Asilomar Conference on Signals, Systems, and Computers

Quantized Compressed Sensing with Score-based Generative Models

Xiangming Meng[1], Yoshiyuki, Kabashima[2]

[1] Zhejiang University, China
[2] The University of Tokyo, Japan

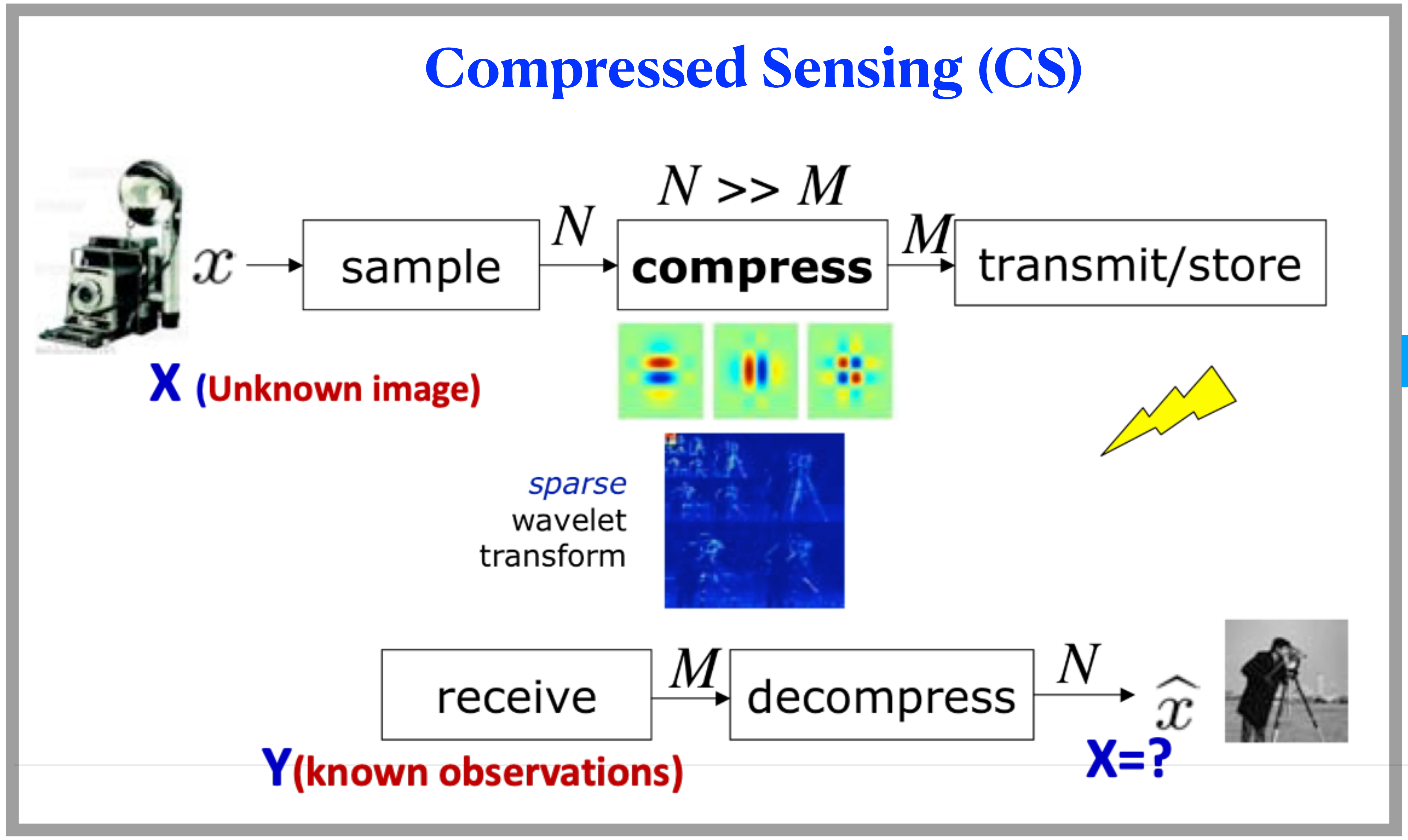
October 31, 2023

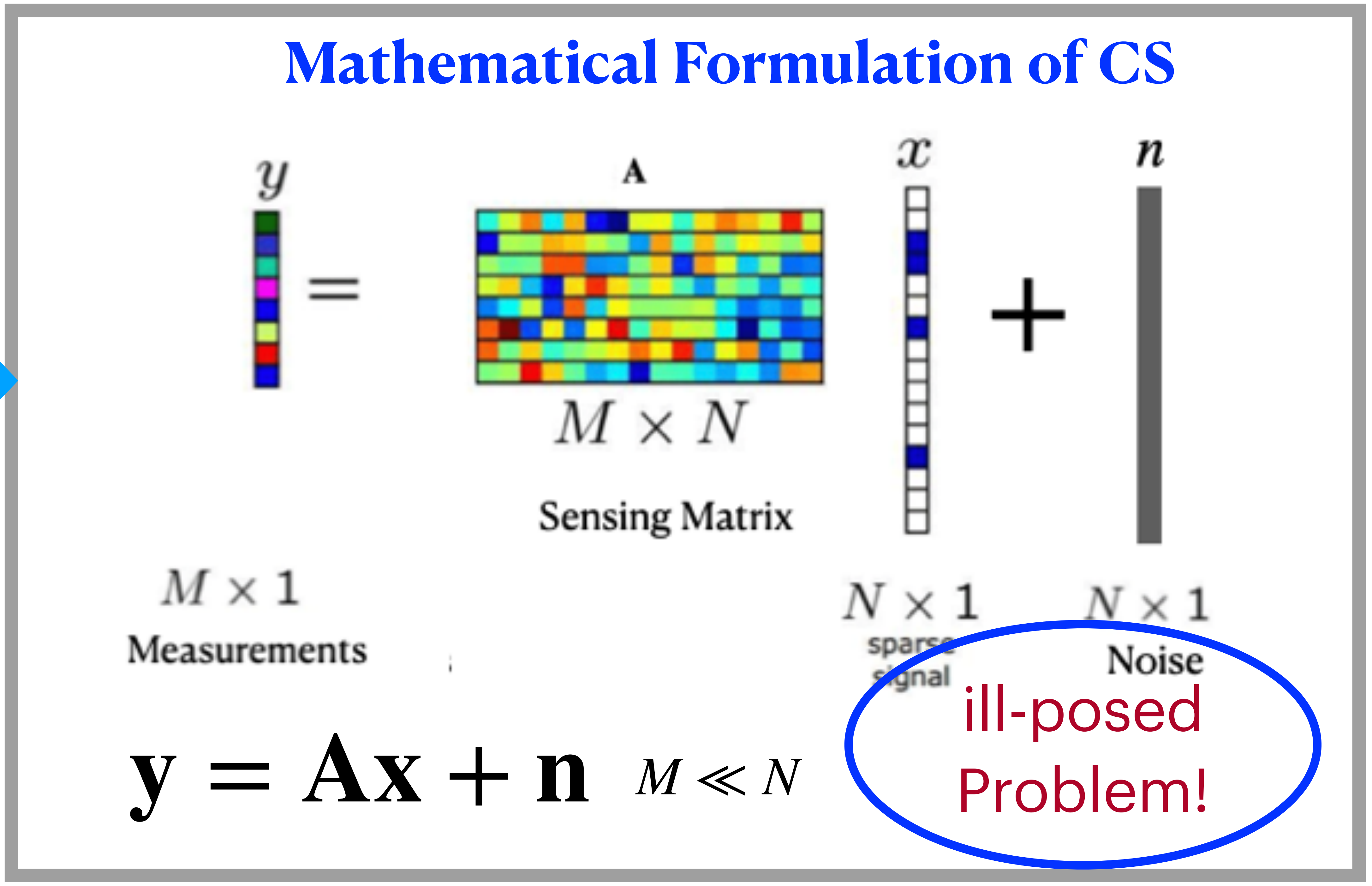
Yoshiyuki Kabashima
The University of Tokyo

Contents

Background

■ **Compressed Sensing**





•**Goal**: Recover a ***sparse*** or ***compressible*** signal from $M \ll N$ measurements

•**Solution**: Exploit the sparsity as regularization, e.g, L1 regularization

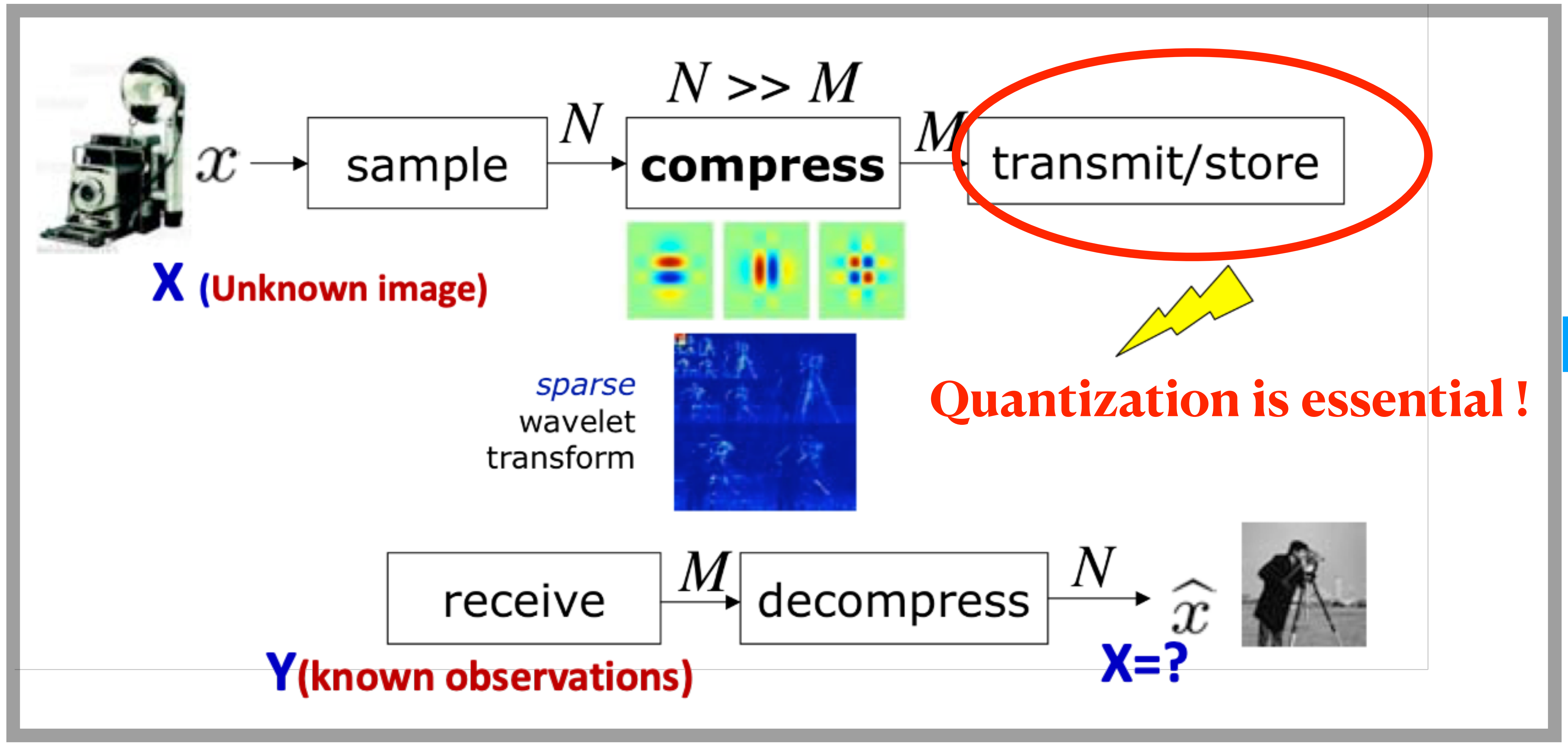
•**Theoretical guarantee** *[Candes-Romberg-Tao2006]*

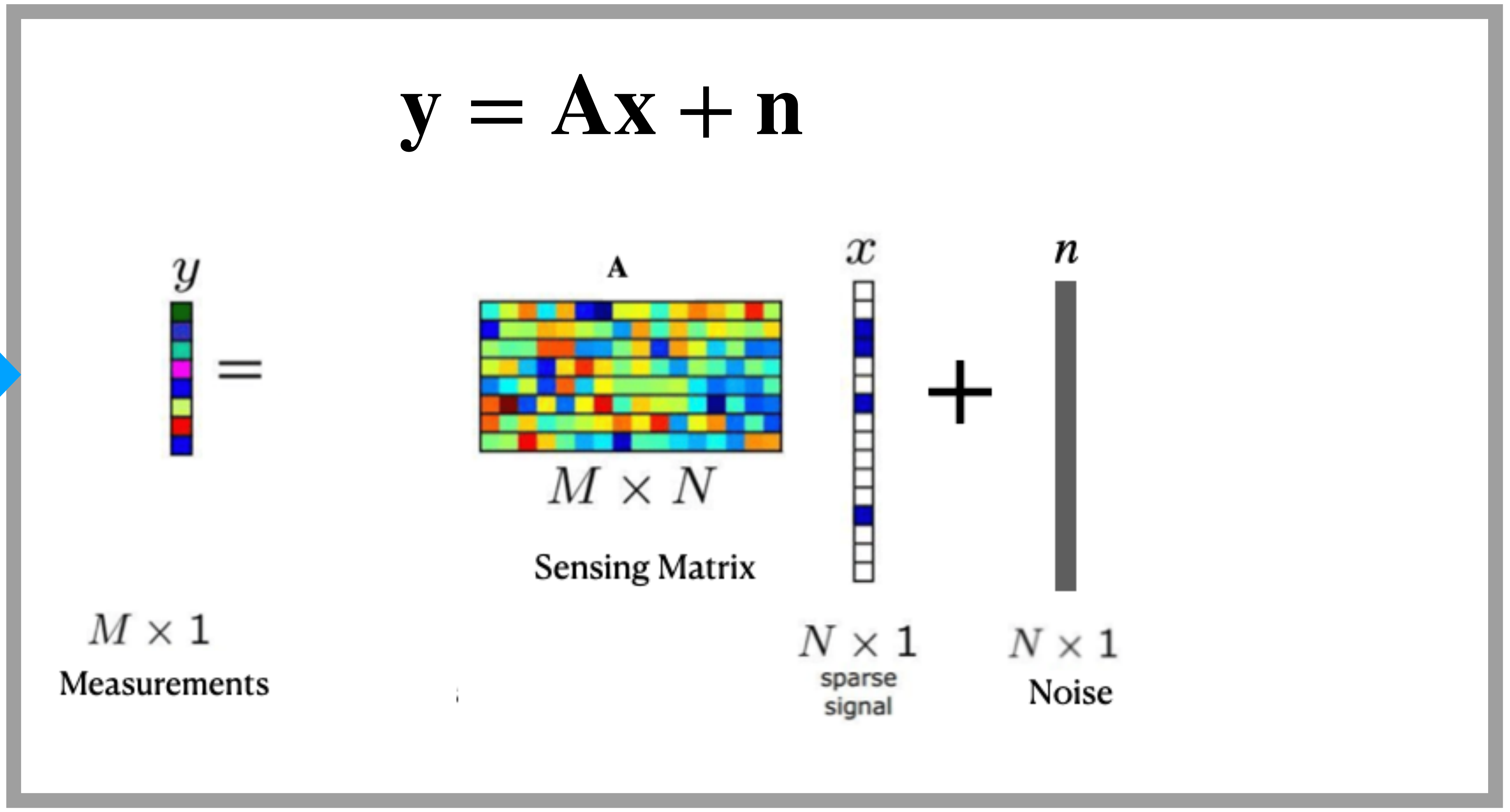
Candes Romberg Tao

Credit https://www.raeng.org.uk/publications/other/candes-presentation-frontiers-of-engineering

Background

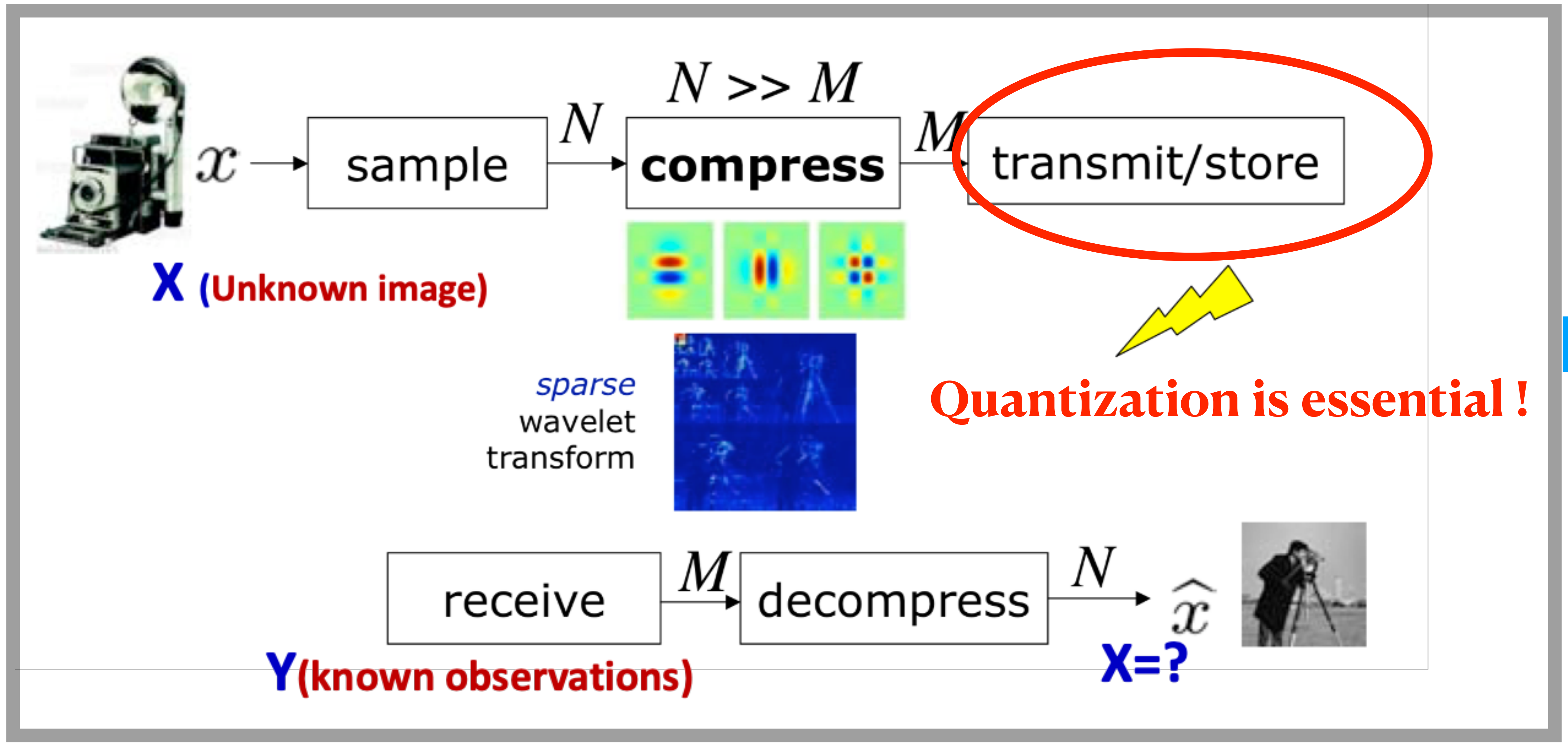
■ Quantized Compressed Sensing

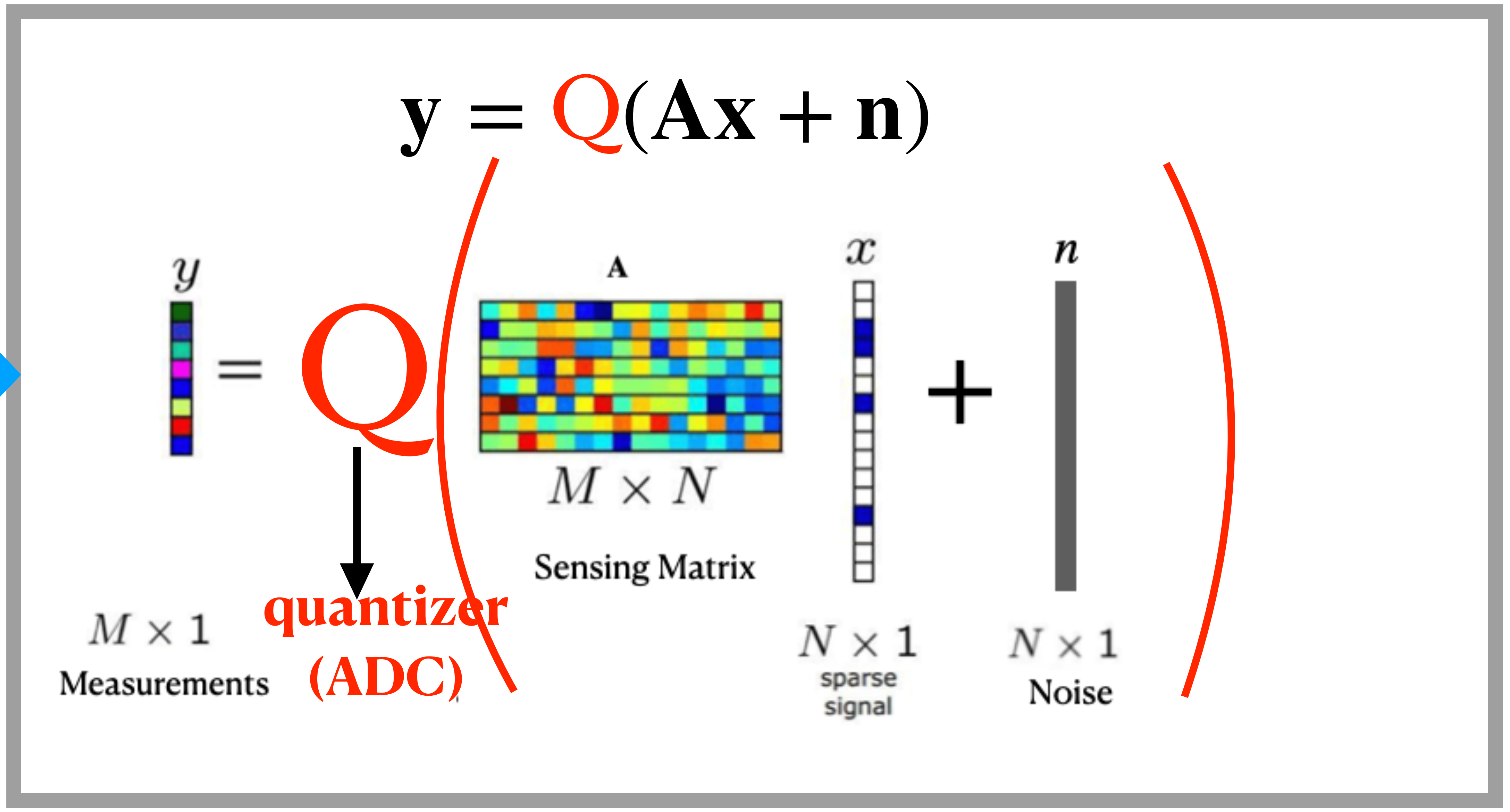




Background

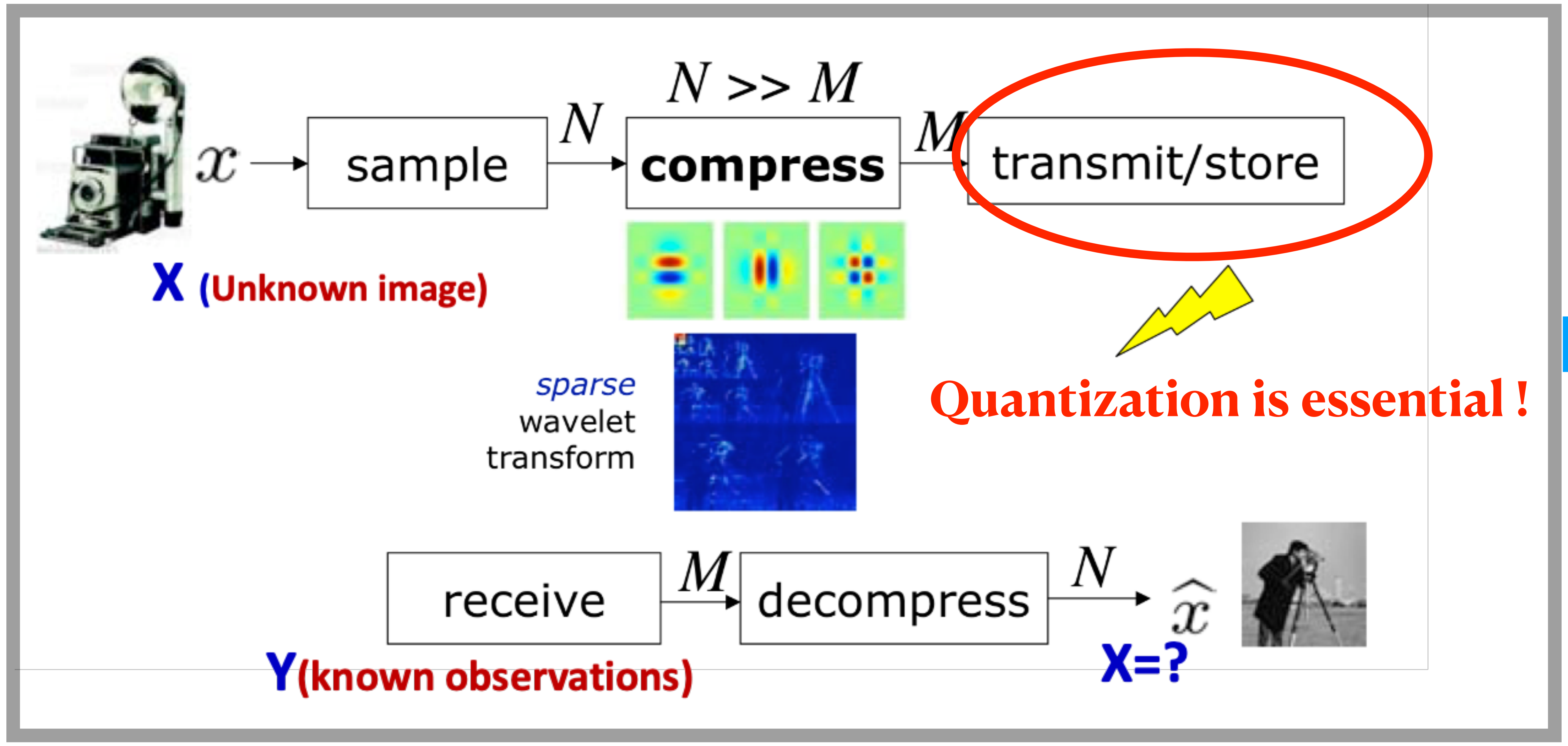
■ Quantized Compressed Sensing

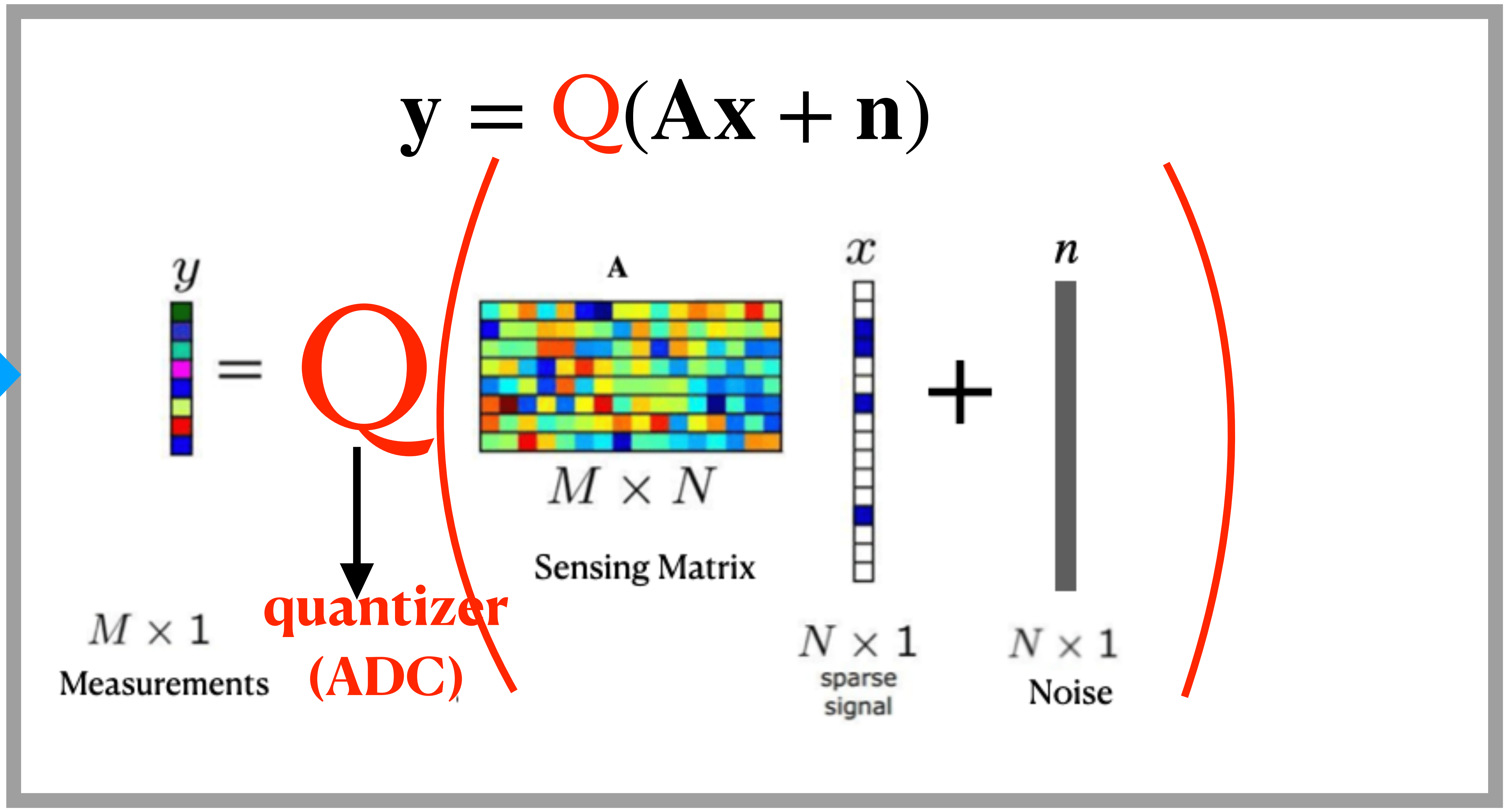




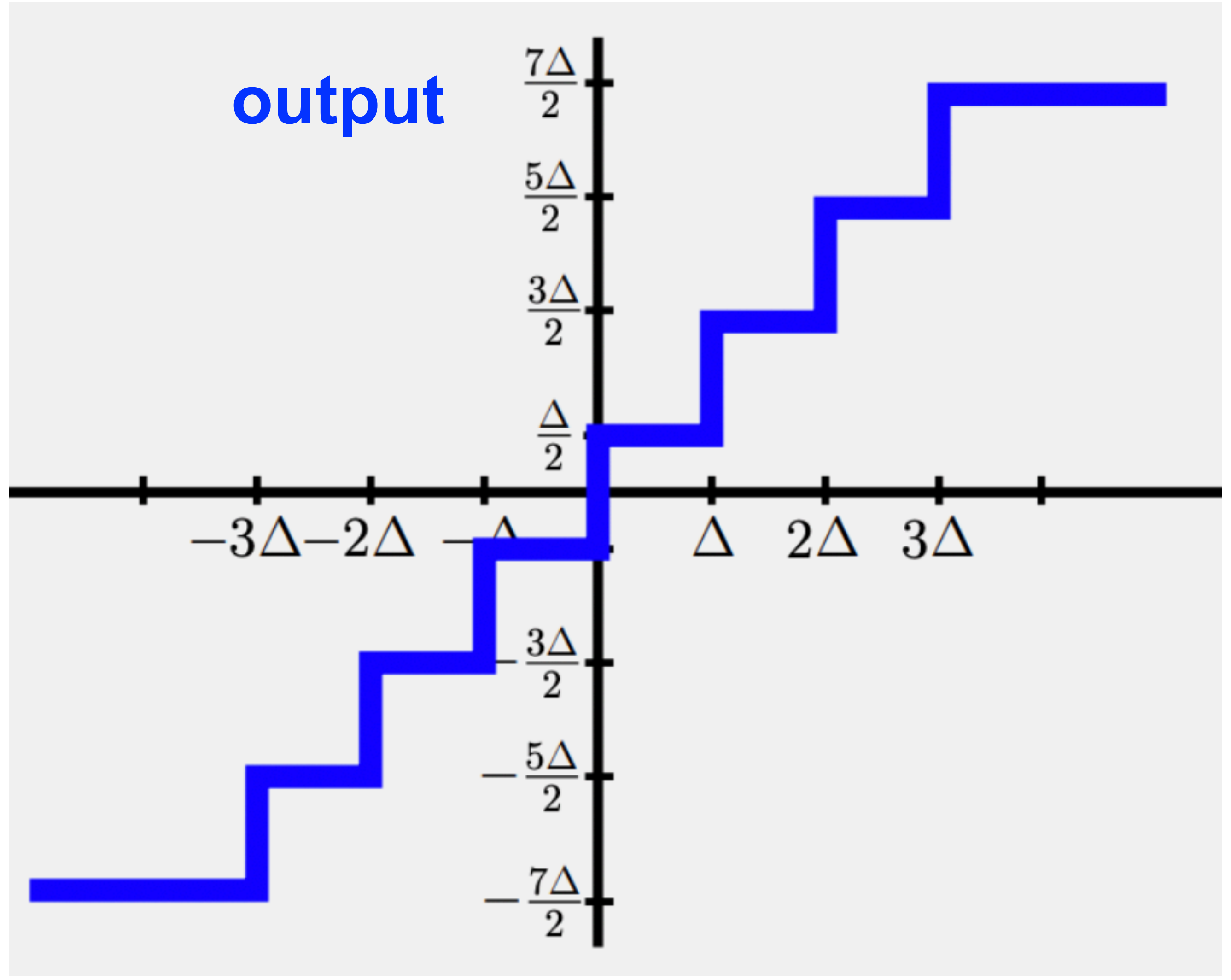
Background

■ Quantized Compressed Sensing





- **Quantizer (ADC converter)**



input

Extreme case: 1-bit quantization

$$\mathbf{y} = \mathrm{sign}(\mathbf{Ax} + \mathbf{n}) \in \{-1, +1\}^M$$

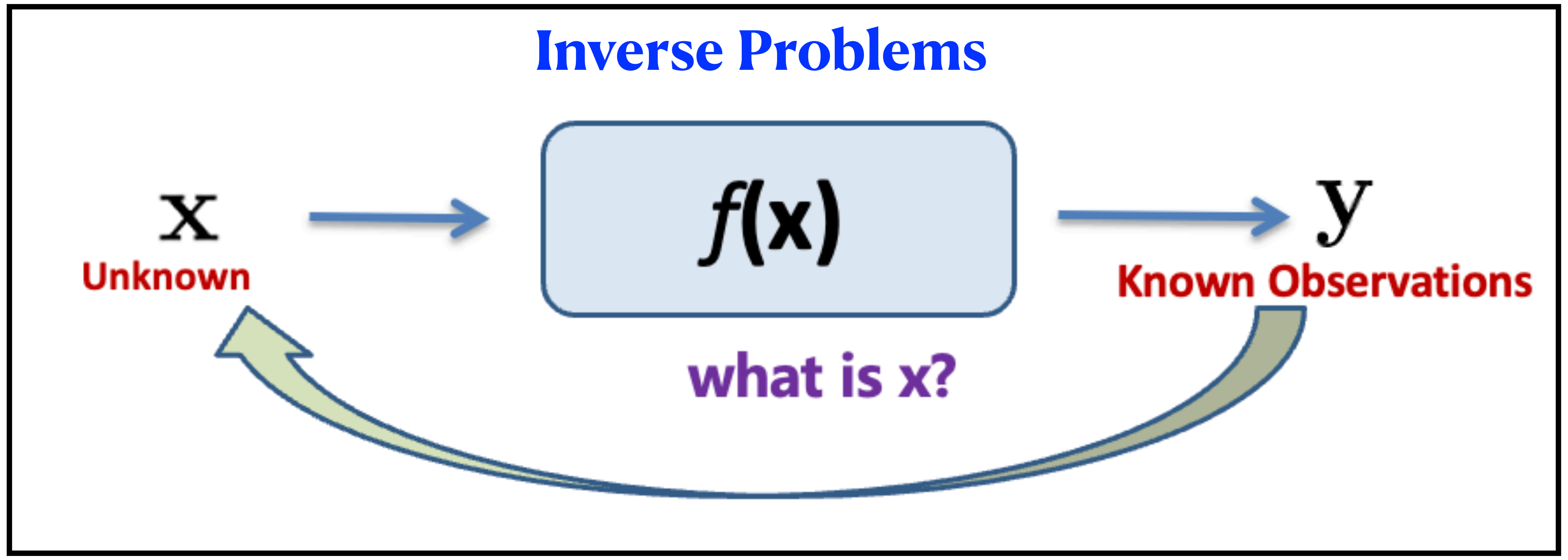
Practical Challenges

✓ Quantization, especially low-precision quantization, leads to severe information loss

✓ Quantization is a non-linear operation, which makes the linear algorithms no longer work

A Bayesian Perspective

■ **Bayesian Perspective**



Bayesian Inference

Posterior　　Prior　　Likelihood

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

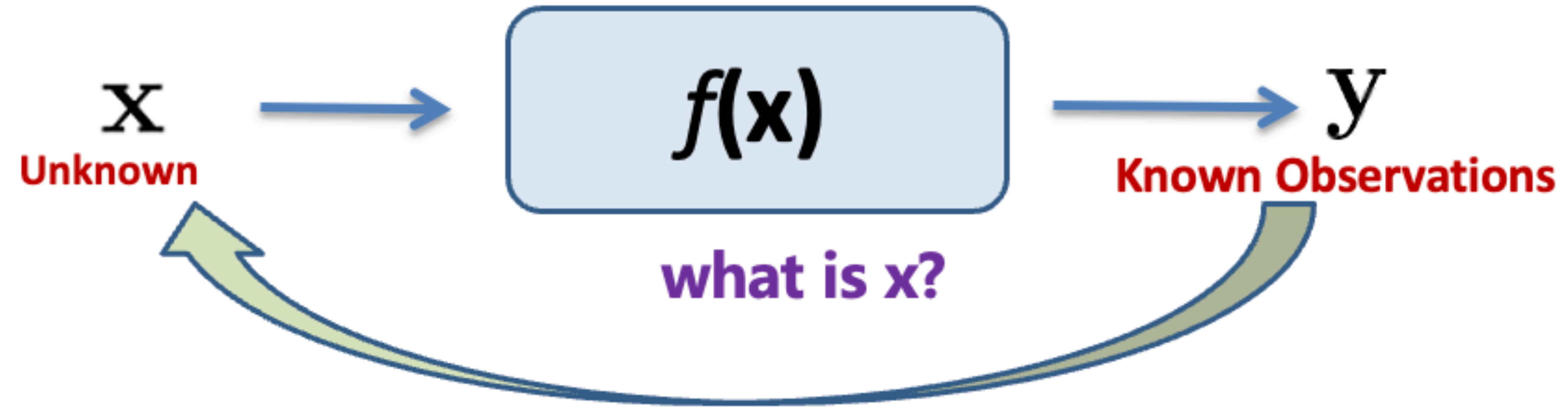
Bayes' rule

Thomas Bayes (1702-1761)

A Bayesian Perspective

■ Bayesian Perspective

Inverse Problems



Bayesian Inference

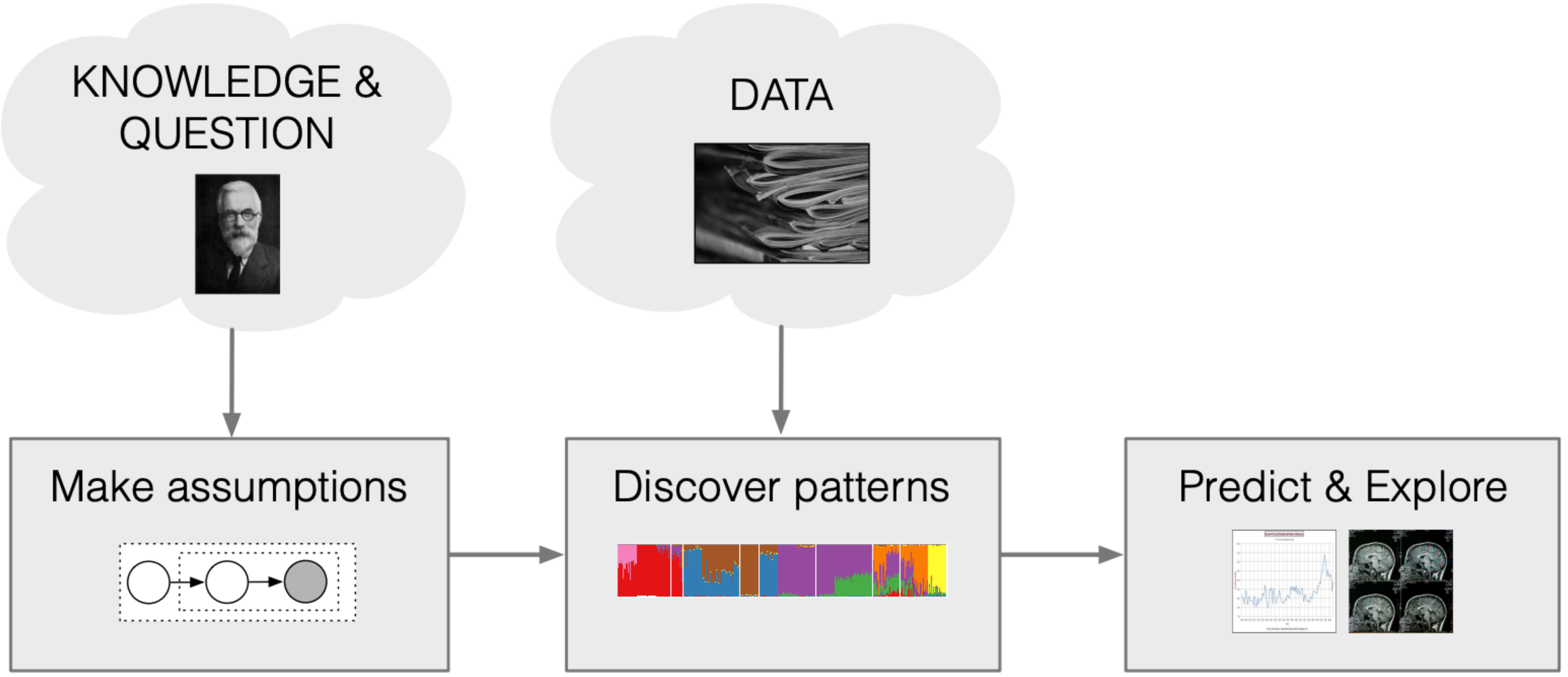
Posterior, Prior, Likelihood

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

Bayes' rule

Thomas Bayes (1702-1761)

■ Why Bayesian?



Bayesian Learning Framework [David Blei 2016]

Structure Constraint as Prior Distribution

1. The standard L1 sparsity can be viewed as a Laplace prior
2. More complicated prior, e.g., structured sparsity, and low-rankness can be used to improve performance.
3. However, hand-crafted priors might still fail to capture the rich structure in natural signals.

A Bayesian Perspective

- **Key idea**

The more you know *a priori* the less you need!

You can easily recognize someone you are familiar with at one single sight

A Bayesian Perspective

- **Key idea**

The more you know *a priori* the less you need!

You can easily recognize someone you are familiar with at one single sight

How to obtain a good prior knowledge?

Learn a good prior using powerful deep generative models

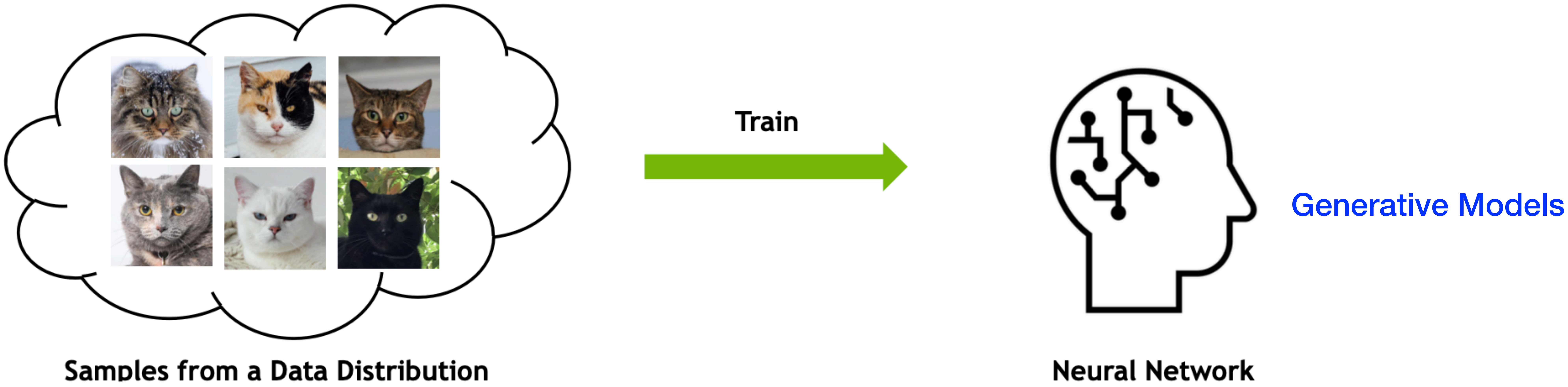
Contents

- Background: Quantized Compressed Sensing
- **Generative Models: Score-based Generative Models (SGM)**
- QCS-SGM: Quantized Compressed Sensing with SGM
- QCS-SGM+: Improved QCS-SGM for general sensing matrices

Generative Models

- **Deep Generative Models**

“What I cannot create, I do not understand” ——Richard Feynman



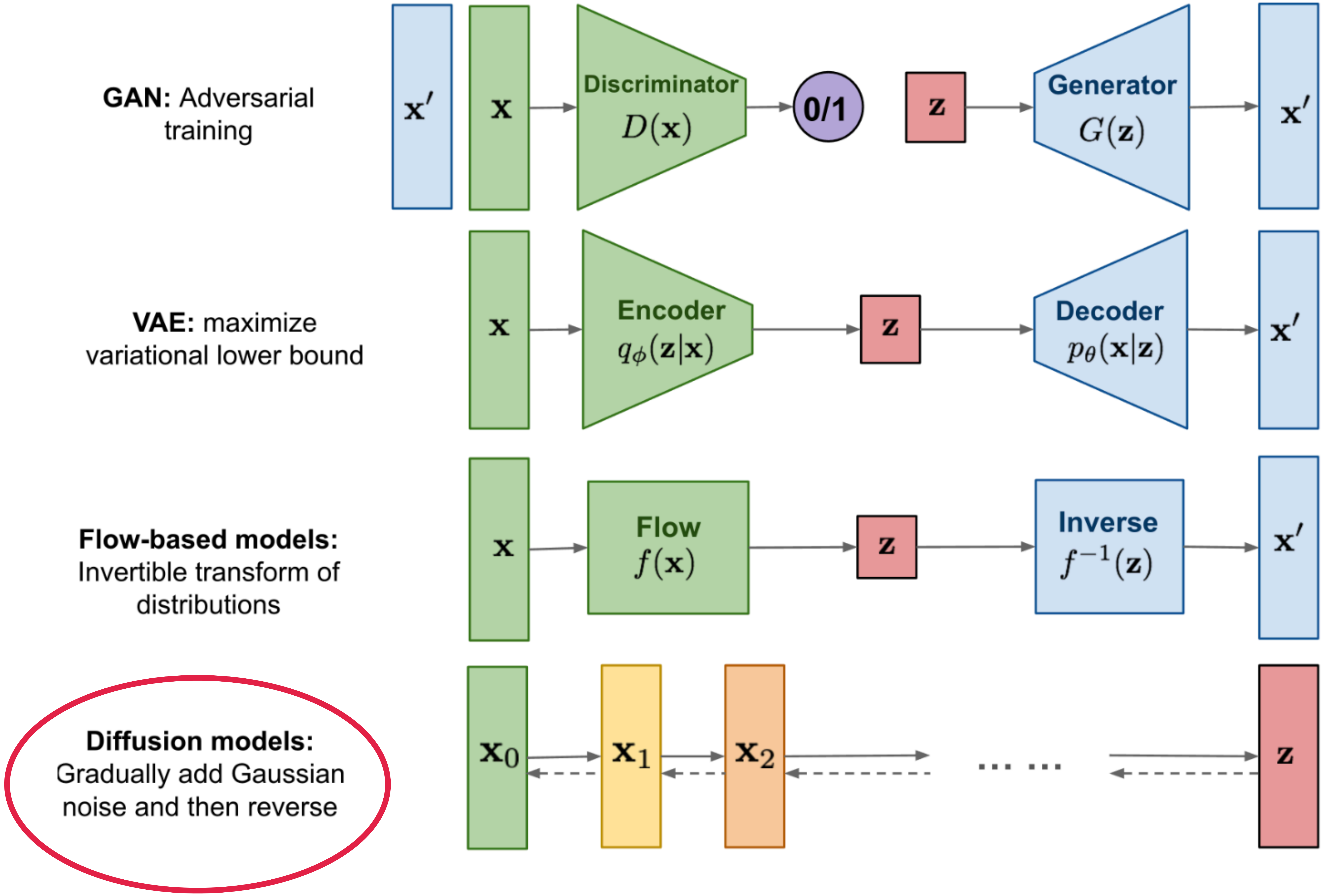
$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$$

Generative Learning

Credit to: https://cvpr2022-tutorial-diffusion-models.github.io

Generative Models

- **Overview of different types of generative models**



Diffusion Models (aka Score-based Generative models):

Emerging as the most powerful generative models !

Score-based Generative Models

- **Score-based Generative Models (SGM)**

To model the gradient of the log probability density function, known as the (Stein) score function

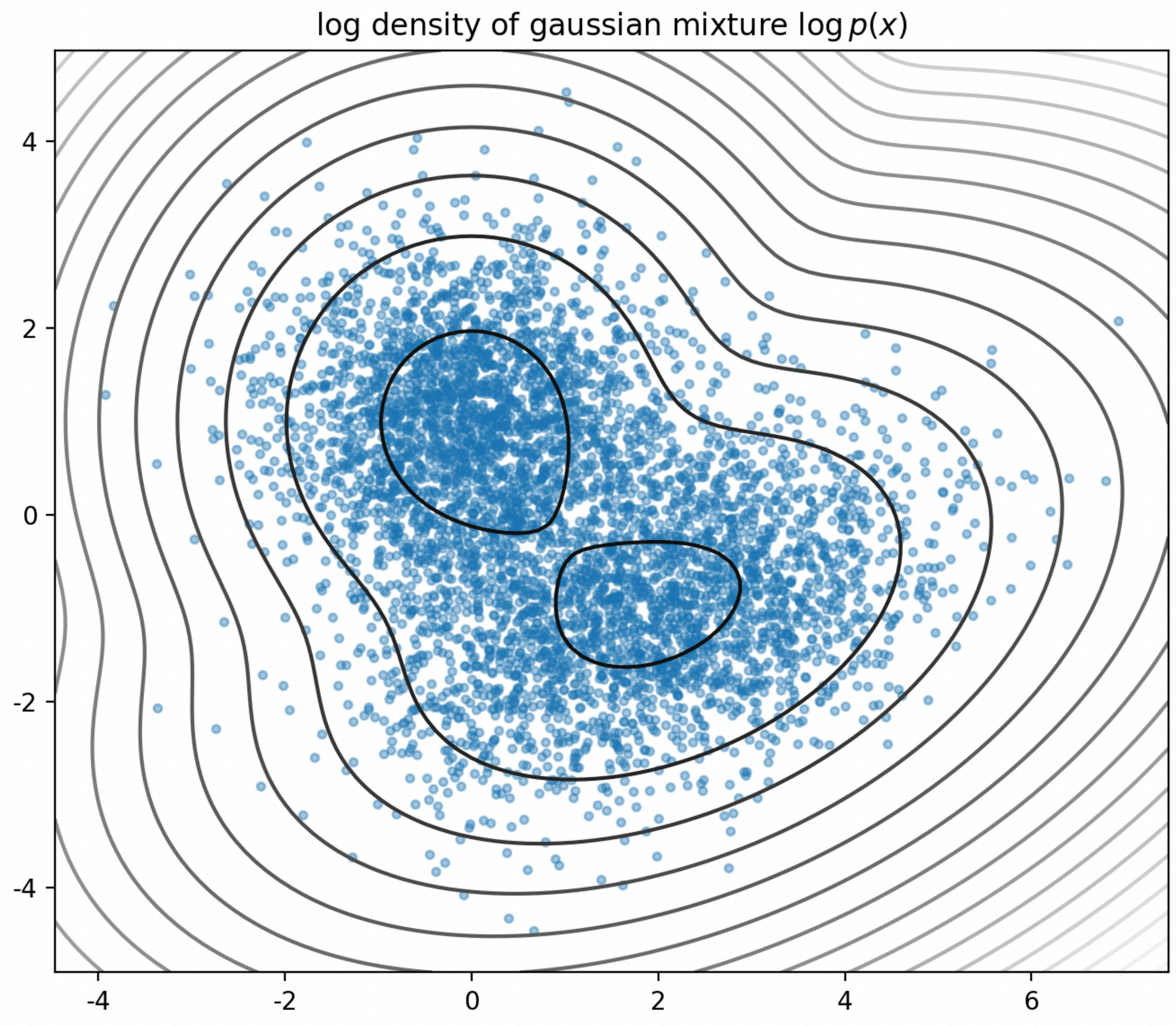
$$\nabla_{\mathbf{x}} \log p(\mathbf{x})$$ **Vector Field**

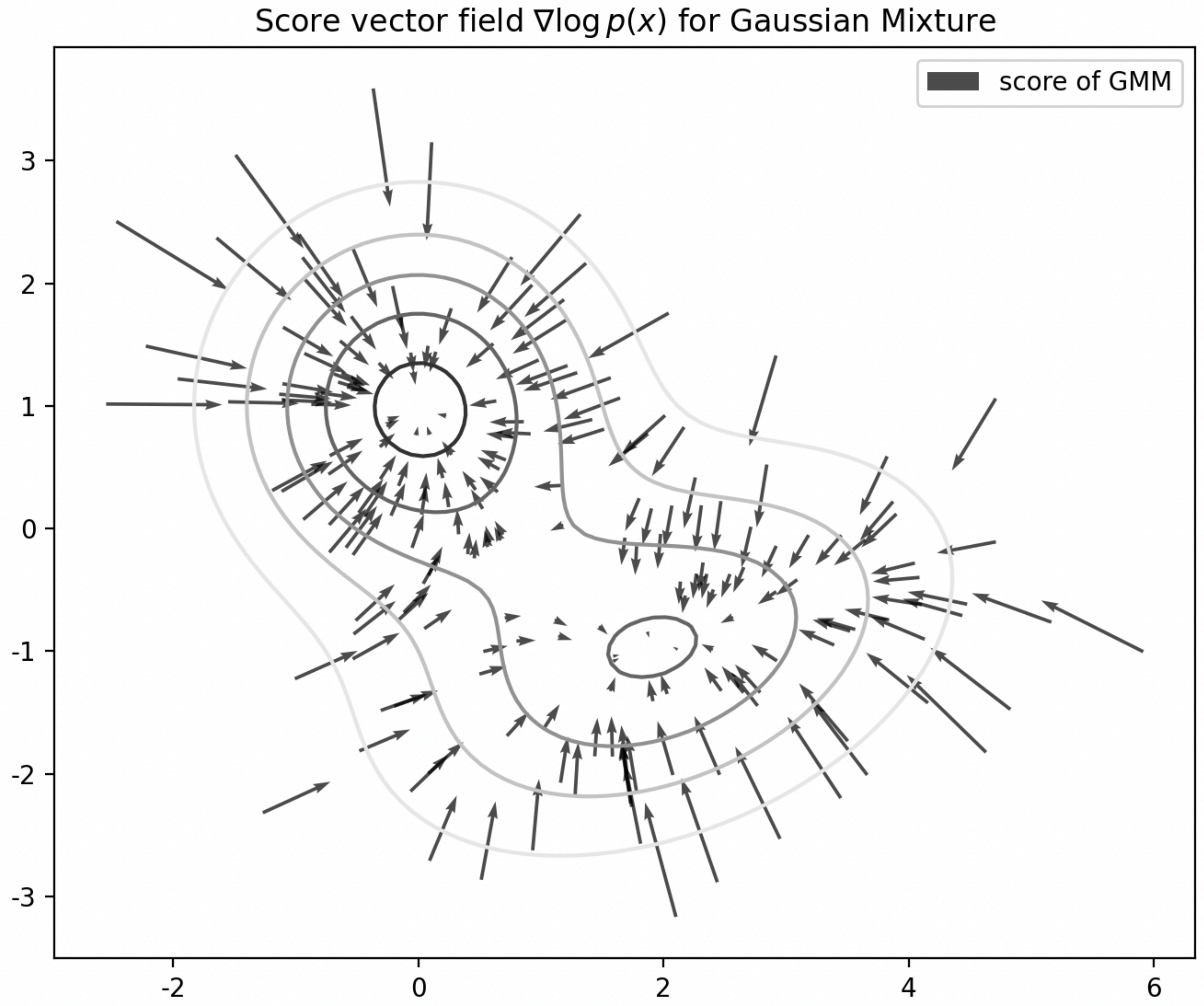
Score-based Generative Models

■ **Score-based Generative Models (SGM)**

To model the gradient of the log probability density function, known as the (Stein) score function

$$\nabla_{\mathbf{x}} \log p(\mathbf{x})$$ **Vector Field**





Score-based Generative Models

- **Why caring about score functions?**
  - **Avoiding the difficulty of intractable normalizing constants.**

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta} \qquad Z_\theta = \int e^{-f_\theta(\mathbf{x})} d\mathbf{x}$$

$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = -\nabla_\mathbf{x} f_\theta(\mathbf{x}) - \underbrace{\nabla_\mathbf{x} \log Z_\theta}_{=0} = -\nabla_\mathbf{x} f_\theta(\mathbf{x})$$

Score-based Generative Models

- **Why caring about score functions?**

◆ Avoiding the difficulty of intractable normalizing constants.

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta} \qquad Z_\theta = \int e^{-f_\theta(\mathbf{x})} d\mathbf{x}$$

$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = -\nabla_\mathbf{x} f_\theta(\mathbf{x}) - \underbrace{\nabla_\mathbf{x} \log Z_\theta}_{=0} = -\nabla_\mathbf{x} f_\theta(\mathbf{x})$$

◆ Enabling sampling using Langevin dynamics G. Parisi 1981

$\mathbf{x}_K$ converges to samples from $p(\mathbf{x})$ when $\epsilon \to 0, K \to \infty$

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_\mathbf{x} \log p(\mathbf{x}) + \sqrt{2\epsilon}\, \mathbf{z}_i, \quad i = 0, 1, \cdots, K \qquad \mathbf{z}_i \sim \mathcal{N}(0, I).$$

Score-based Generative Models

- **Why caring about score functions?**

◆ Avoiding the difficulty of intractable normalizing constants.

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta} \qquad Z_\theta = \int e^{-f_\theta(\mathbf{x})} d\mathbf{x}$$

$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = -\nabla_\mathbf{x} f_\theta(\mathbf{x}) - \underbrace{\nabla_\mathbf{x} \log Z_\theta}_{=0} = -\nabla_\mathbf{x} f_\theta(\mathbf{x})$$

◆ Enabling sampling using Langevin dynamics G. Parisi 1981

$\mathbf{x}_K$ converges to samples from $p(\mathbf{x})$ when $\epsilon \to 0, K \to \infty$

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_\mathbf{x} \log p(\mathbf{x}) + \sqrt{2\epsilon}\, \mathbf{z}_i, \quad i = 0, 1, \cdots, K \qquad \mathbf{z}_i \sim \mathcal{N}(0, I).$$

How to obtain the score function?

Score-based Generative Models

- **Noise Perturbed Score-Matching** Song et al 2019

Annealing: using multiple noise scales $\{\beta_t\}_{t=1}^{T}$ for the perturbation!

$$\mathbf{x}_t = \mathbf{x} + \beta_t \mathbf{z} \qquad 0 < \beta_1 < \beta_2 < \cdots < \beta_T$$

$$p_{\beta_t}(\mathbf{x}_t) = \int p(\mathbf{x}) N(\mathbf{x}_t \mid \mathbf{x}, \beta_t^2) d\mathbf{x}$$

Score-based Generative Models

- **Noise Perturbed Score-Matching** Song et al 2019

Annealing: using multiple noise scales $\{\beta_t\}_{t=1}^{T}$ for the perturbation!

$$\mathbf{x}_t = \mathbf{x} + \beta_t \mathbf{z} \qquad 0 < \beta_1 < \beta_2 < \cdots < \beta_T$$

$$p_{\beta_t}(\mathbf{x}_t) = \int p(\mathbf{x}) N(\mathbf{x}_t \,|\, \mathbf{x}, \beta_t^2) d\mathbf{x}$$

Noise Conditional Score Network (NCSN)

Using neural network to estimate the score $\nabla_{\mathbf{x}_t} \log p_{\beta_t}(\mathbf{x}_t)$ of each noise-perturbed distribution $p_{\beta_t}(\mathbf{x}_t)$

$$\mathbf{s}_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_{\beta_t}(\mathbf{x}_t) \ \forall t$$

Estimated Score   True Score

Loss function:

$$\sum_{t=1}^{T} \lambda_t \mathbf{E}_{p_{\beta_t}(\mathbf{x}_t)} \| \nabla_{\mathbf{x}_t} \log p_{\beta_t}(\mathbf{x}_t) - \mathbf{s}_\theta(\mathbf{x}_t, t) \|^2$$

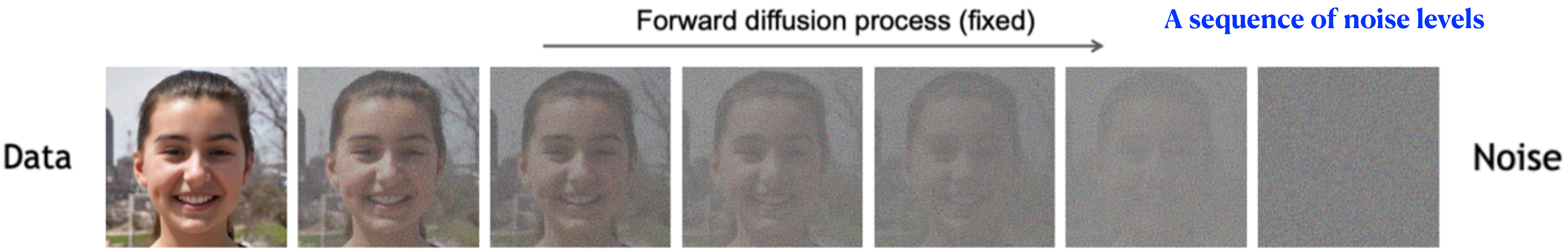
Score-based Generative Models

- **A Big Picture**

$$\mathbf{x}_t = \mathbf{x}_0 + \beta_t \mathbf{z}_t$$

$$0 < \beta_1 < \beta_2 < \cdots < \beta_T$$

A sequence of noise levels



Forward Process

Score-based Generative Models

- **A Big Picture**

$$\mathbf{x}_t = \mathbf{x}_0 + \beta_t \mathbf{z}_t \qquad 0 < \beta_1 < \beta_2 < \cdots < \beta_T$$

Forward diffusion process (fixed)

A sequence of noise levels

Reverse denoising process (generative)

Forward Process

$$\mathbf{x}_{t-1}^k = \mathbf{x}_t^k + \alpha_t \boxed{\nabla_{\mathbf{x}_t} \log p_{\beta_t}(\mathbf{x}_t)} + \sqrt{2\alpha_t}\mathbf{z}_t^k$$

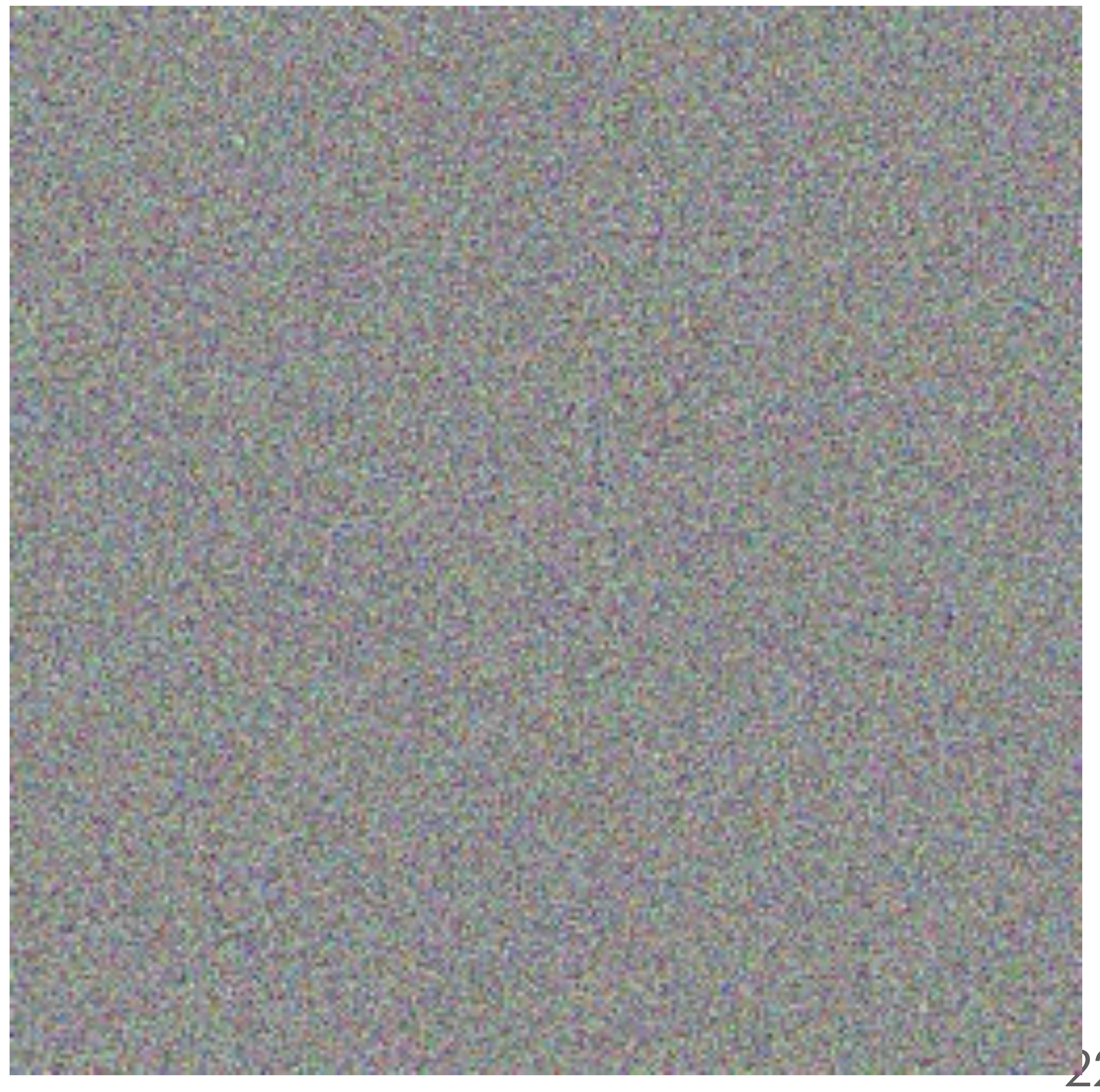
Score function

Approximated by neural network

$\mathbf{s}_\theta(\mathbf{x}_t, t)$

Annealed Langevin dynamics

Reverse Process



Score-based Generative Models

- **Connection to denoising diffusion probabilistic models (DDPM)**

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \qquad \alpha_1 > \alpha_2 > \cdots > \alpha_T > 0$$

Forward diffusion process (fixed)

Data

Noise

Reverse denoising process (generative)

The forward noise ϵ_t is estimated by a denoting network $\epsilon_\theta(\mathbf{x}_t, t)$

DDPM loss: $L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right\|^2\right]$

After some reformulation

Score Matching Loss $L_{\text{SM}} = \mathbf{E}_{t,\mathbf{x},\mathbf{x}_t}\|\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t) - \mathbf{s}_\theta(\mathbf{x}_t, t)\|^2$

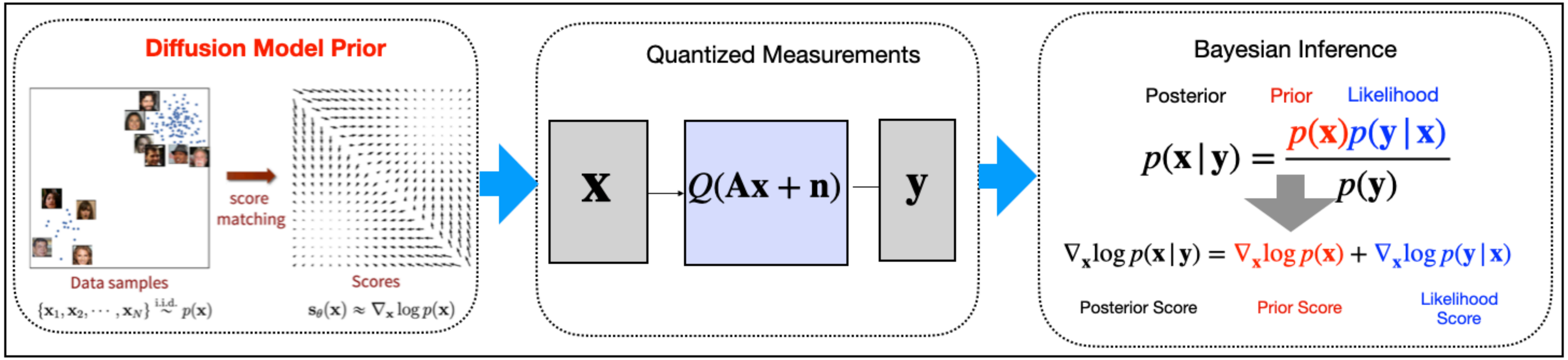
Score Estimation of $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)$

$$\mathbf{s}_\theta(\mathbf{x}_t, t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)$$

Contents

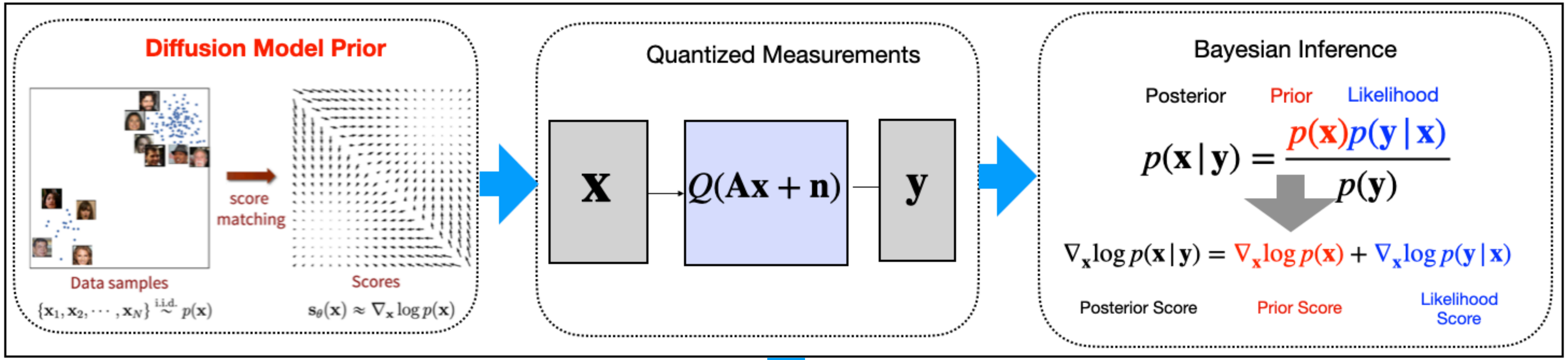
QCS-SGM: Quantized CS with SGM

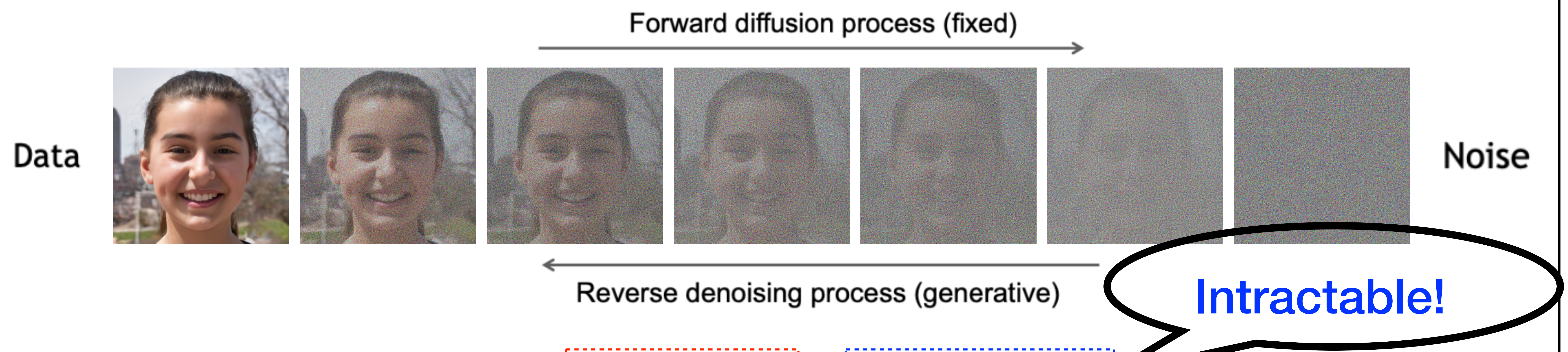
- **Our solution: QCS-SGM**



QCS-SGM: Quantized CS with SGM

- **Our solution: QCS-SGM**





Iterative Sampling Equation

$$\mathbf{x}_{t-1} = \mathbf{x}_t + \alpha_t [\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t)] + \sqrt{2\alpha_t}\mathbf{z}_t$$

Samples From diffusion models **From quantized measurements**

QCS-SGM: Quantized CS with SGM

- **Our solution: QCS-SGM**

$$p(\mathbf{y} \mid \mathbf{x}_t) = \int p(\mathbf{y} \mid \mathbf{x}_0) p(\mathbf{x}_0 \mid \mathbf{x}_t) d\mathbf{x}_0$$

Perturbed signal ($\mathbf{x}_t$) **Original signal** ($\mathbf{x}_0$) **Reverse transition probability** ($p(\mathbf{x}_0 \mid \mathbf{x}_t)$)

Using the Bayes' rule:

$$p(\mathbf{x}_0 \mid \mathbf{x}_t) = \frac{p(\mathbf{x}_t \mid \mathbf{x}_0) p(\mathbf{x}_0)}{\int p(\mathbf{x}_t \mid \mathbf{x}_0) p(\mathbf{x}_0) d\mathbf{x}_0}$$

Tractable (Gaussian): $p(\mathbf{x}_t \mid \mathbf{x}_0)$ **unknown**: $p(\mathbf{x}_0)$

Note: The result is intractable even for linear model $\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{n}$

QCS-SGM: Quantized CS with SGM

■ **Two Assumptions of QCS-SGM**

- **Assumption 1**

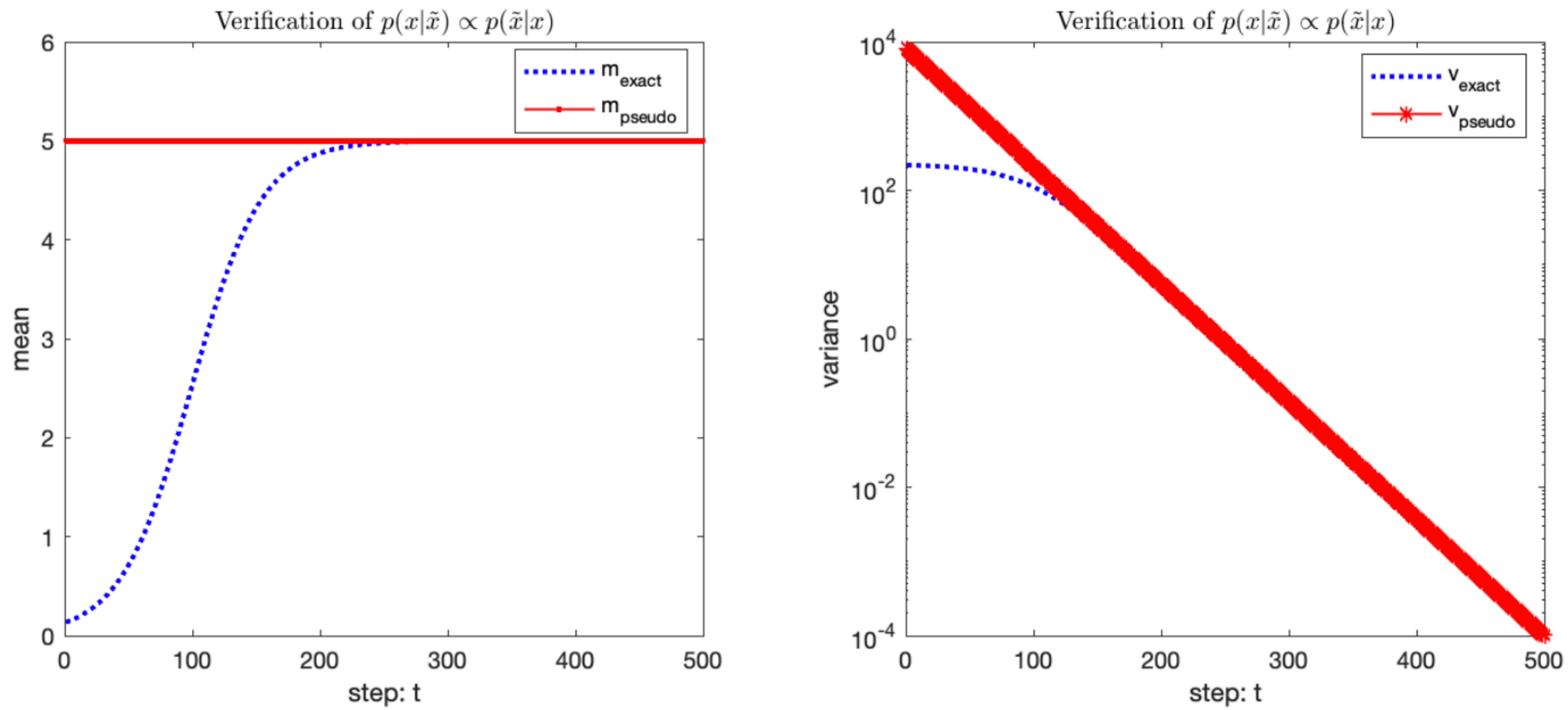
The prior $p(\mathbf{x_0})$ is non-informative w.r.t. $p(\mathbf{x}_t \mid \mathbf{x}_0)$

$$p(\mathbf{x}_0 \mid \mathbf{x}_t) \propto p(\mathbf{x}_t \mid \mathbf{x}_0)\cancel{p(\mathbf{x}_0)} \propto p(\mathbf{x}_t \mid \mathbf{x}_0)$$

Closed-form Approximation

$$p(\mathbf{x}_0 \mid \mathbf{x}_t) \approx N(\mathbf{x}_0 \mid \mathbf{x}_t, \beta_t^2 \mathbf{I})$$

Asymptotically accurate when the perturbed noise is negligible



QCS-SGM: Quantized CS with SGM

■ Two Assumptions of QCS-SGM

- **Assumption 1**

The prior $p(\mathbf{x_0})$ is non-informative w.r.t. $p(\mathbf{x}_t | \mathbf{x}_0)$

$$p(\mathbf{x}_0 | \mathbf{x}_t) \propto p(\mathbf{x}_t | \mathbf{x}_0) p(\mathbf{x}_0) \propto p(\mathbf{x}_t | \mathbf{x}_0)$$

→

Closed-form Approximation

$$p(\mathbf{x}_0 | \mathbf{x}_t) \approx N(\mathbf{x}_0 | \mathbf{x}_t, \beta_t^2 \mathbf{I})$$

Asymptotically accurate when the perturbed noise is negligible

- **Assumption 2**

The sensing matrix **A** is row-orthogonal, i.e.,

$$\mathbf{A}\mathbf{A}^T = \text{Diagonal matrix}$$

(Approximately) satisfied by many popular CS matrices
e.g., DFT, DCT, Hadamard, and random Gaussian matrices, etc.

QCS-SGM: Quantized CS with SGM

■ **Results of Pseudo-likelihood Score**

- **Theorem 1**: Under assumptions 1 and 2, we obtain **a closed-form solution** to the likelihood score

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) = \mathbf{A}^T \mathbf{G}(\beta_t, \mathbf{y}, \mathbf{A}, \mathbf{x}_t)$$

where

$$\mathbf{G}(\beta_t, \mathbf{y}, \mathbf{A}, \mathbf{x}_t) = [g_1, g_2, \ldots, g_M]^T \in \mathbb{R}^{M\times 1}$$

$$g_m = \frac{\exp\left(-\frac{\tilde{u}_{y_m}^2}{2}\right) - \exp\left(-\frac{\tilde{l}_{y_m}^2}{2}\right)}{\sqrt{\sigma^2 + \beta_t^2 \left\| \mathbf{a}_m^T \right\|_2^2} \int_{\tilde{l}_{y_m}}^{\tilde{u}_{y_m}} \exp\left(-\frac{t^2}{2}\right) dt} \quad \tilde{u}_{y_m} = \frac{\mathbf{a}_m^T \mathbf{x}_t - u_{y_m}}{\sqrt{\sigma^2 + \beta_t^2 \left\| \mathbf{a}_m^T \right\|_2^2}} \quad \tilde{l}_{y_m} = \frac{\mathbf{a}_m^T \mathbf{x}_t - l_{y_m}}{\sqrt{\sigma^2 + \beta_t^2 \left\| \mathbf{a}_m^T \right\|_2^2}}$$

- **Corollary**: In the special case of linear case y=Ax + n

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) = \mathbf{A}^T \left(\sigma^2 \mathbf{I} + \beta_t^2 \mathbf{A}\mathbf{A}^T\right)^{-1} (\mathbf{y} - \mathbf{A}\mathbf{x}_t)$$

✓Explain the necessity of annealing term in Jalal et al. (2021a)

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) = \frac{\mathbf{A}^T (\mathbf{y} - \mathbf{A}\mathbf{x}_t)}{\sigma^2 + \gamma_t^2}$$

✓Extend and improve Jalal et al. (2021a) in the general case

QCS-SGM: Quantized CS with SGM

■ **Resultant Algorithm**

Algorithm 1: Quantized Compressed Sensing with SGM (QCS-SGM)

Input: $\{\beta_t\}_{t=1}^{T}$, ϵ, K, $\mathbf{y}$, $\mathbf{A}$, σ^2, quantization codewords $\mathcal{Q}$ and thresholds $\{[l_q, u_q)|q \in \mathcal{Q}\}$

Initialization: $\mathbf{x}_1^0 \sim \mathcal{U}(0,1)$

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

  $\alpha_t \leftarrow \epsilon\beta_t^2/\beta_T^2$

  for $k = 1$ ***to*** K **do**

    Draw $\mathbf{z}_t^k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

    Compute $\mathbf{G}(\beta_t, \mathbf{y}, \mathbf{A}, \mathbf{x}_t^{k-1})$ as (12) (or (15) for 1-bit)

    $\mathbf{x}_t^k = \mathbf{x}_t^{k-1} + \alpha_t \left[\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{k-1}, \beta_t) + \mathbf{A}^T\mathbf{G}(\beta_t, \mathbf{y}, \mathbf{A}, \mathbf{x}_t^{k-1}) \right] + \sqrt{2\alpha_t}\mathbf{z}_t^k$

  $\mathbf{x}_{t+1}^0 \leftarrow \mathbf{x}_t^K$

Output: $\hat{\mathbf{x}} = \mathbf{x}_T^K$

Only this term is different from SGM!

Code Available: https://github.com/mengxiangming/QCS-SGM

QCS-SGM: Quantized CS with SGM

■ **Experimental Results**

(a) MNIST, $M = 200, \sigma = 0.05$ (b) CelebA, $M = 4000, \sigma = 0.001$

The proposed QCS-SGM achieves remarkably better performances

QCS-SGM: Quantized CS with SGM

■ **Experimental Results**

Results of QCS-SGM on CelebA
in the **fixed budget** case
(Q×M = 12288)

(a) Ground Truth

(b) 1-bit, $M = 12288$

(c) 2-bit, $M = 6144$

(d) 3-bit, $M = 4096$

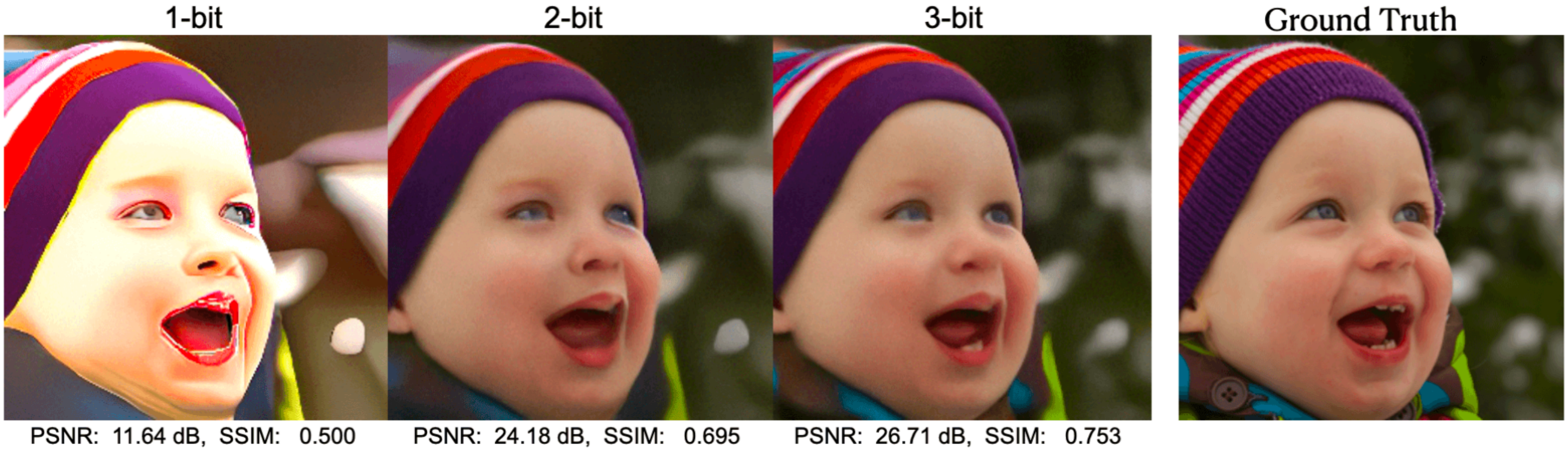
QCS-SGM: Quantized CS with SGM

■ **Experimental Results**

FFHQ 256×256 high-resolution images

Compression Ratio $\frac{M}{N} = \frac{1}{8} \ll 1$

$M = \frac{1}{8}N$



The proposed QCS-SGM can even accurately recover high-resolution image from only a few low-resolution (1,2,3-bit) quantized measurements

QCS-SGM: Quantized CS with SGM

■ **Experimental Results**

Comparison with Jalal et in the special linear case on MNIST

(a) Truth (b) ALD (Jalal et al., 2021a) (c) Ours

M = 200, σ = 0.05 and the condition number of matrix A is cond(A) = 1000

The proposed QCS-SGM outperforms the Jalal et al for general matrices

Contents

QCS-SGM+: Improved Quantized CS with SGM

■ **Limitation of QCS-SGM**

QCS-SGM is limited to
(approximately) row-orthogonal matrices A

Why? The pseudo-likelihood is otherwise intractable

$$\mathbf{C}_t^{-1} = \sigma^2\mathbf{I} + \beta_t^2\mathbf{A}\mathbf{A}^T$$

$$p(\mathbf{y}|\mathbf{x}_t) \simeq \tilde{p}(\mathbf{y}|\mathbf{z}_t = \mathbf{A}\mathbf{x}_t) = \int \prod_{m=1}^{M} \mathbb{1}\left((z_{t,m} + \tilde{n}_{t,m}) \in \mathbf{Q}^{-1}(y_m)\right)\mathcal{N}(\tilde{\mathbf{n}}_t; \mathbf{0}, \mathbf{C}_t^{-1})d\tilde{\mathbf{n}}_t$$

Intractable integration

QCS-SGM+: Improved Quantized CS with SGM

- **A New Perspective**

pseudo-likelihood

$$\underbrace{p(\mathbf{y}|\mathbf{x}_t) \simeq \tilde{p}(\mathbf{y}|\mathbf{z}_t = \mathbf{A}\mathbf{x}_t)}_{} = \int \prod_{m=1}^{M} \underbrace{\mathbb{1}\left((z_{t,m} + \tilde{n}_{t,m}) \in \mathbf{Q}^{-1}(y_m)\right)}_{} \underbrace{\mathcal{N}(\tilde{\mathbf{n}}_t; \mathbf{0}, \mathbf{C}_t^{-1})}_{} d\tilde{\mathbf{n}}_t$$

Partition Function (normalization term) Likelihood Prior

One fundamental Problem in Bayesian Inference

The pseudo-likelihood can be viewed as the partition function of random variables $\tilde{\mathbf{n}}_t$

QCS-SGM+: Improved Quantized CS with SGM

- **A New Perspective**

pseudo-likelihood

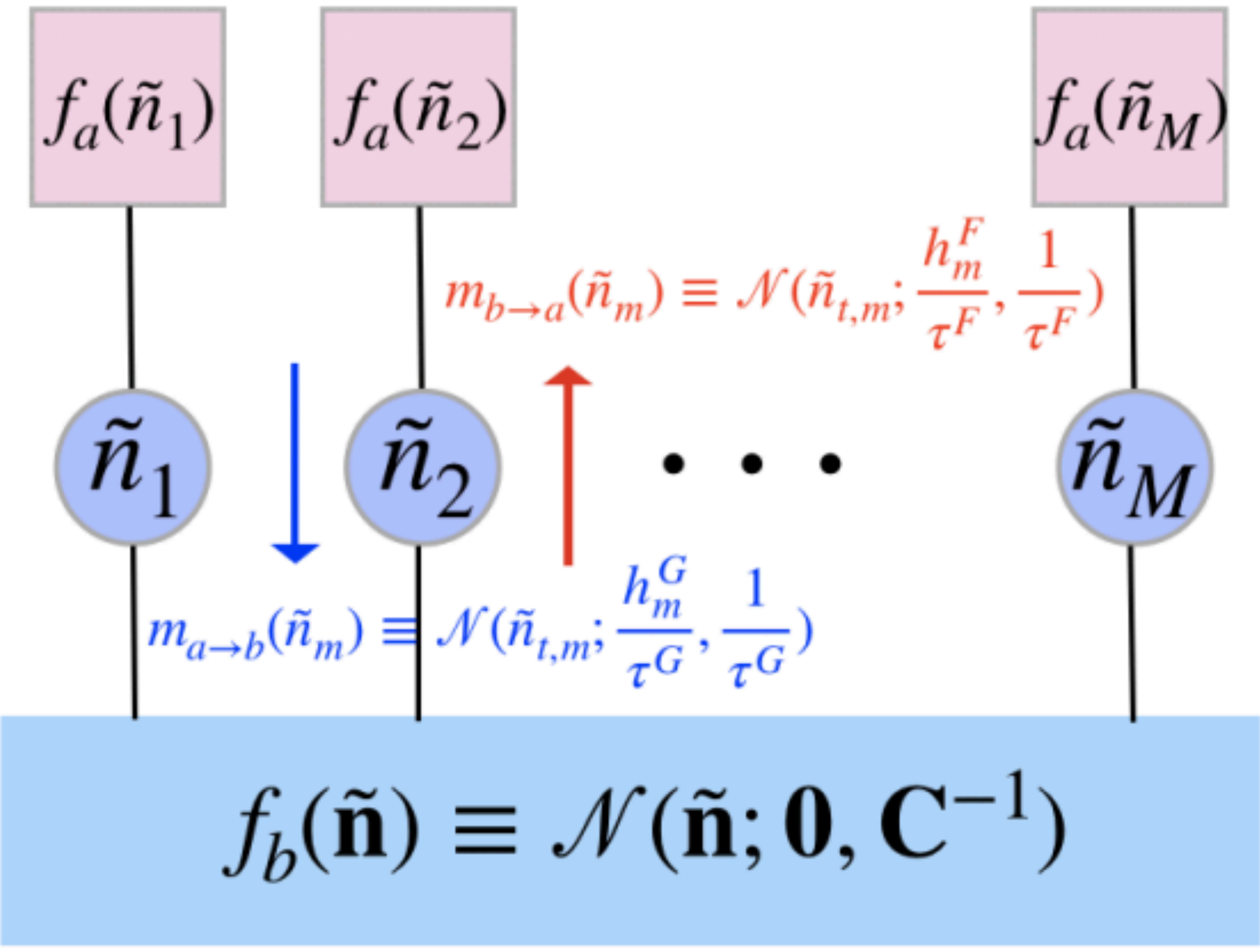
$$\underbrace{p(\mathbf{y}|\mathbf{x}_t) \simeq \tilde{p}(\mathbf{y}|\mathbf{z}_t = \mathbf{A}\mathbf{x}_t)}_{\text{Partition Function (normalization term)}} = \int \prod_{m=1}^{M} \mathbb{1}\underbrace{\left((z_{t,m} + \tilde{n}_{t,m}) \in \mathrm{Q}^{-1}(y_m)\right)}_{\text{Likelihood}} \underbrace{\mathcal{N}(\tilde{\mathbf{n}}_t; \mathbf{0}, \mathbf{C}_t^{-1})}_{\text{Prior}} d\tilde{\mathbf{n}}_t$$

One fundamental Problem in Bayesian Inference

The pseudo-likelihood can be viewed as the partition function of random variables $\tilde{\mathbf{n}}_t$

Resort to the famous expectation propagation (EP) Minka 2001

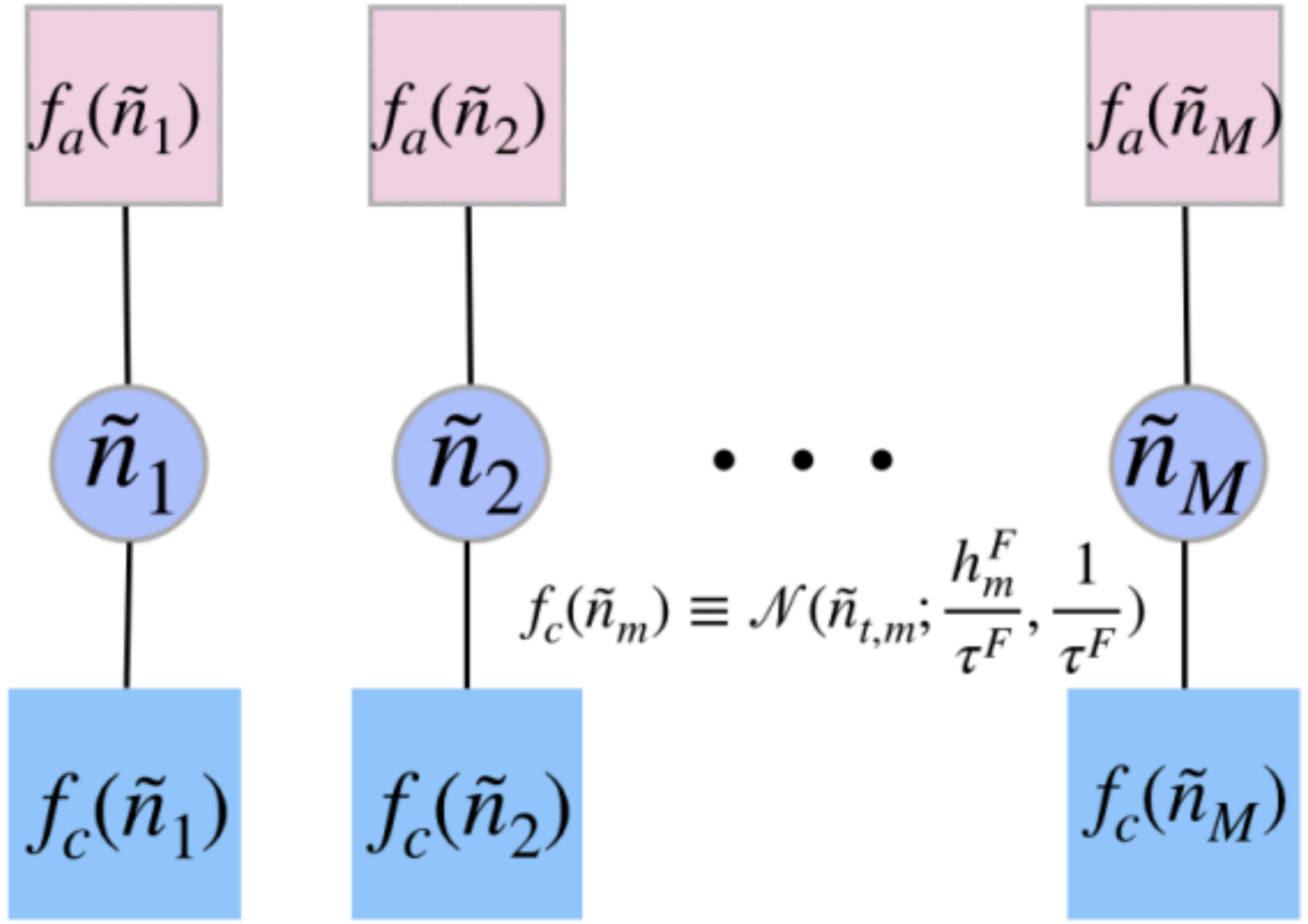
$f_a(\tilde{n}_m) \equiv 1((z_m + \tilde{n}_m) \in \mathrm{Q}^{-1}(y_m))$



(a) Original factor graph

After EP

$f_a(\tilde{n}_m) \equiv 1((z_m + \tilde{n}_m) \in \mathrm{Q}^{-1}(y_m))$



(b) Factor graph after EP

QCS-SGM+: Improved Quantized CS with SGM

- **QCS-SGM+**

Algorithm 1: QCS-SGM+

Input: $\{\beta_t\}_{t=1}^T$, ϵ, γ, $IterEP$, K, $\mathbf{y}$, $\mathbf{A}$, σ^2, quantization thresholds $\{[l_q, u_q)|q \in \mathcal{Q}\}$

Initialization: $\mathbf{x}_1^0 \sim \mathcal{U}(0, 1)$

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

  $\alpha_t \leftarrow \epsilon\beta_t^2/\beta_T^2$

  for $k = 1$ ***to*** K **do**

    Draw $\mathbf{z}_t^k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

    Initialization: $\boldsymbol{h}^F, \tau^F, \boldsymbol{h}^G, \tau^G$

    for $it = 1$ ***to*** $IterEP$ **do**

      $\boldsymbol{h}^G = \frac{\boldsymbol{m}^a}{\chi^a} - \boldsymbol{h}^F$

      $\tau^G = \frac{1}{\chi^a} - \tau^F$

      $\boldsymbol{h}^F = \frac{\boldsymbol{m}^b}{\chi^b} - \boldsymbol{h}^G$

      $\tau^F = \frac{1}{\chi^b} - \tau^G$

    Compute $\nabla_{\mathbf{x}_t} \log p_{\beta_t}(\mathbf{y} \mid \mathbf{x}_t)$ as (11)

    $\mathbf{x}_t^k = \mathbf{x}_t^{k-1} + \alpha_t \left[\mathrm{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{k-1}, \beta_t) + \gamma \nabla_{\mathbf{x}_t} \log p_{\beta_t}(\mathbf{y} \mid \mathbf{x}_t) \right] + \sqrt{2\alpha_t}\mathbf{z}_t^k$

  $\mathbf{x}_{t+1}^0 \leftarrow \mathbf{x}_t^K$

Output: $\hat{\mathbf{x}} = \mathbf{x}_T^K$

Running EP to approximate the pseudo-likelihood

Code: https://github.com/mengxiangming/QCS-SGM-plus

QCS-SGM+: Improved Quantized CS with SGM

■ **Experimental Results**

- General Matrices

(a) ill-conditioned matrices

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^{\mathbf{T}}$$

$\mathbf{V}$ and $\mathbf{U}$ are independent Harr-distributed matrices

nonzero singular values of $\mathbf{A}$ satisfy $\frac{\lambda_i}{\lambda_{i+1}} = \kappa^{1/M}$, where κ is the condition number.

(b) correlated matrices

$$\mathbf{A} = \mathbf{R}_L \mathbf{H} \mathbf{R}_R$$

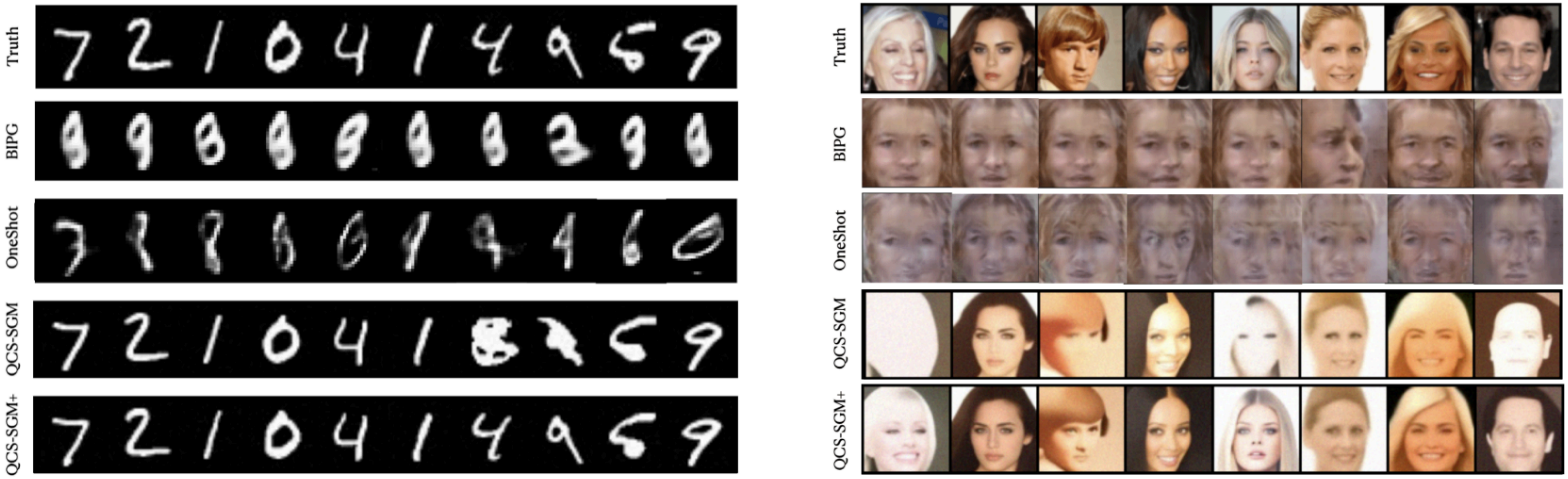
where $\mathbf{R}_L = \mathbf{R}_1^{\frac{1}{2}} \in \mathbb{R}^{M \times M}$ and $\mathbf{R}_R = \mathbf{R}_2^{\frac{1}{2}} \in \mathbb{R}^{N \times N}$, $\mathbf{H} \in \mathbb{R}^{M \times N}$ is a random matrix

The (i, j) th element of both R1 and R2 is $\rho^{|i-j|}$ and ρ is termed the correlation coefficient

QCS-SGM+: Improved Quantized CS with SGM

■ **Experimental Results**

1-bit CS on MNIST and CelebA for ill-conditioned A ($\kappa = 10^3$ for MNIST and $\kappa = 10^6$ for CelebA)

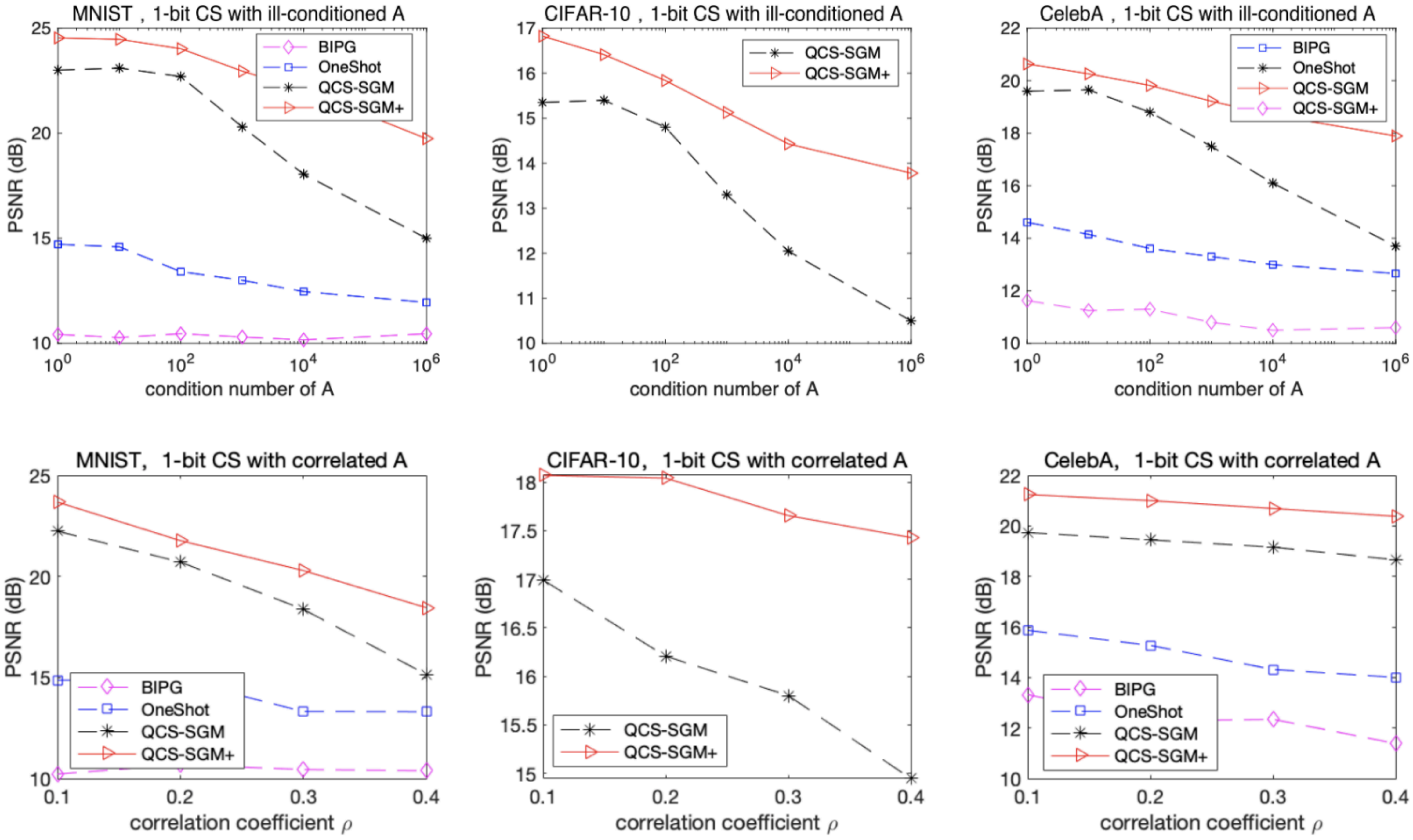


(a) MNIST, $M = 400, \sigma = 0.05, \kappa = 10^3$ (b) CelebA, $M = 4000, \sigma = 0.001, \kappa = 10^6$

It can be seen that QCS-SGM+ apparently outperforms the original QCS-SGM and other methods.

QCS-SGM+: Improved Quantized CS with SGM

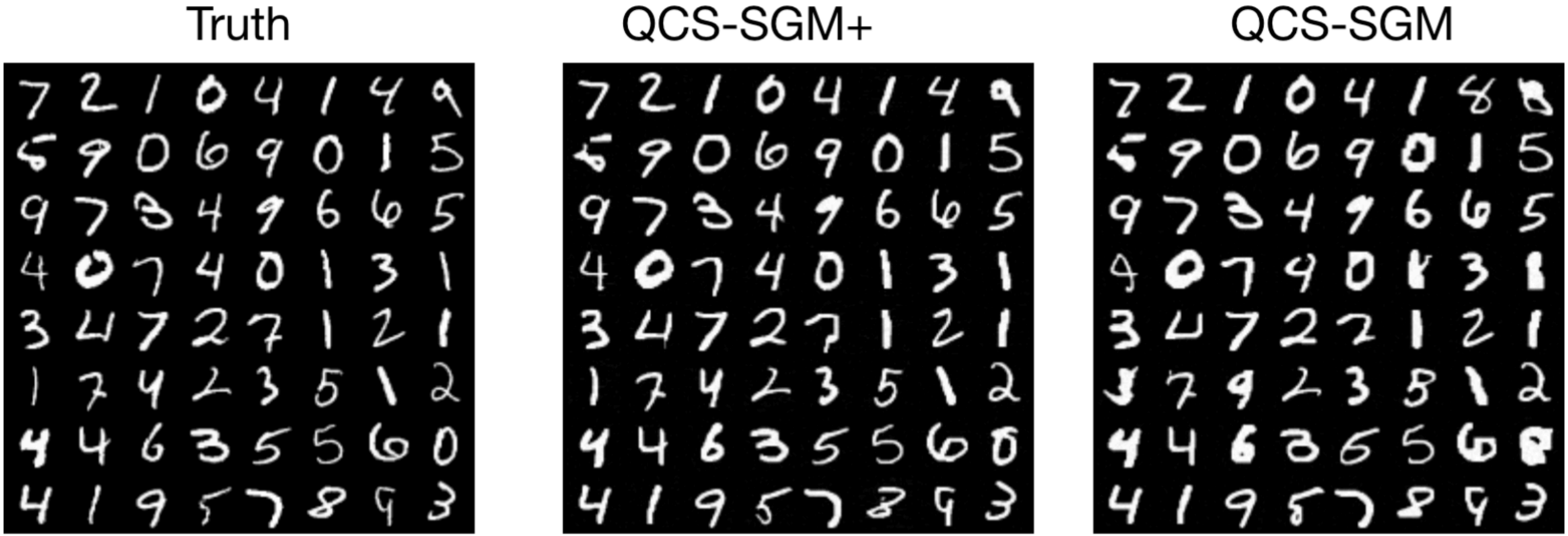
■ **Experimental Results**



It can be seen that QCS-SGM+ apparently outperforms the original QCS-SGM and other methods.

QCS-SGM+: Improved Quantized CS with SGM

■ **Experimental Results**



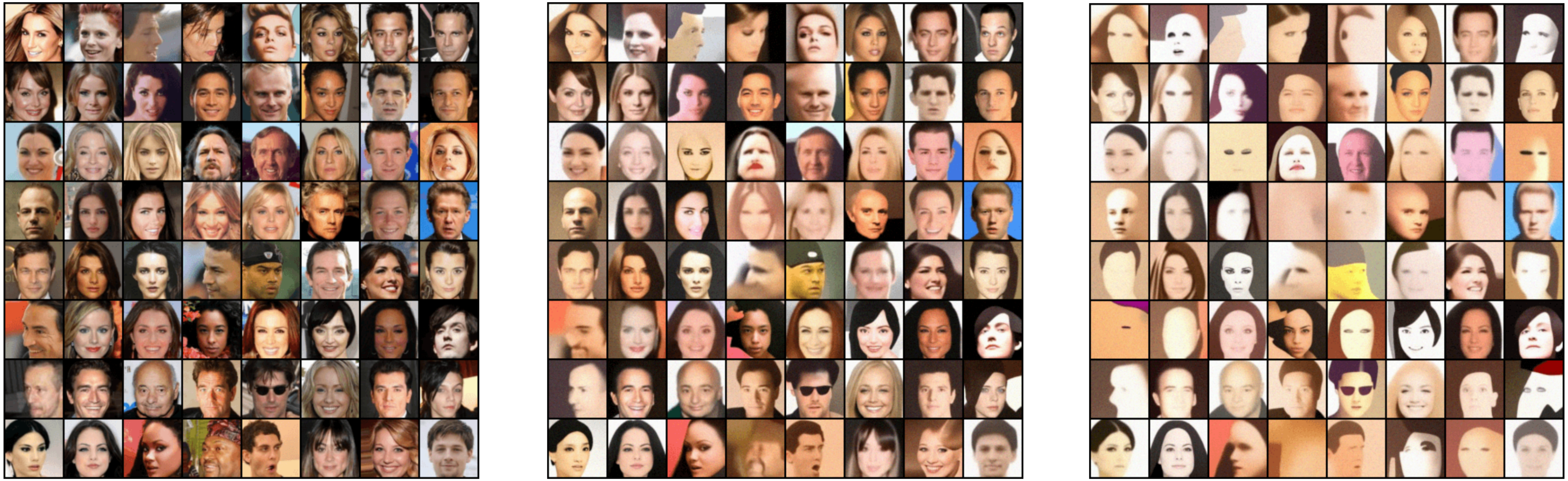
(b) 1-bit CS with correlated $\mathbf{A}, \rho = 0.4, M = 400, \sigma = 0.1$

It can be seen that QCS-SGM+ apparently outperforms the original QCS-SGM and other methods.

QCS-SGM+: Improved Quantized CS with SGM

■ Experimental Results

Truth | QCS-SGM+ | QCS-SGM



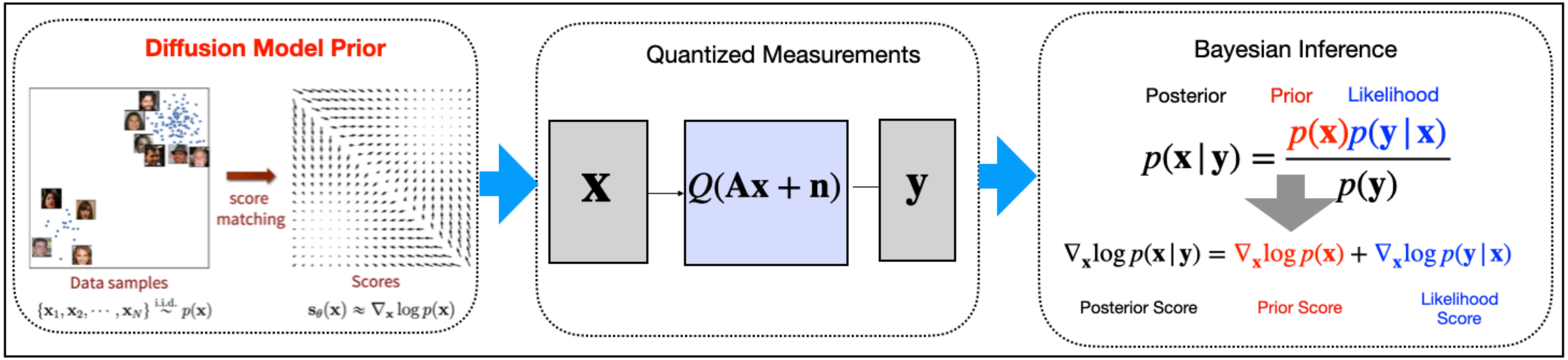
1-bit CS on CelebA for ill-conditioned A ($\kappa = 10^6$ for CelebA), $M = 4000 \ll N, \sigma = 0.1$

It can be seen that QCS-SGM+ apparently outperforms the original QCS-SGM.

Brief Summary

- **Summary**

We proposed QCS-SGM, one quantized CS algorithm using score-based models (diffusion models), as well as an advanced variant QCS-SGM+ for general sensing matrices.



Code: https://github.com/mengxiangming/QCS-SGM
Code: https://github.com/mengxiangming/QCS-SGM-plus

Thank you!

Q&A

Score-based Generative Models

- **Noise Perturbed Score-Matching**

Estimated scores are only accurate in high density regions.

Original distribution
$p(\mathbf{x})$

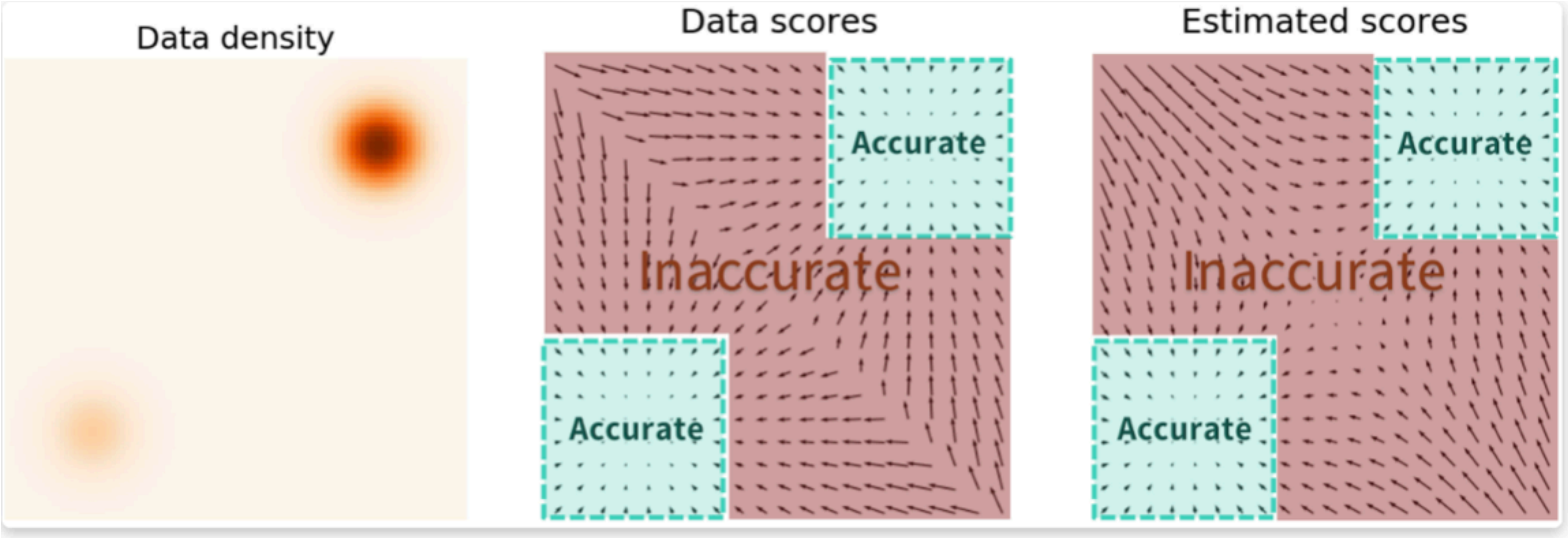


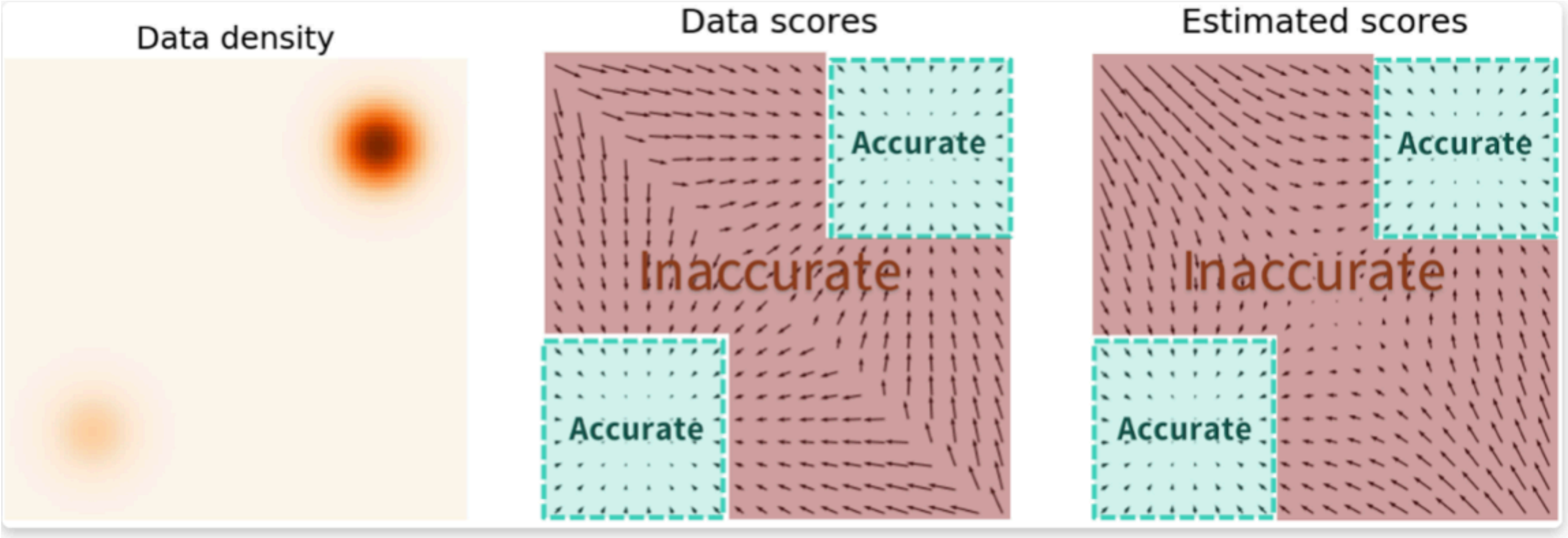
Figure credit to Yang Song

Score-based Generative Models

- **Noise Perturbed Score-Matching**

Estimated scores are only accurate in high density regions.

Original distribution
$p(\mathbf{x})$



Estimated scores are accurate everywhere for noise perturbed data

Corrupted noise

$$\mathbf{x}' = \mathbf{x} + \beta \mathbf{z}$$

Noise-perturbed
$p_\beta(\mathbf{x}')$

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

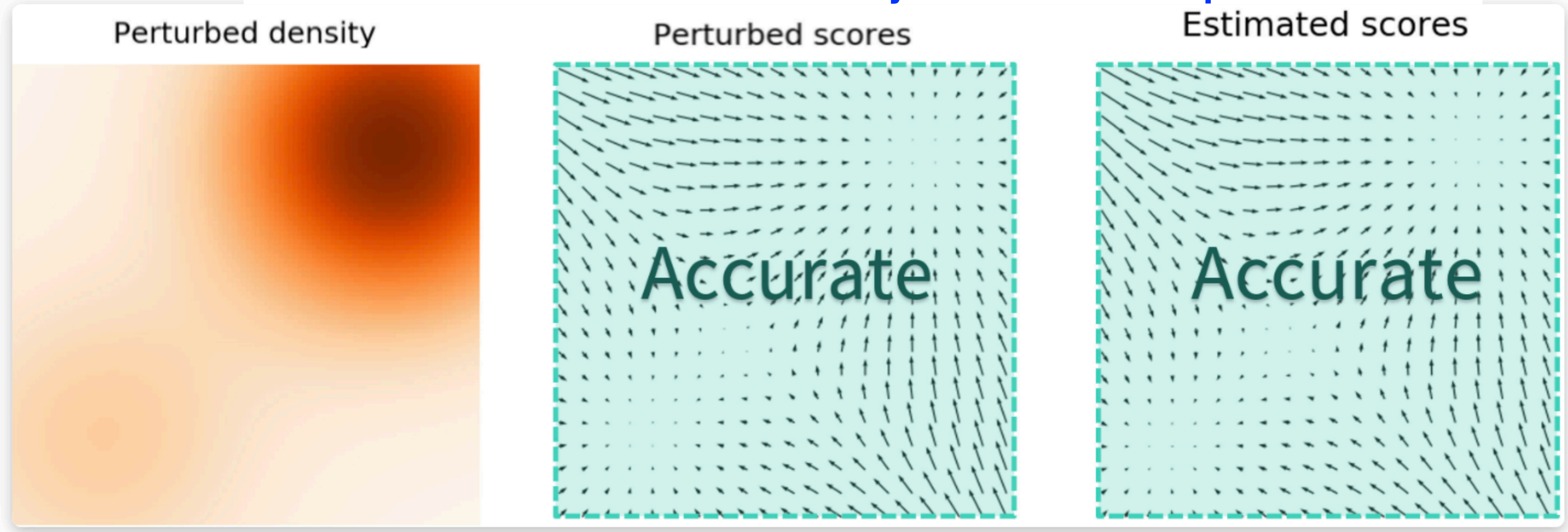


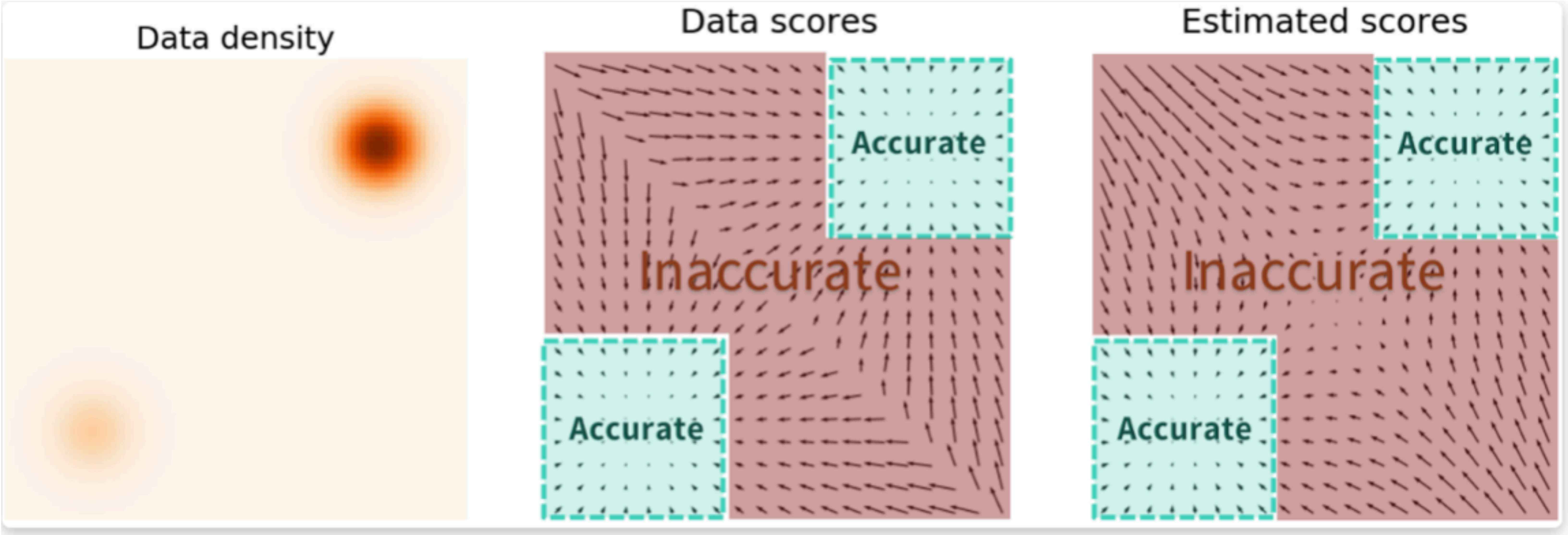
Figure credit to Yang Song

Score-based Generative Models

- Noise Perturbed Score-Matching

Estimated scores are only accurate in high density regions.

Original $p(\mathbf{x})$



Estimated scores are accurate everywhere for noise perturbed data

Q: how to choose an appropriate noise scale β for the perturbation?

Large noise: cover the low-density regions well, but different from the original distribution

Small noise: similar to the original distribution, but does not cover low-density regions well

Score-based Generative Models

- **Noise Perturbed Score-Matching**

Annealing: using multiple noise scales $\{\beta_t\}_{t=1}^{T}$ for the perturbation!

$$\mathbf{x}_t = \mathbf{x} + \beta_t \mathbf{z} \qquad 0 < \beta_1 < \beta_2 < \cdots < \beta_T$$

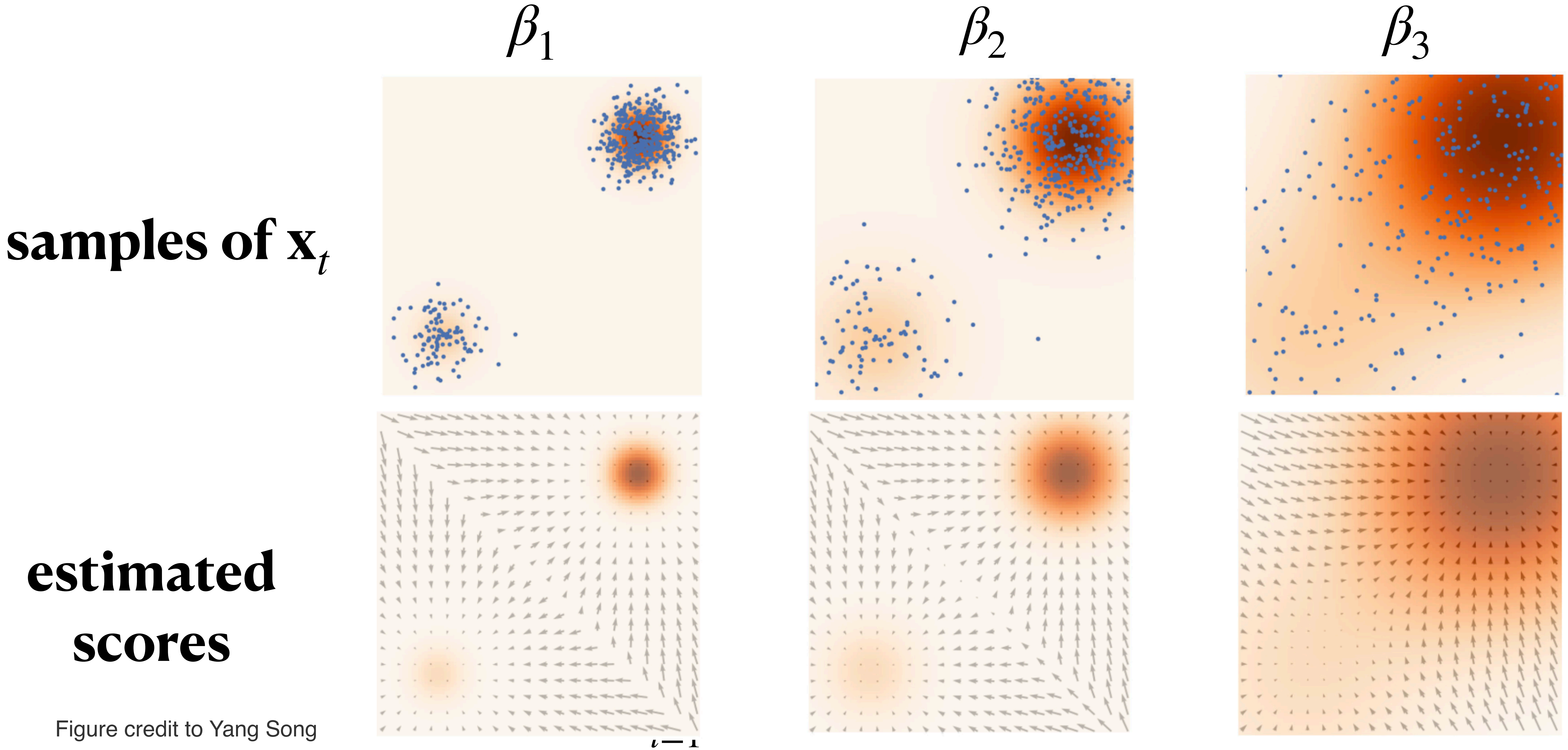


Figure credit to Yang Song