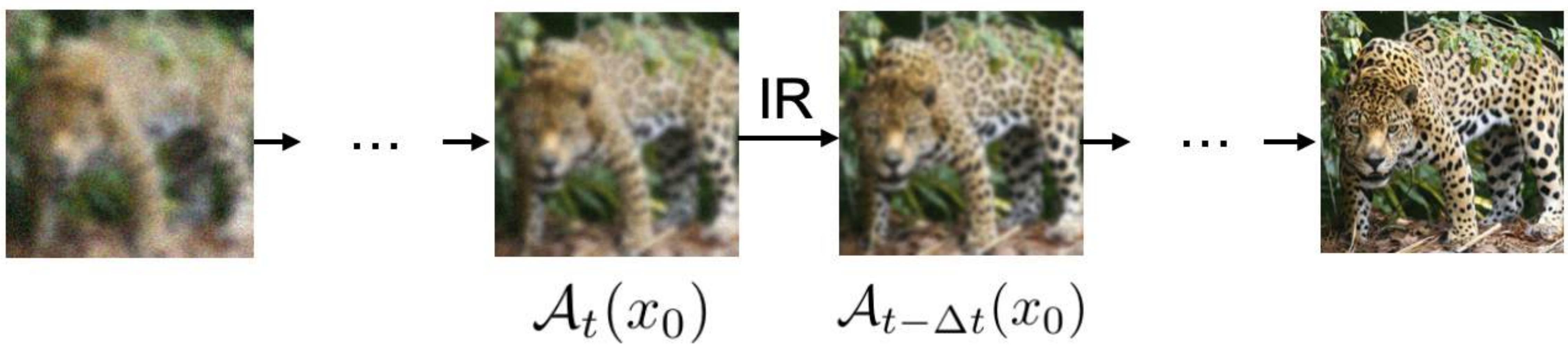


<u>Property 1</u>: $\mathcal{A}_t$ is more severe than $\mathcal{A}_{t'}$, $\forall t > t'$

<u>Property 2</u>: $\mathcal{A}_0(\mathbf{x_0}) = \mathbf{x_0}$

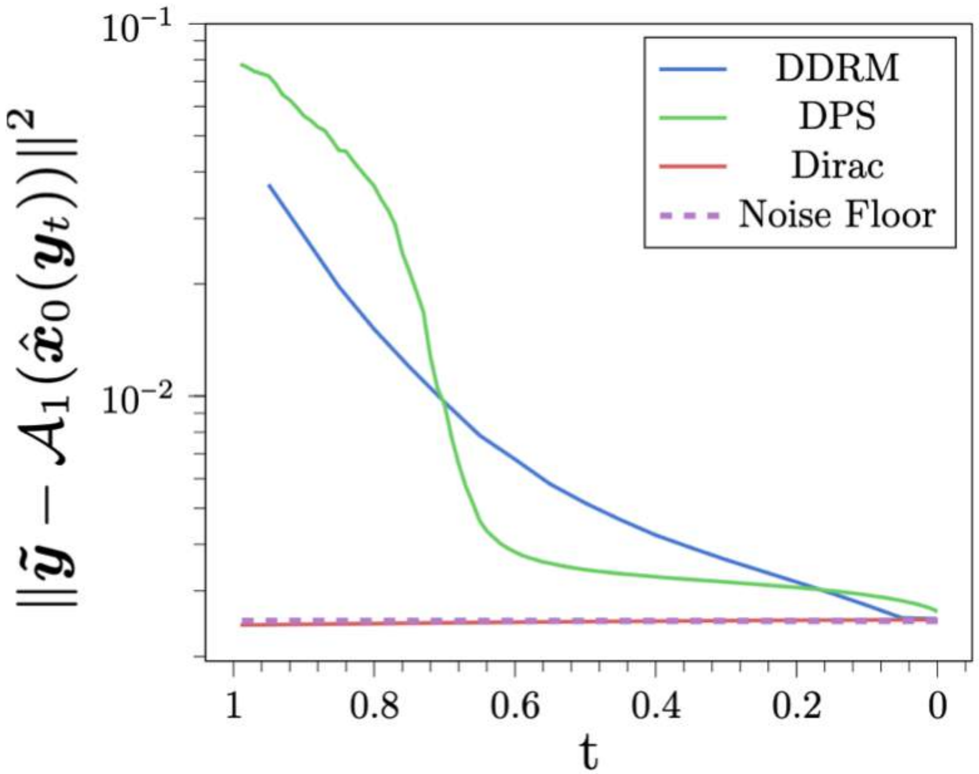
DiracDiffusion
(Denoising and Incremental Reconstruction with Assured Data-Consistency)

We learn to iteratively reverse small steps of degradation, which we call **incremental reconstruction** (IR).

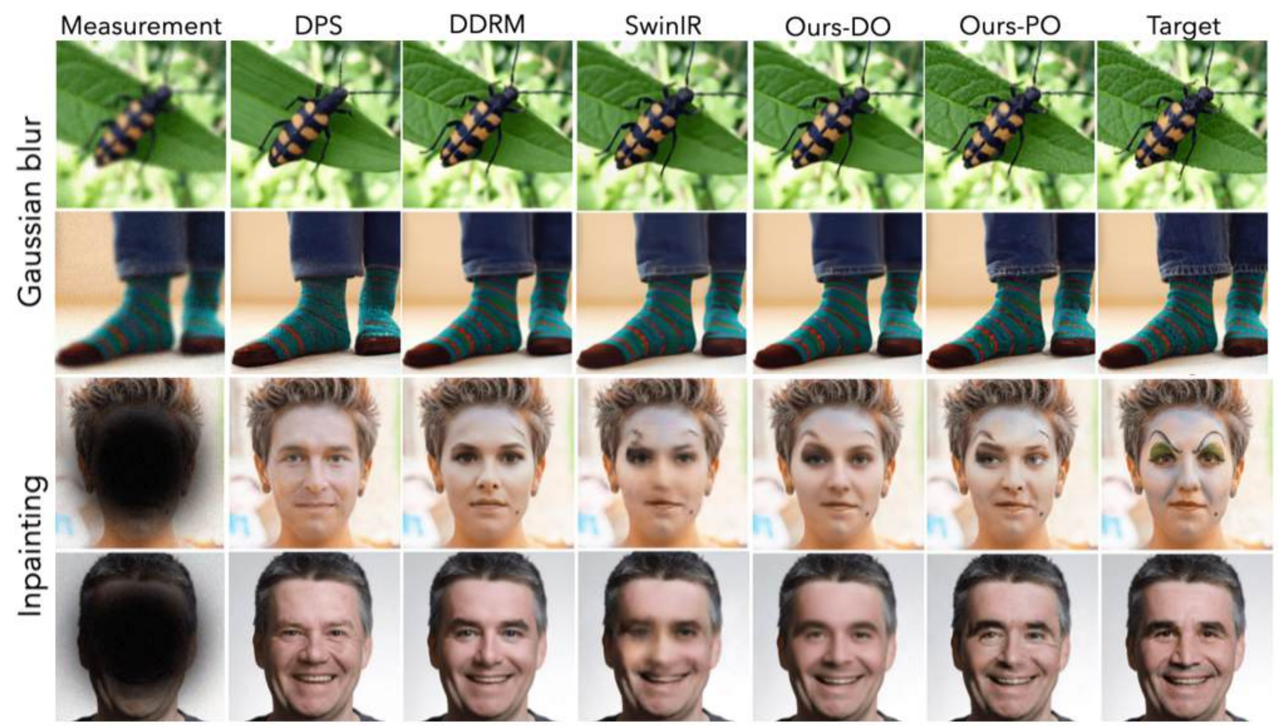


Experimental Results

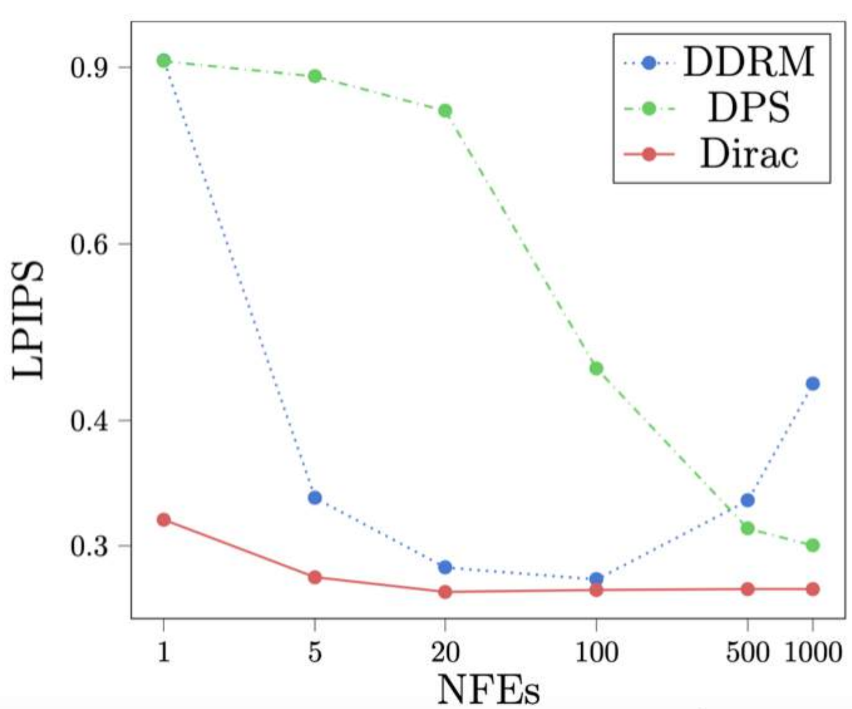
Data-consistency



Excellent reconstruction quality



Fast sampling



Method	Deblurring PSNR(↑)	SSIM(↑)	LPIPS(↓)	FID(↓)	Inpainting PSNR(↑)	SSIM(↑)	LPIPS(↓)	FID(↓)
Dirac-PO (ours)	26.67	0.7418	**0.2716**	**53.36**	25.41	0.7595	0.2611	**39.43**
Dirac-DO (ours)	28.47	0.8054	0.2972	69.15	**26.98**	**0.8435**	**0.2234**	51.87
DPS (Chung et al., 2022a)	25.56	0.6878	0.3008	65.68	21.06	0.7238	0.2899	57.92
DDRM (Kawar et al., 2022a)	27.21	0.7671	0.2849	65.84	25.62	0.8132	0.2313	54.37
SwinIR (Liang et al., 2021)	**28.53**	**0.8070**	0.3048	72.93	24.46	0.8134	0.2660	59.94
PnP-ADMM (Chan et al., 2016)	27.02	0.7596	0.3973	74.17	12.27	0.6205	0.4471	192.36
ADMM-TV	26.03	0.7323	0.4126	89.93	11.73	0.5618	0.5042	264.62

Method	Deblurring PSNR(↑)	SSIM(↑)	LPIPS(↓)	FID(↓)	Inpainting PSNR(↑)	SSIM(↑)	LPIPS(↓)	FID(↓)
Dirac-PO (ours)	24.68	0.6582	**0.3302**	53.91	26.36	0.8087	0.2079	34.33
Dirac-DO (ours)	**25.76**	**0.7085**	0.3705	83.23	**28.92**	**0.8958**	**0.1676**	38.25
DPS (Chung et al., 2022a)	21.51	0.5163	0.4235	**52.60**	22.71	0.8026	0.1986	34.55
DDRM (Kawar et al., 2022a)	24.53	0.6676	0.3917	61.06	25.92	0.8347	0.2138	**33.71**
SwinIR (Liang et al., 2021)	25.07	0.6801	0.4159	84.80	26.87	0.8490	0.2161	45.69
PnP-ADMM (Chan et al., 2016)	25.02	0.6722	0.4565	98.72	18.14	0.7901	0.2709	101.25
ADMM-TV	24.31	0.6441	0.4578	88.26	17.60	0.7229	0.3157	120.22

Table 1. Experimental results on the FFHQ (top) and ImageNet (bottom) test splits.

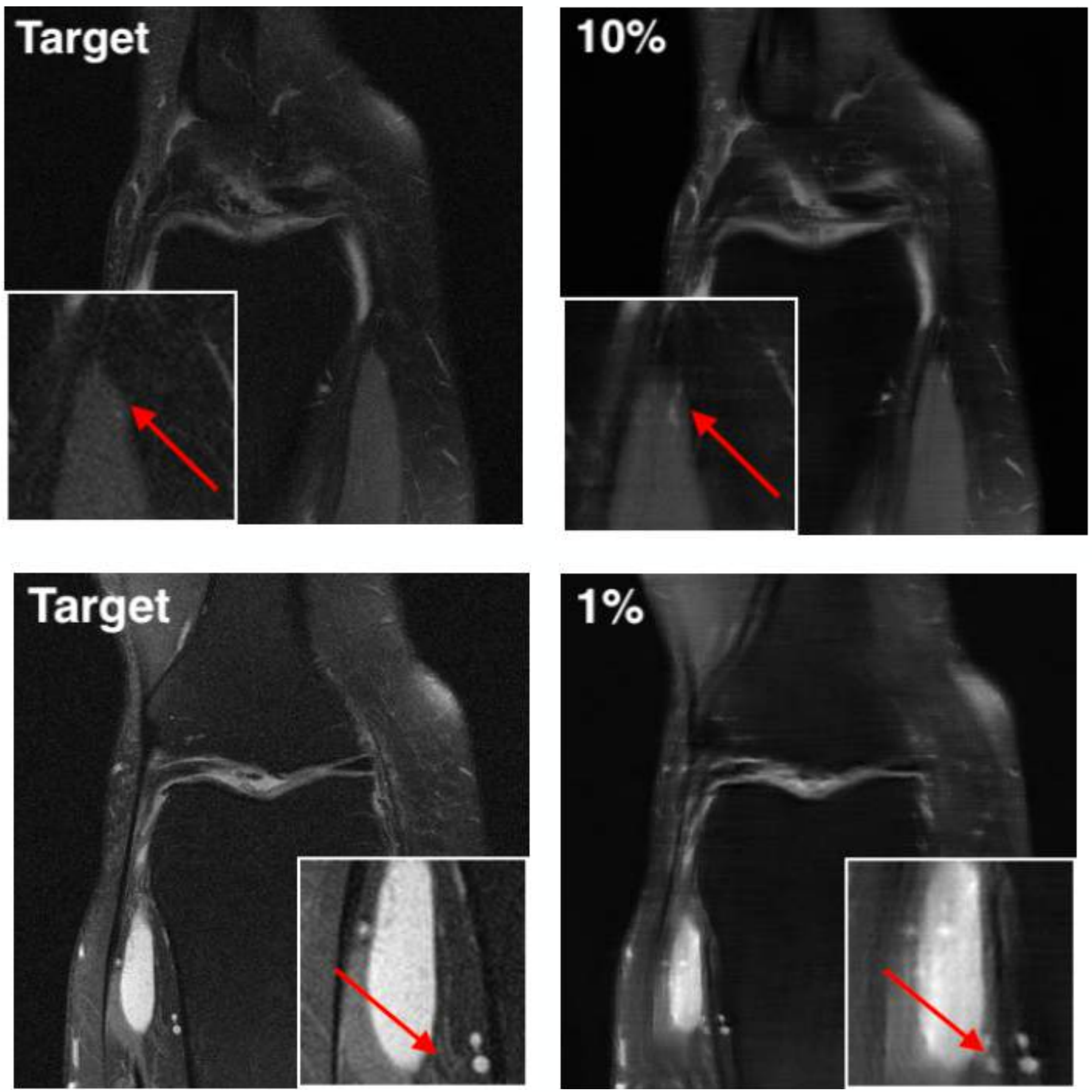
Untrained methods


TIME IS
RUNNING
OUT
0:53 47
0:27 08

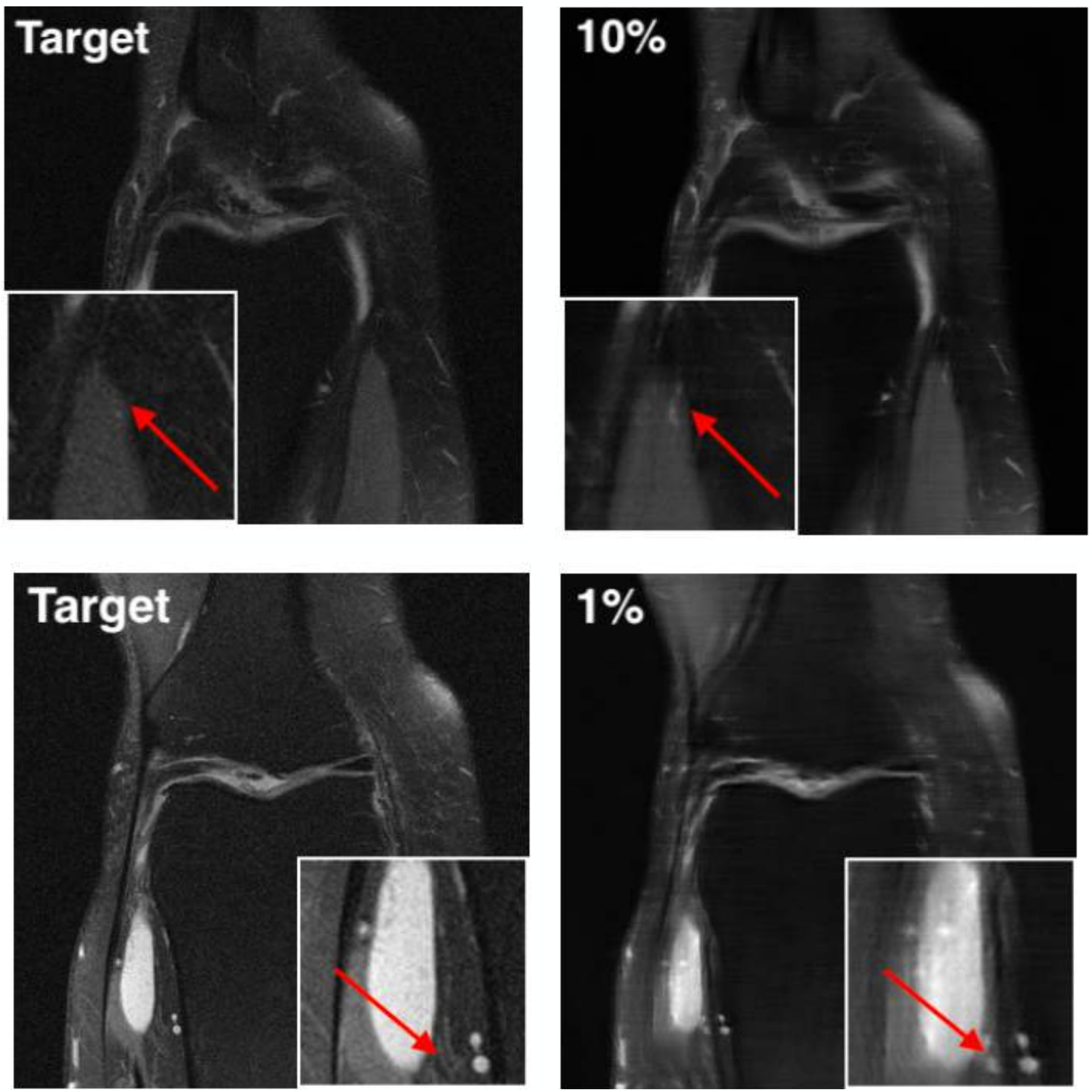
Future Directions

Reliability Challenge

Hallucinations

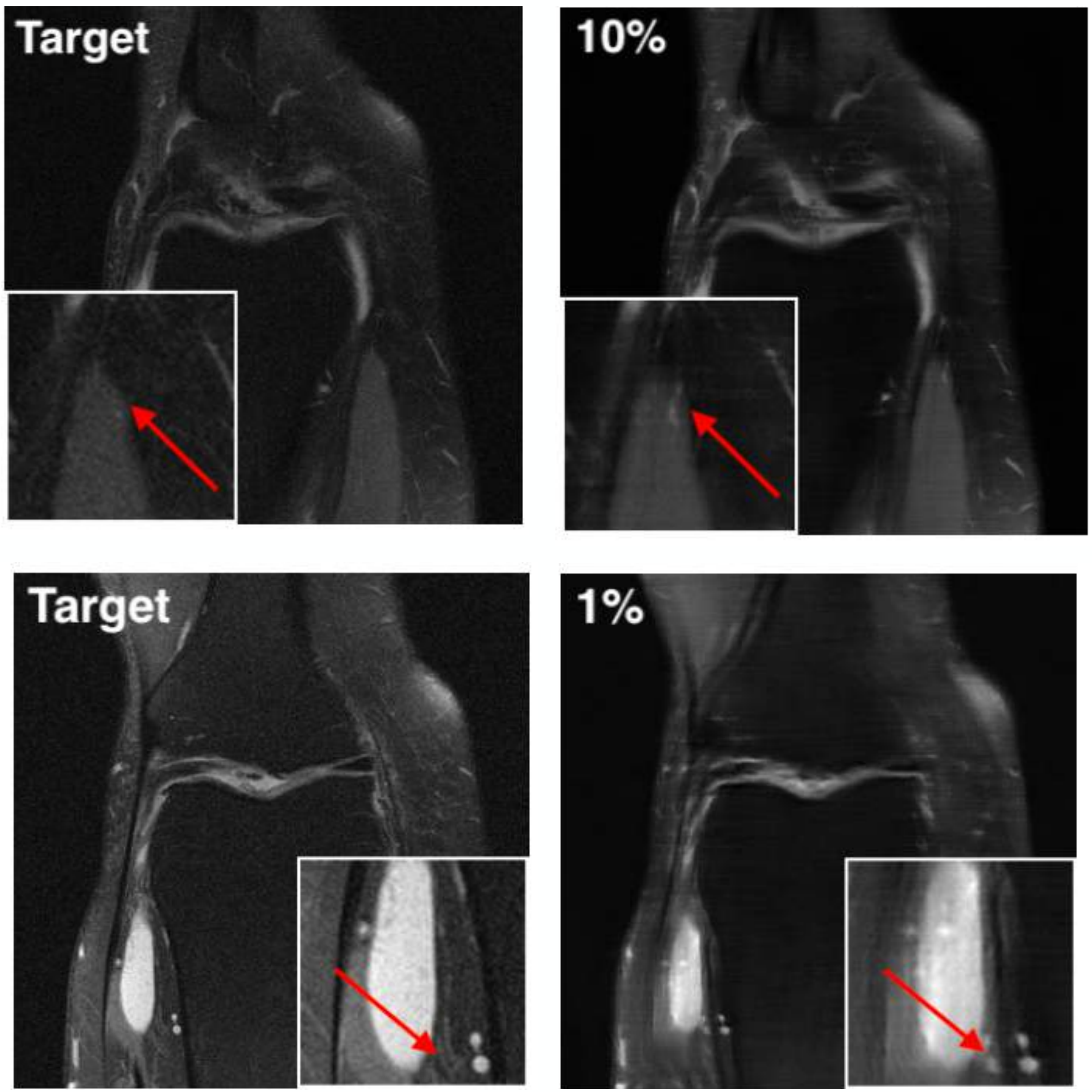


Hallucinations



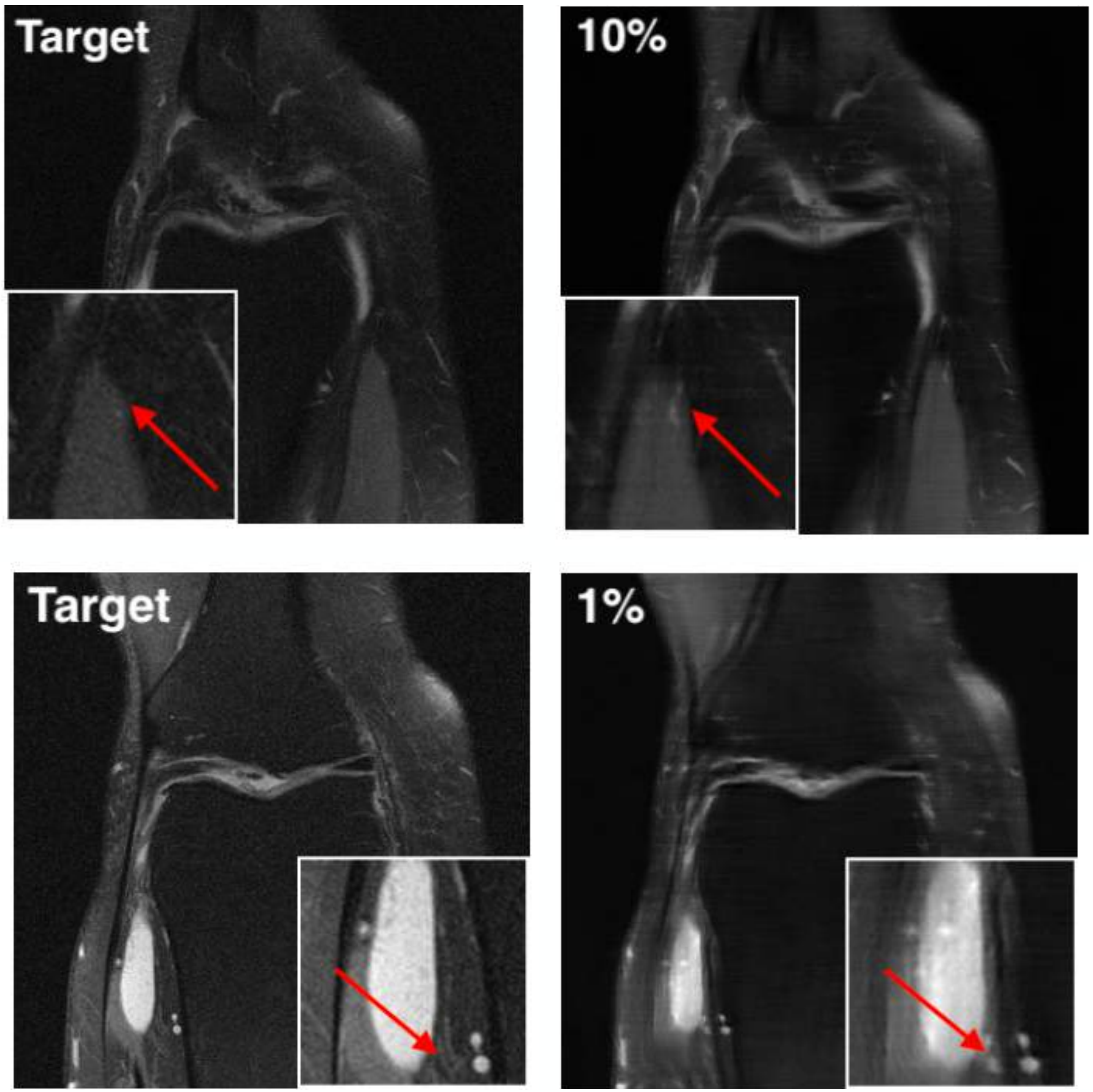
- DL model “sees” features on reconstructions that shouldn’t be present

Hallucinations



- DL model "sees" features on reconstructions that shouldn't be present
- Hallucinated features appear real as opposed to reconstruction artifacts

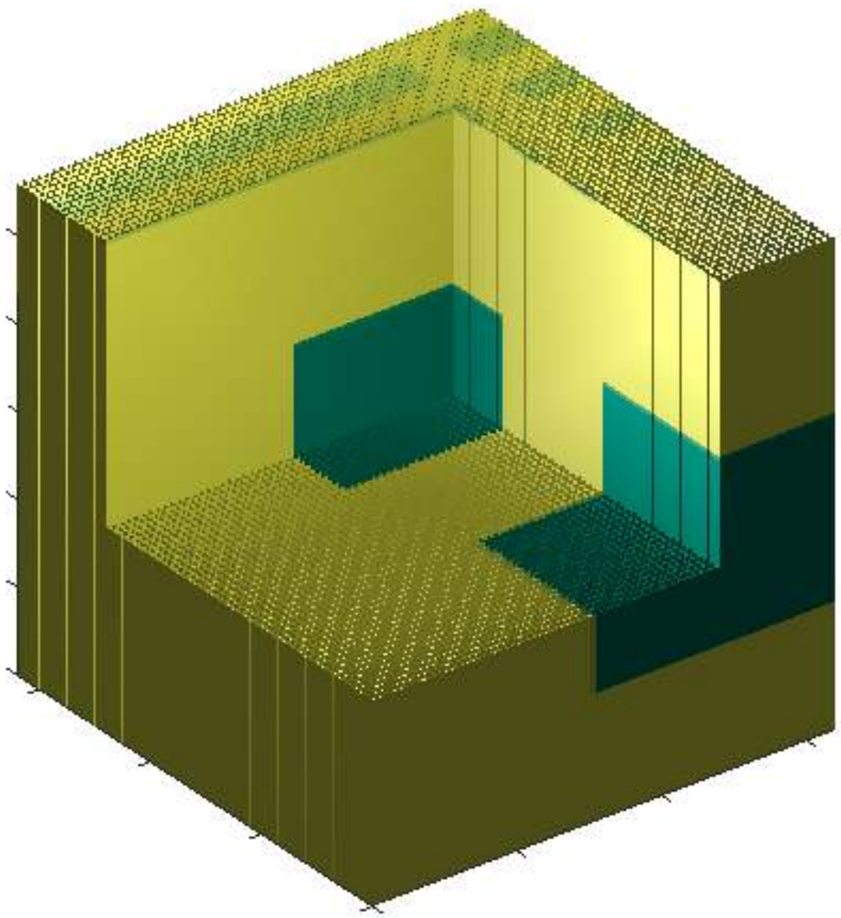
Hallucinations



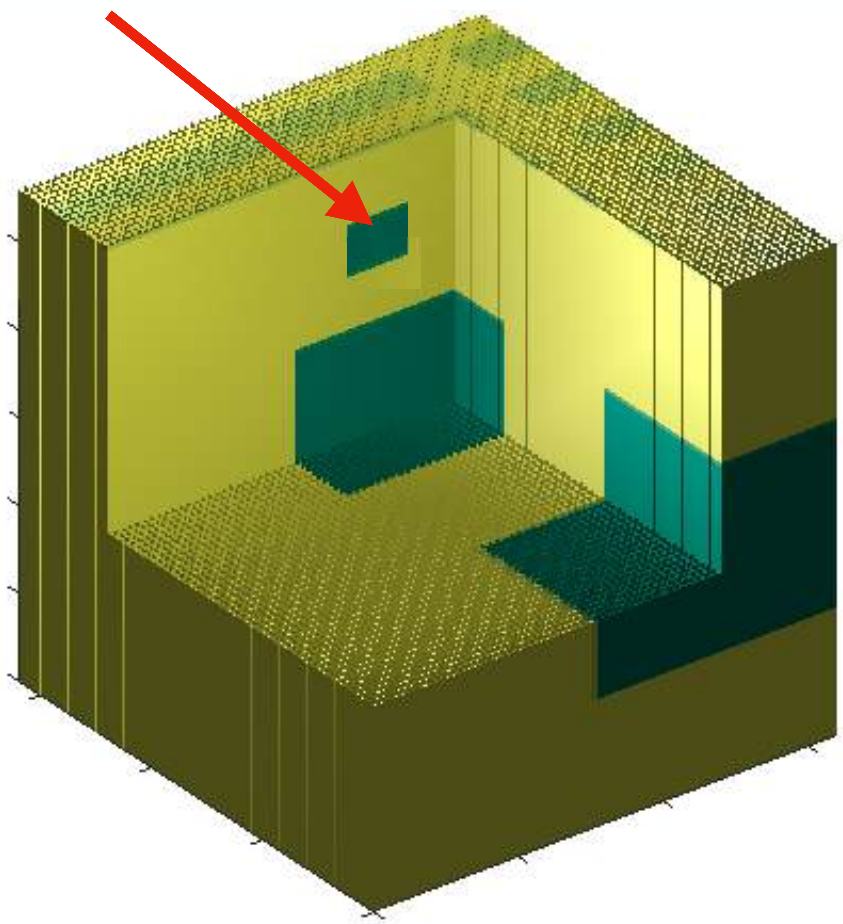
- DL model "sees" features on reconstructions that shouldn't be present
- Hallucinated features appear real as opposed to reconstruction artifacts
- Can lead to critical mistakes (misdiagnosis)

Critical reconstruction errors

- 3D nano-scale imaging example



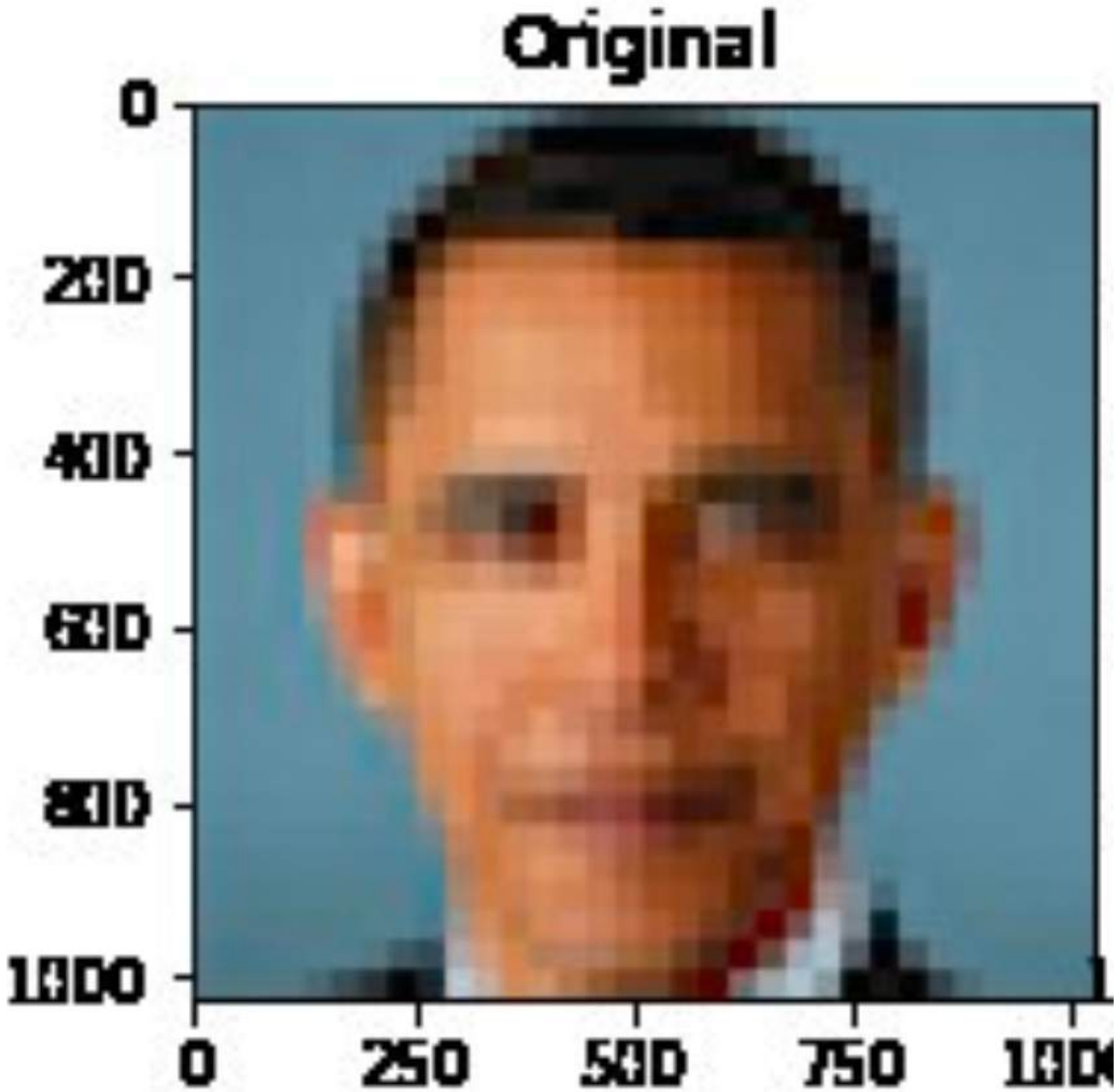
Target nano-structure



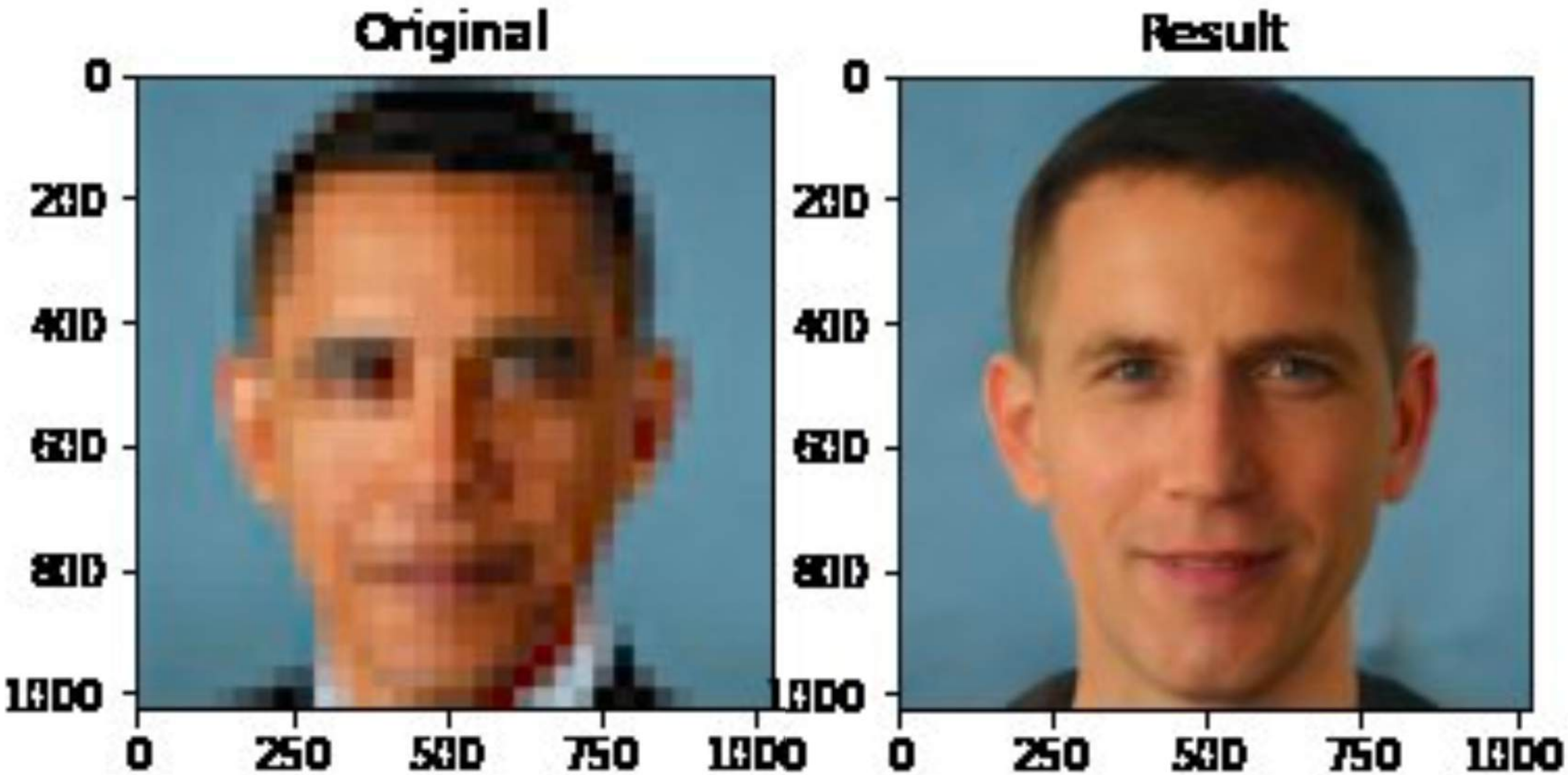
DL reconstruction

Bias

Bias

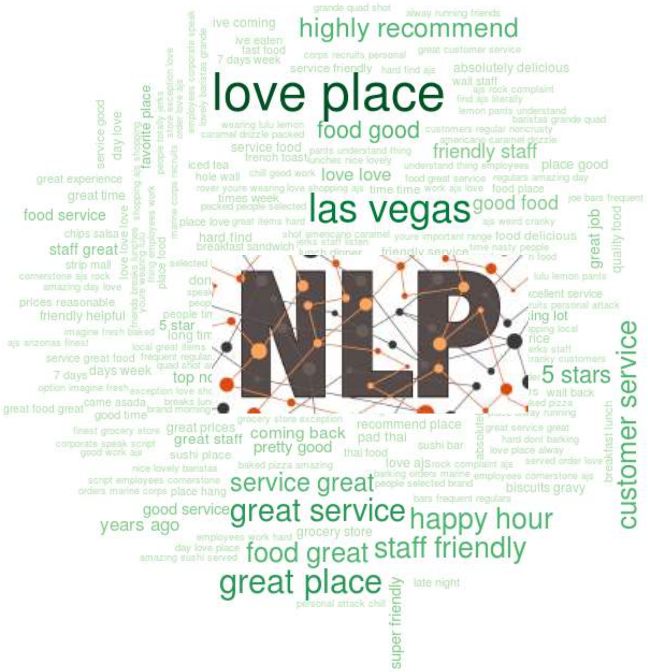


Bias

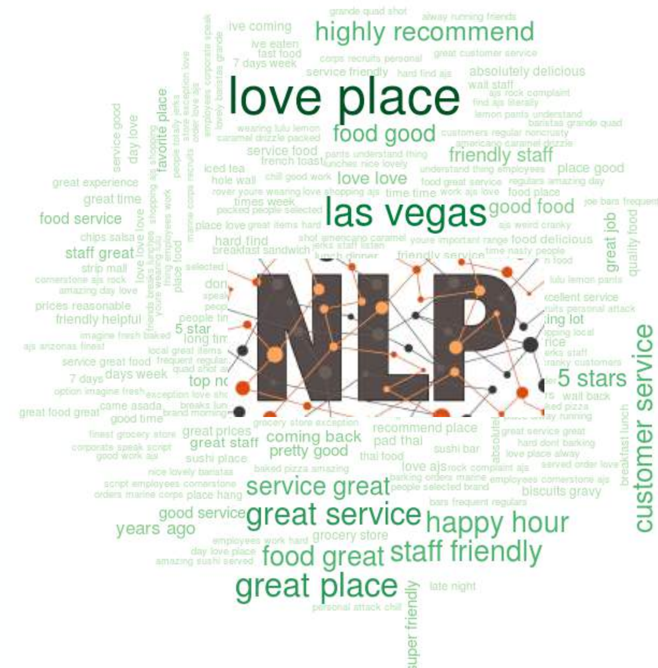


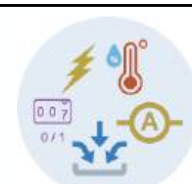
Data Scarcity Challenge

Deep learning is data-hungry

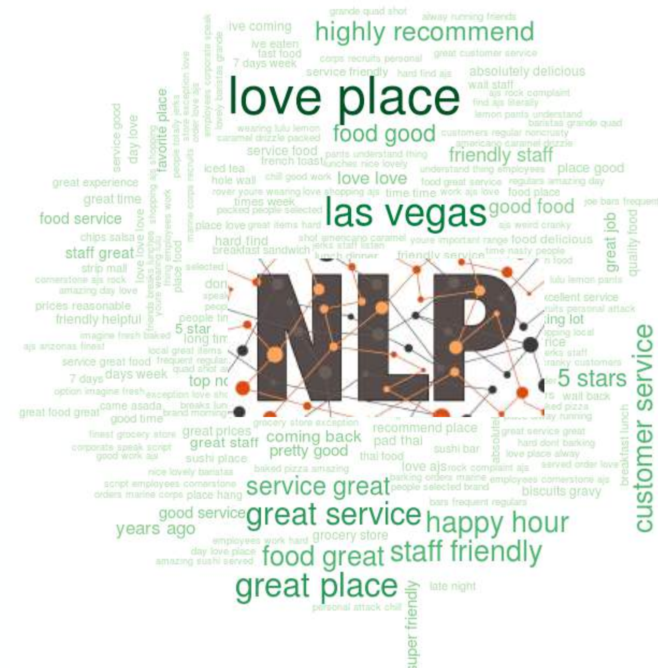


Deep learning is data-hungry



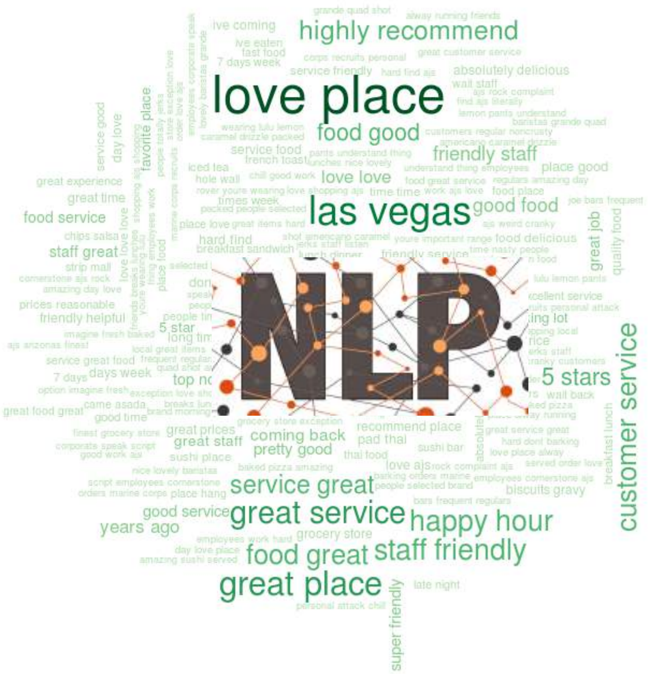
	'Internet data'
Data acquisition 	readily available

Deep learning is data-hungry



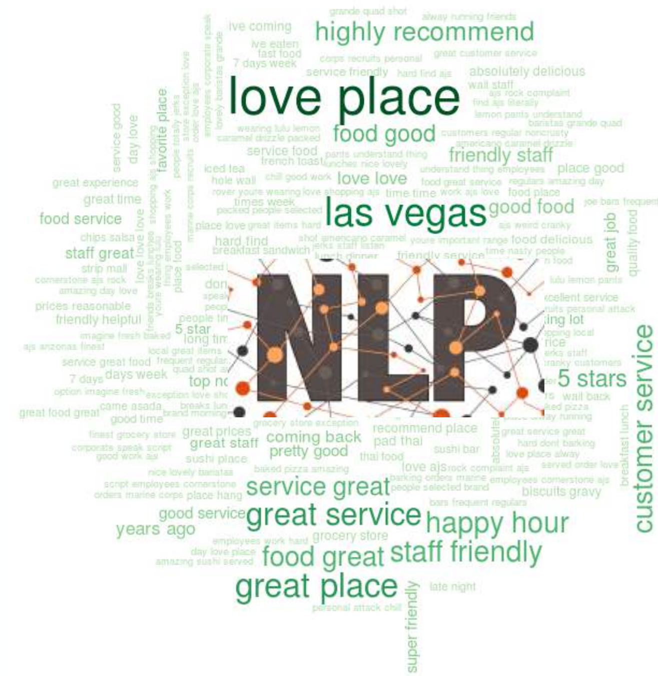
	'Internet data'
Data acquisition	readily available
Cost	very low

Deep learning is data-hungry



	'Internet data'
Data acquisition	readily available
Cost	very low
Amount of data	huge

Deep learning is data-hungry



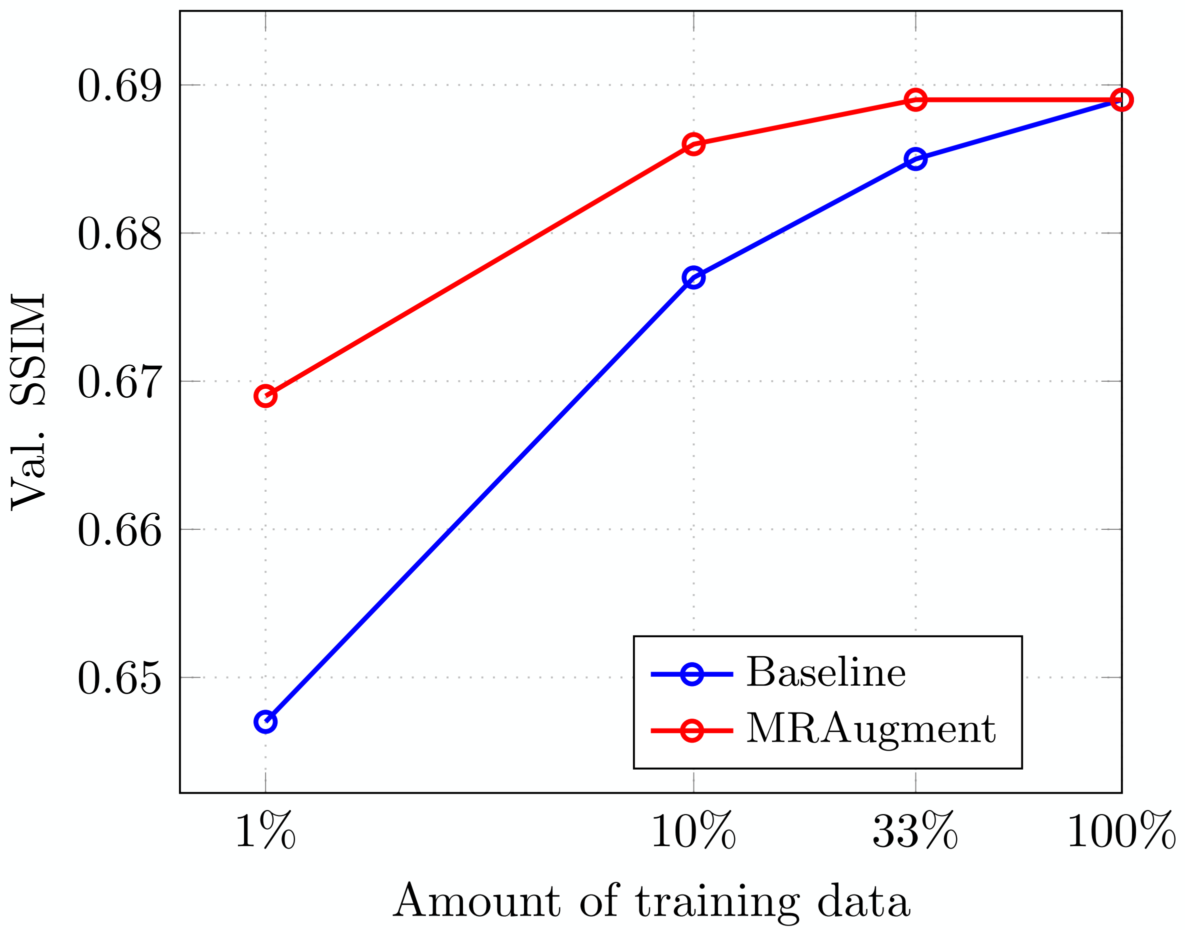
	'Internet data'
Data acquisition	readily available
Cost	very low
Amount of data	huge

How about scientific data?

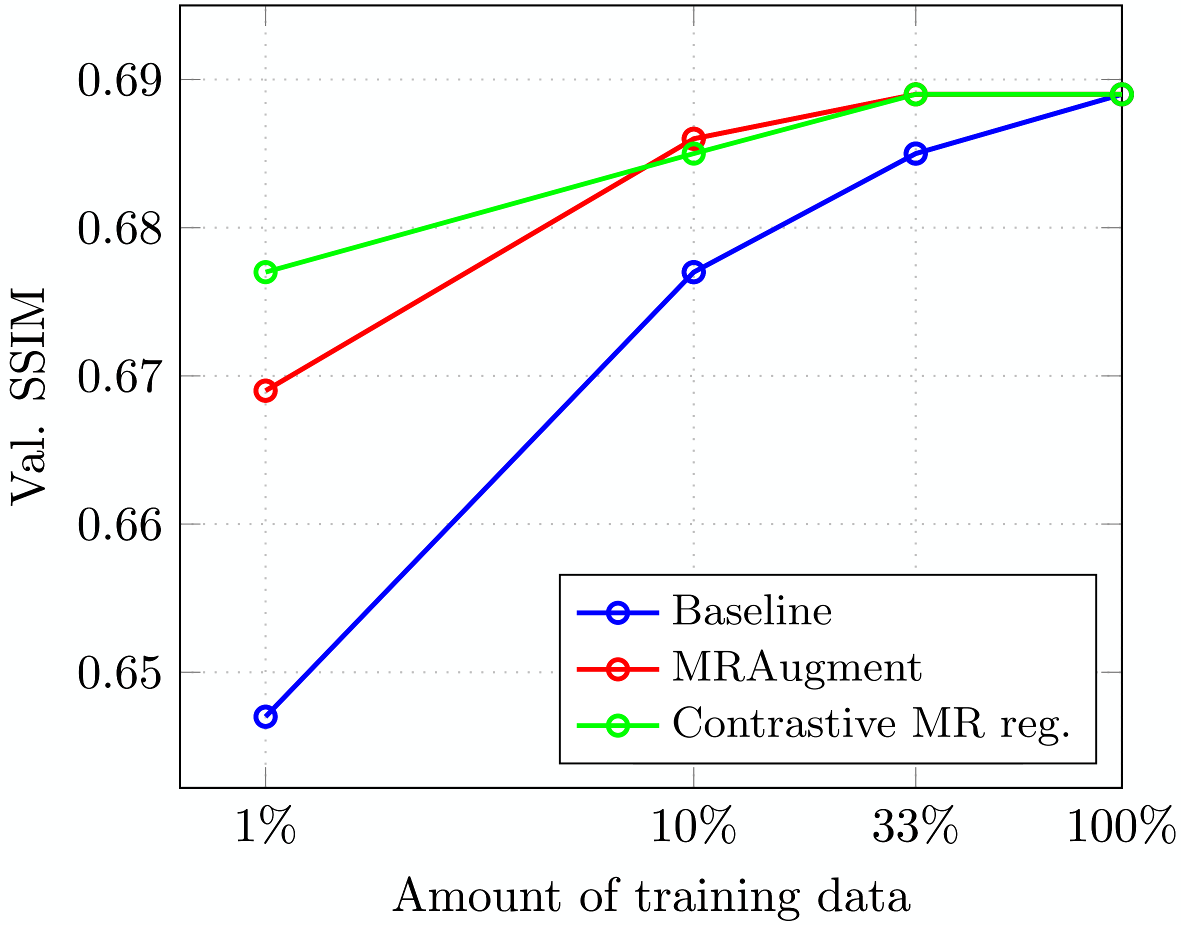
Create a Phase Retrieval DataSet

Reduce reliance on training data

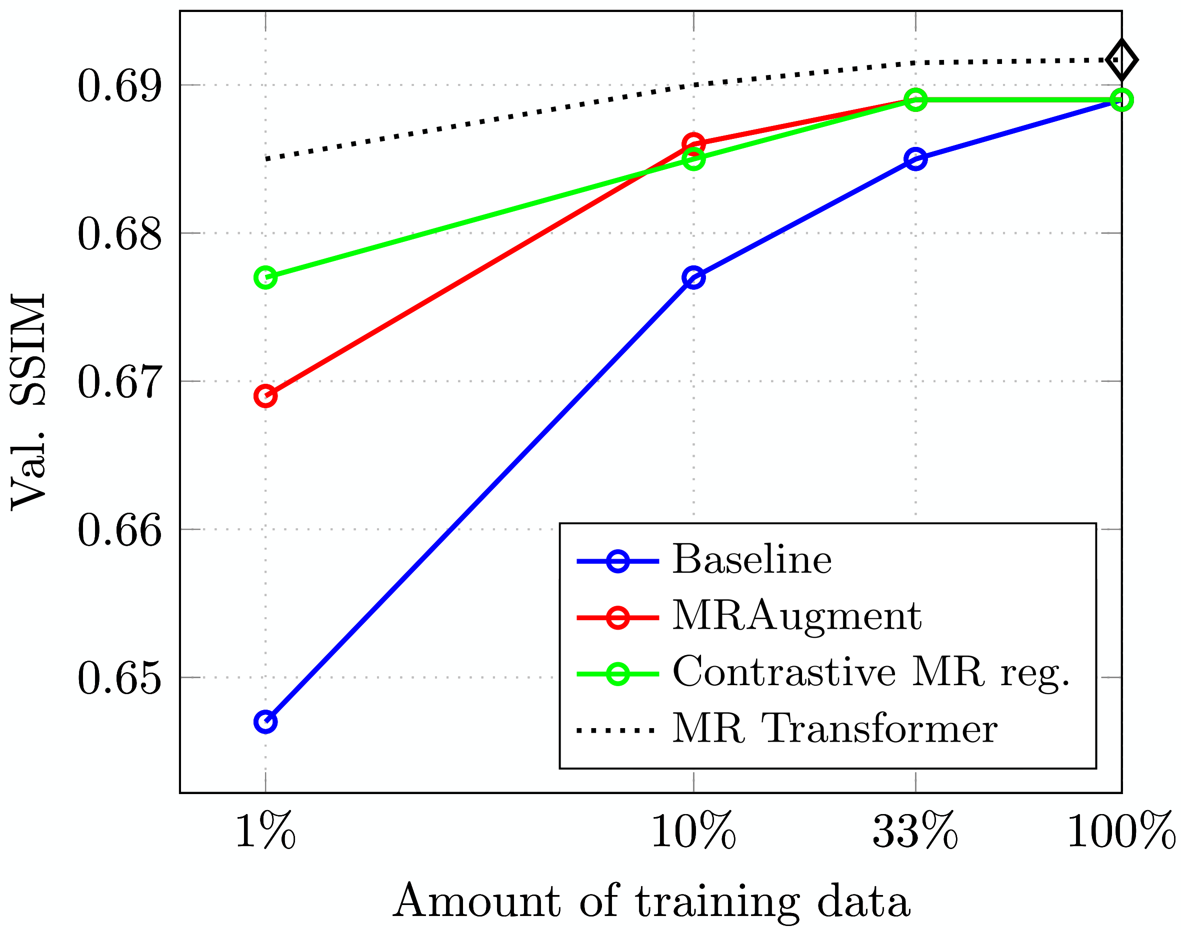
Reduce reliance on training data



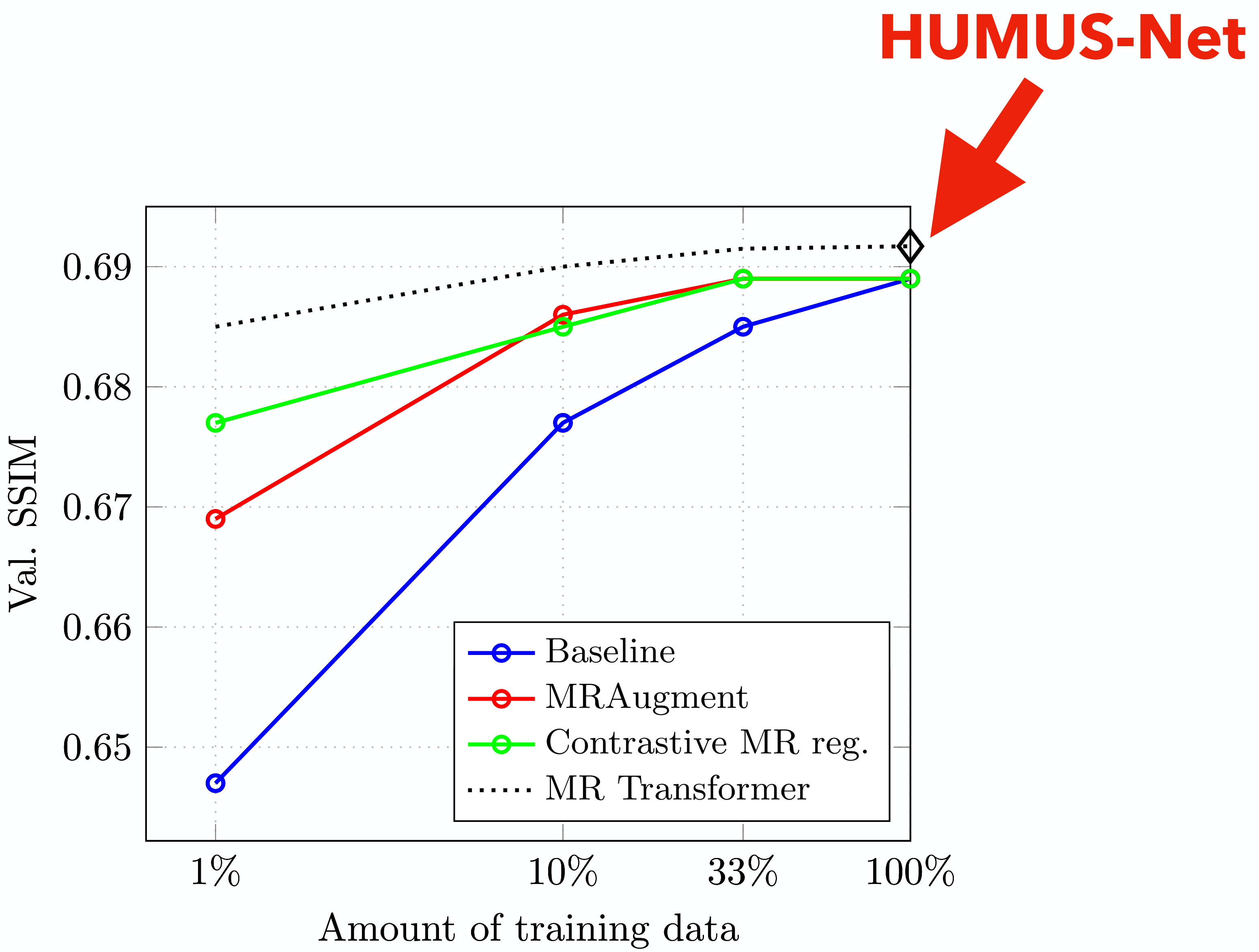
Reduce reliance on training data



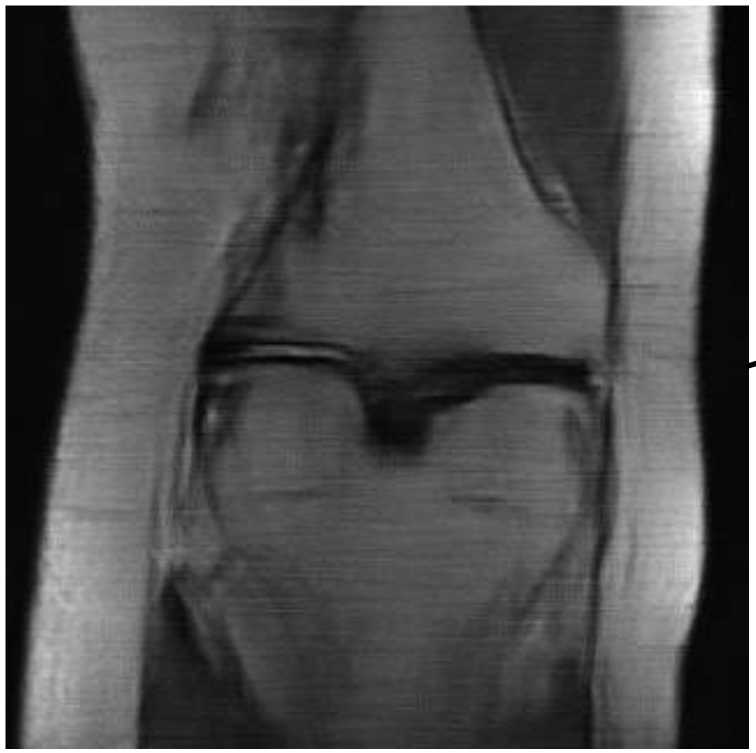
Reduce reliance on training data

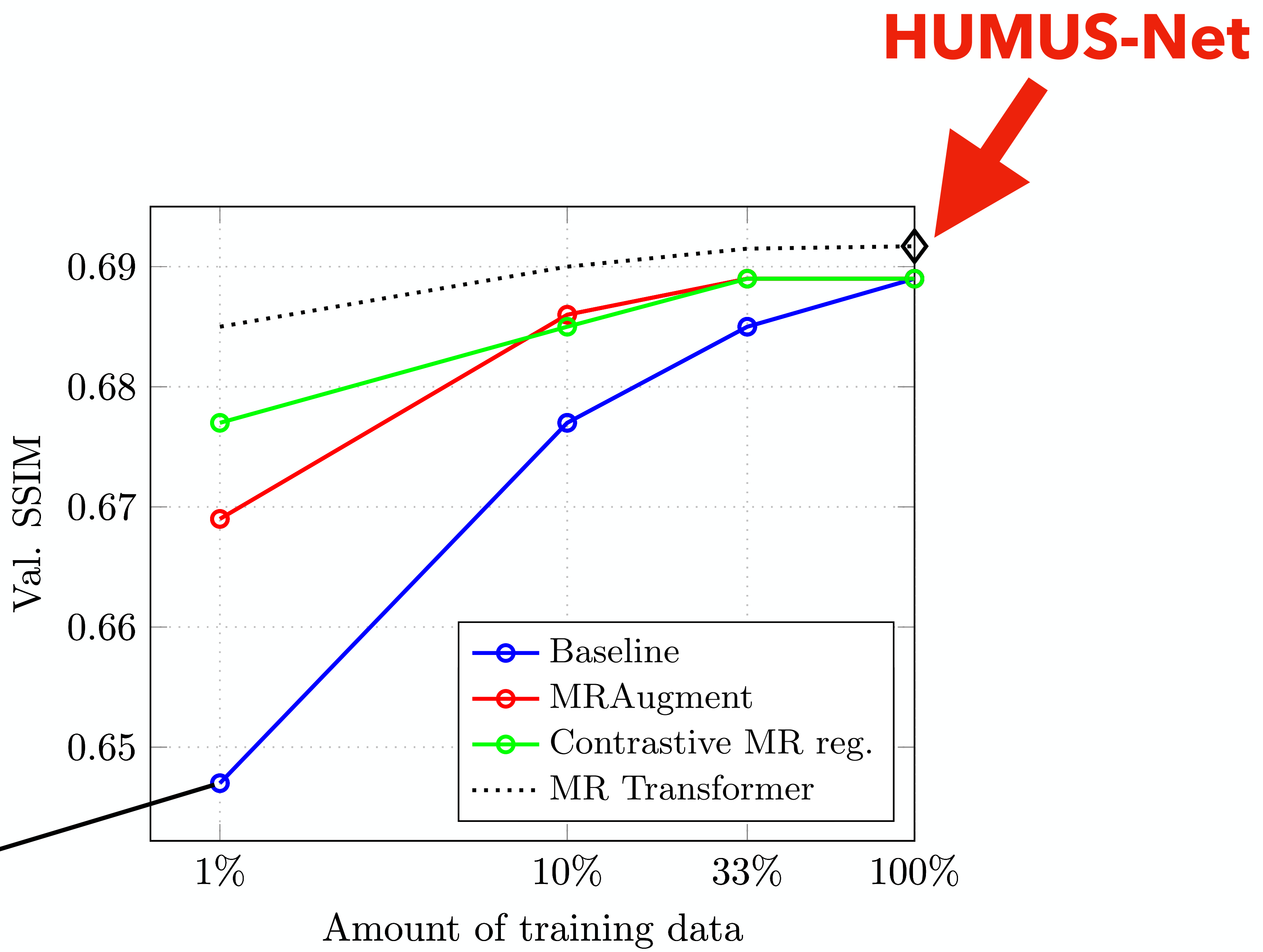


Reduce reliance on training data

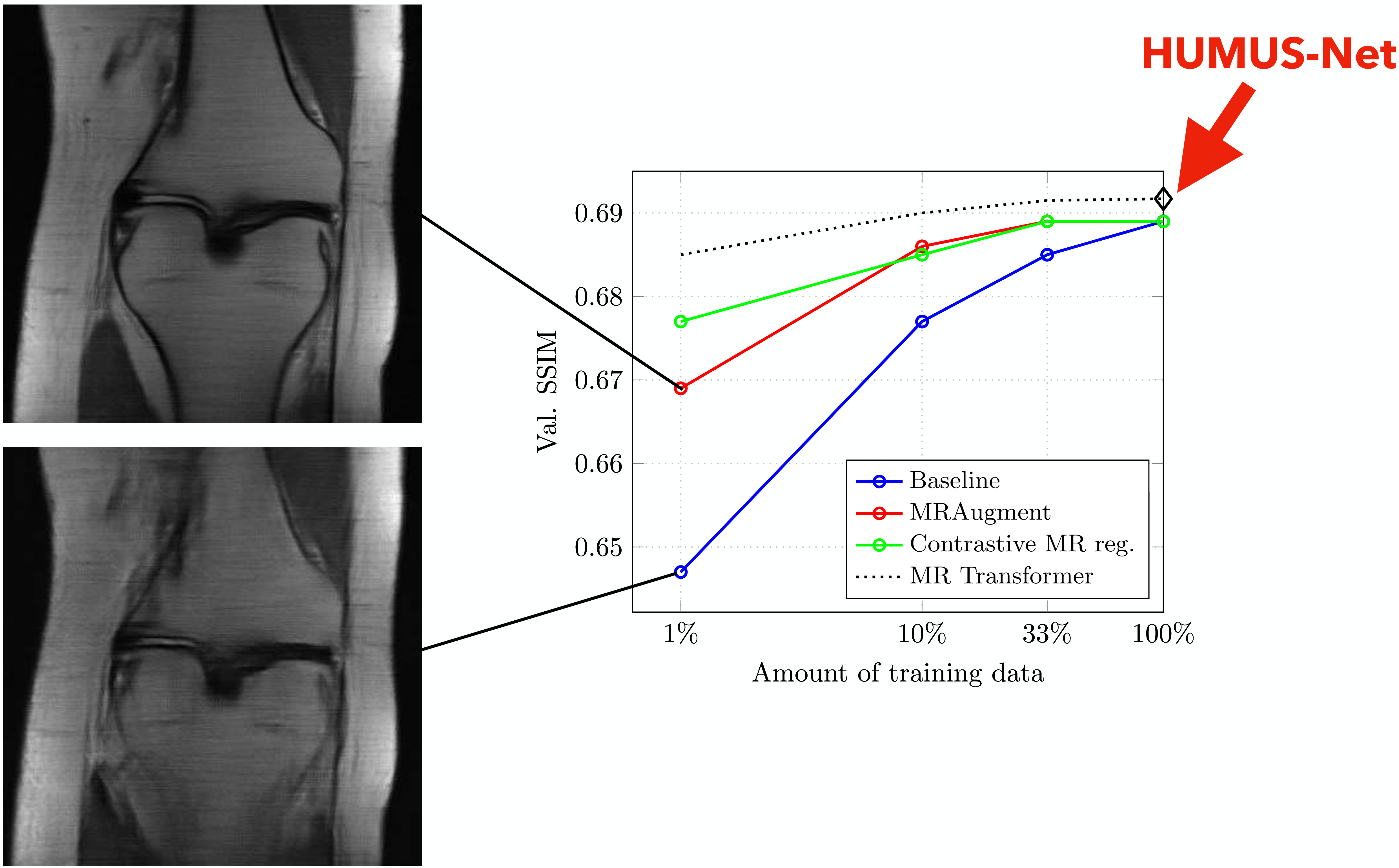


Reduce reliance on training data





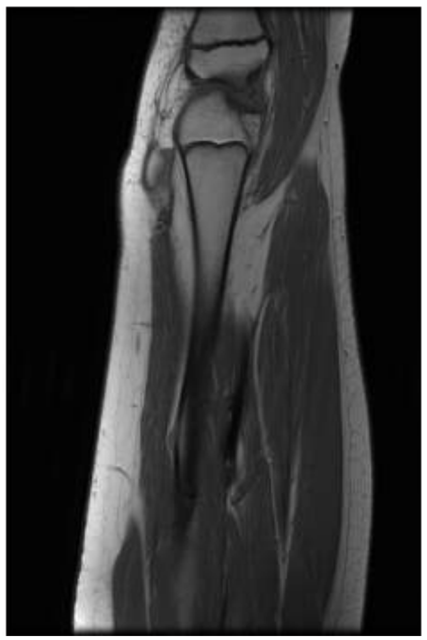
Reduce reliance on training data



Computational/complexity Challenge

Complexity of scientific data

High spatial resolution

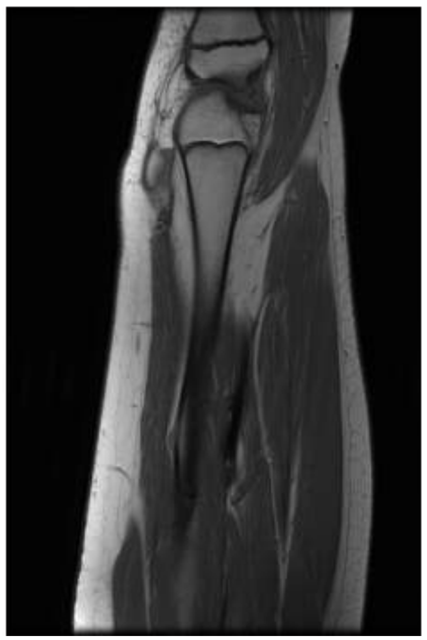


640×368

32×32

Complexity of scientific data

High spatial resolution



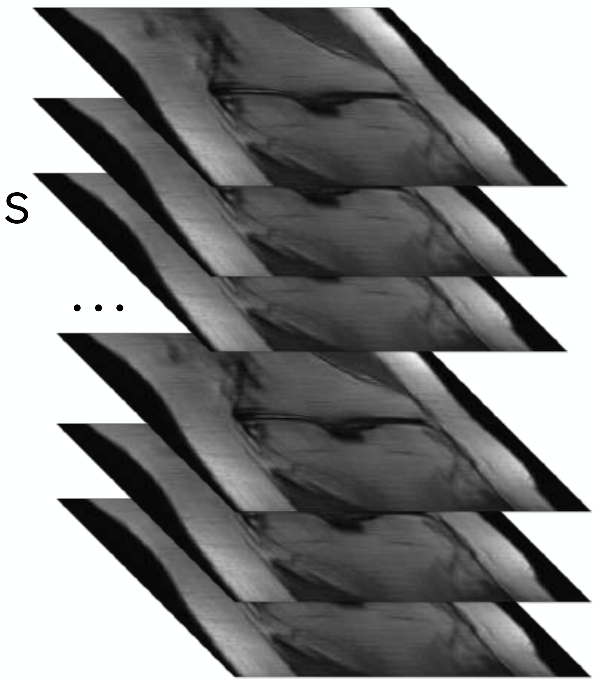
640×368

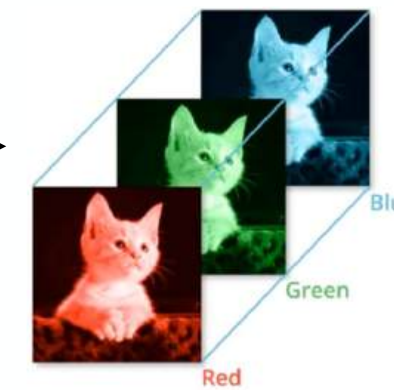
32×32

Large number of image channels

15+ coils

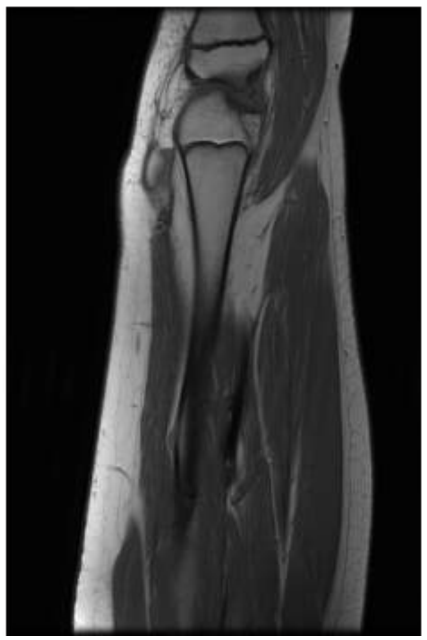
…





Complexity of scientific data

High spatial resolution

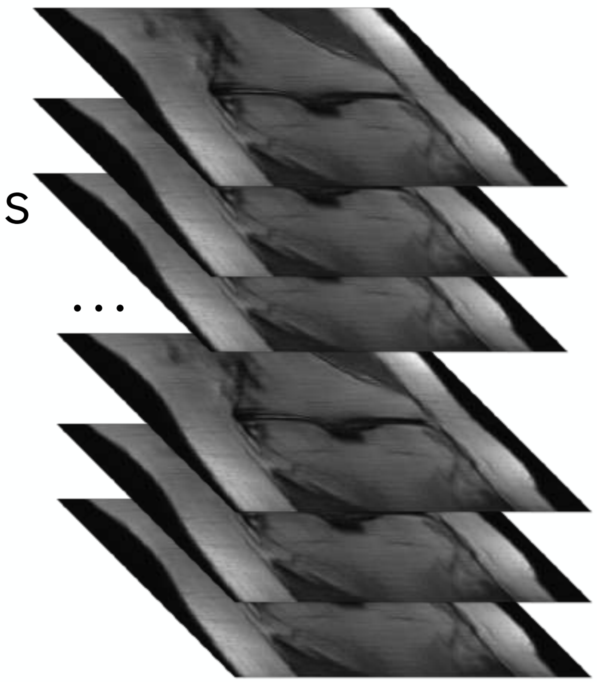


640×368

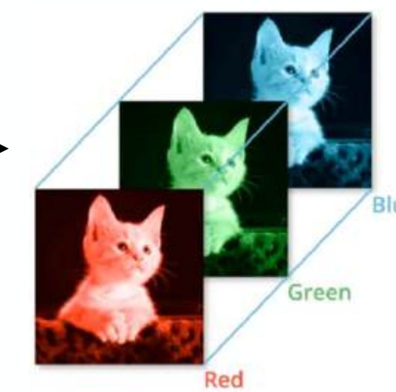
32×32

Large number of image channels

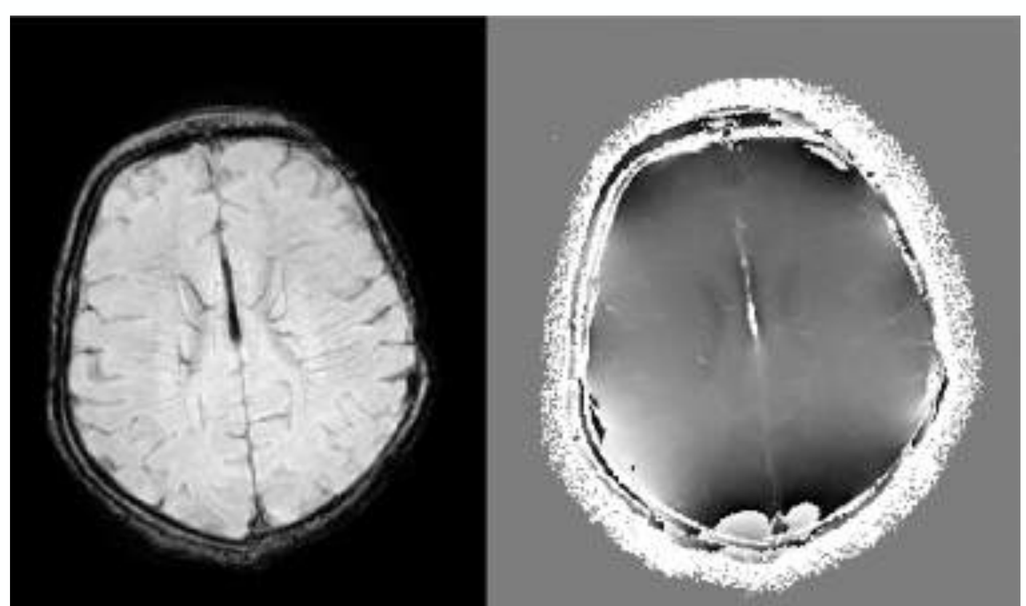
15+ coils



...



Complex valued images



magnitude *phase*

Thanks!

Funding acknowledgement

the David & Lucile Packard FOUNDATION

ALFRED P. SLOAN FOUNDATION 1934

NSF

DARPA

FastNICs

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
UNITED STATES AIR FORCE

DARPA

LwLL

NSF

Google

amazon

NORTHROP GRUMMAN

References

Deep Learning Techniques for Inverse Problems in Imaging

Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, Rebecca Willett

https://arxiv.org/abs/2005.06001

Great Talk by Alex Dimakis on Generative models

https://simons.berkeley.edu/sites/default/files/docs/18458/simons-dimakis.pdf

Great Tutorial By Arash Vahdat and Collaborators on Diffusion Models

https://www.youtube.com/watch?v=cS6JQpEY9cs

https://www.youtube.com/watch?v=1d4r19GEVos

- Additional References: https://docs.google.com/spreadsheets/d/1e36mDqj8SY9ODT3Kz8frqSchK2QDWZdHuzVy-EKjhFo/edit?usp=sharing