

Presentation | 2023-5-30

# **Paper Reading: Modeling Gene Regulatory Networks Using Neural Network Architectures**

---

**Yuzhe Wang<sup>1</sup>, Shan Liu<sup>2</sup>**

<sup>1</sup> Peking University

<sup>2</sup> Beijing Computational Science Research Center

# Outline

---

- Background and Motivation
  - Single-cell RNA Sequencing
  - Single-cell Computational Tasks
  - Overview of DeepSEM
- Methods
  - Variational Autoencoder (VAE)
  - Structural Equation Model (SEM)
  - DeepSEM: Framework and Implementation
- Model Performance
  - GRN Inference
  - Single-cell Clustering and Embedding
  - scRNA-seq Data Simulation

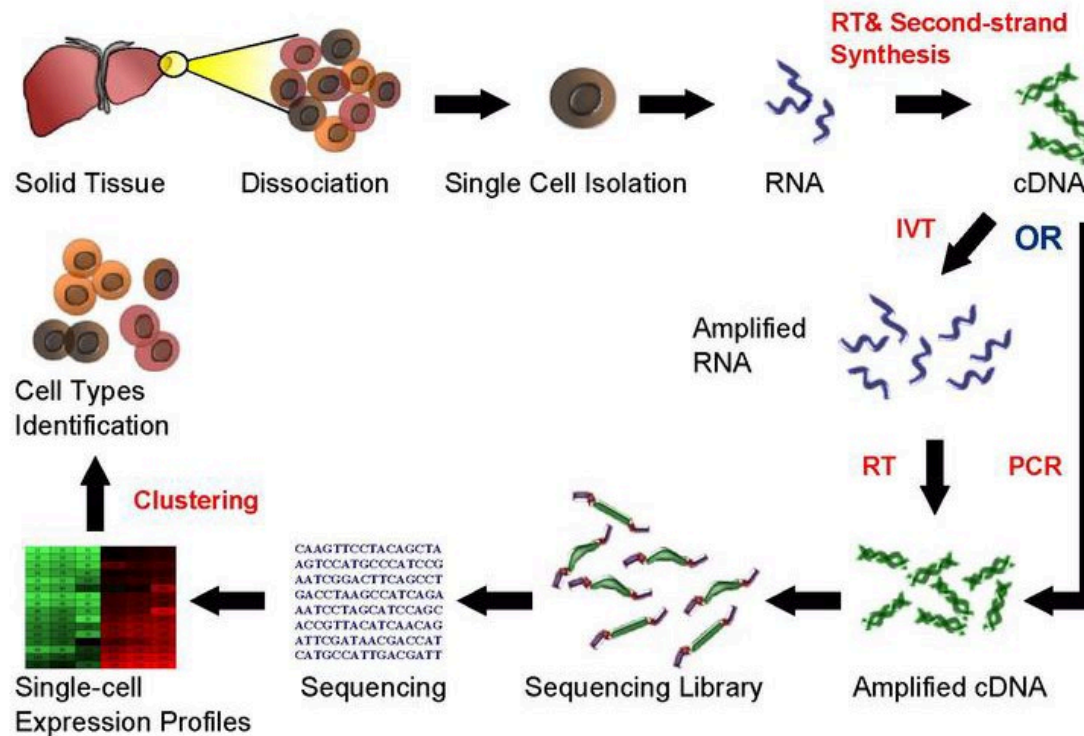
# Outline

---

- Background and Motivation
  - Single-cell RNA Sequencing
  - Single-cell Computational Tasks
  - Overview of DeepSEM
- Methods
  - Variational Autoencoder (VAE)
  - Structural Equation Model (SEM)
  - DeepSEM: Framework and Implementation
- Model Performance
  - GRN Inference
  - Single-cell Clustering and Embedding
  - scRNA-seq Data Simulation

# Single-cell RNA Sequencing

- Single-cell ENA Sequencing (scRNA-seq)
  - Experimental noise introduces biases in the gene expression
  - Deep learning benefits single-cell transcriptome data analysis



# Single-cell Computational Tasks

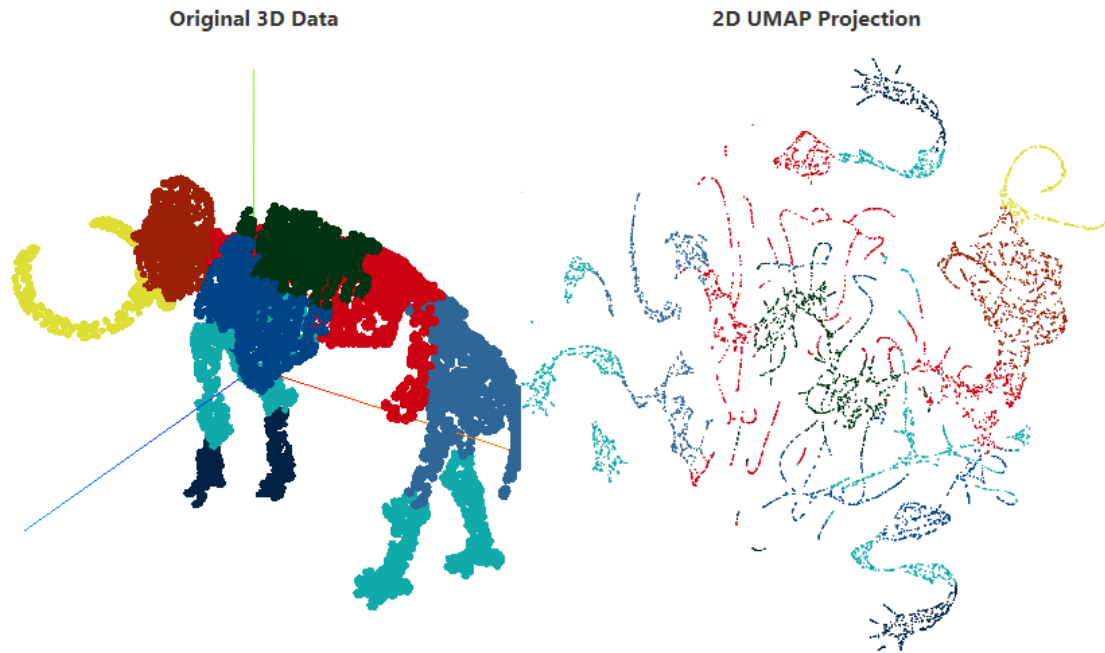
---

- GRN Inference
  - Methods relying on side measurements
    - single-cell chromatin accessibility or TF binding motifs
    - Introduce additional noise
  - Methods solely based on scRNA-seq data
    - Focus on the co-expression networks instead of decoding the casual relationships among TFs and their corresponding target genes
  - Methods that incorporate linear models or tree-based models
    - Hard to directly generalize these to more comprehensive nonlinear frameworks and benefit from the computational power that the deep learning model brought to us

# Single-cell Computational Tasks

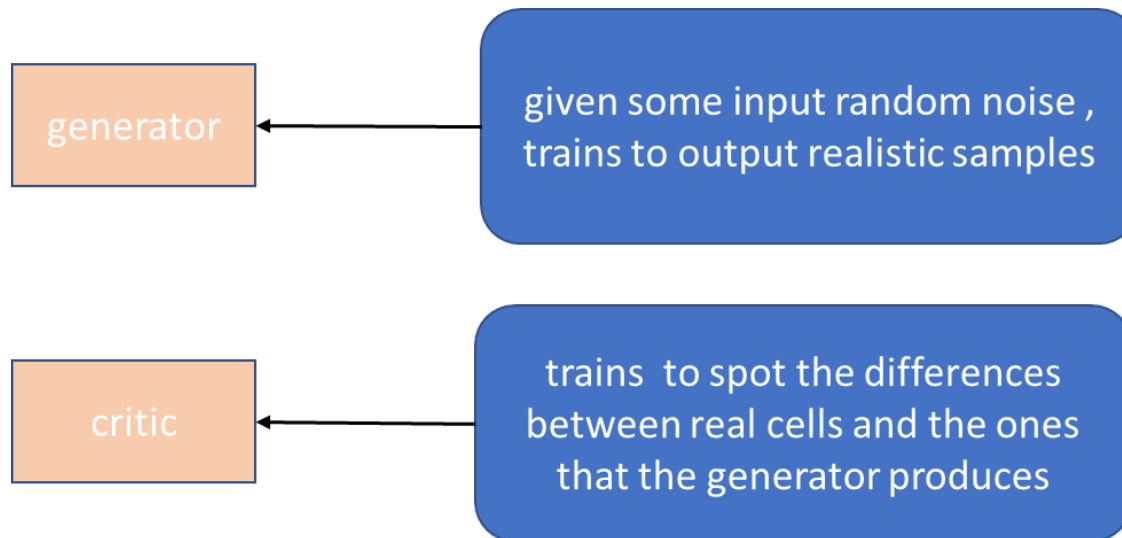
---

- scRNA-seq Data Visualization
  - Uniform Manifold Approximation Projection (UMAP)
    - The data is uniformly distributed on Riemannian manifold
    - The Riemannian metric is locally (approximately) constant
    - The manifold is locally connected



# Single-cell Computational Tasks

- scRNA-seq Data Simulation
  - Single-cell generative adversarial neural networks (scGAN)
  - Conditional single-cell generative adversarial neural networks (cscGAN)



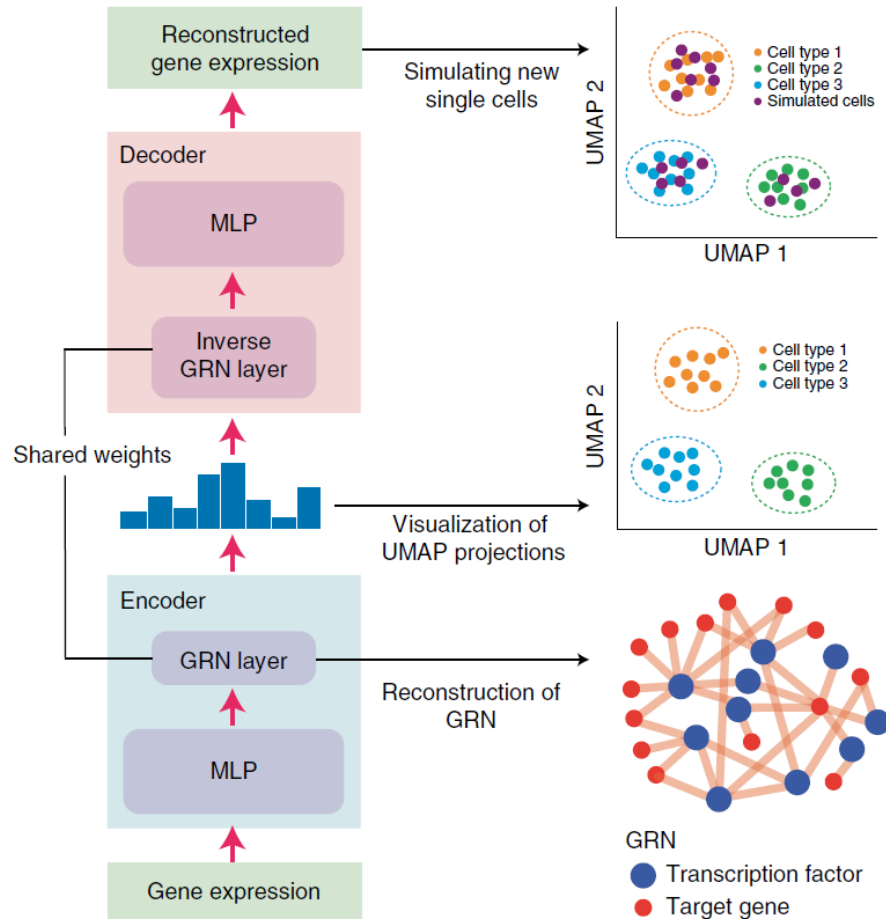
$$W(P_r, P_s) = \inf_{\gamma \in \Pi(P_r, P_s)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\|$$

Bahrami, M., Maitra, M., Nagy, C., Turecki, G., Rabiee, H. R., & Li, Y. (2021). Deep feature extraction of single-cell transcriptomes by generative adversarial network. *Bioinformatics*, 37(10), 1345-1351.

Marouf, M., Machart, P., Bansal, V., Kilian, C., Magruder, D. S., Krebs, C. F., & Bonn, S. (2020). Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. *Nature communications*, 11(1), 166.

# Overview of DeepSEM

- DeepSEM:  $\beta$ -VAE + SEM





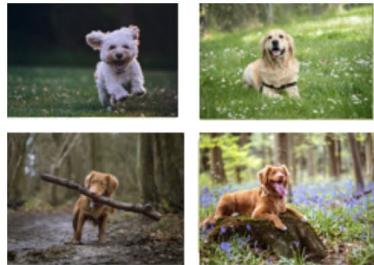
# Outline

---

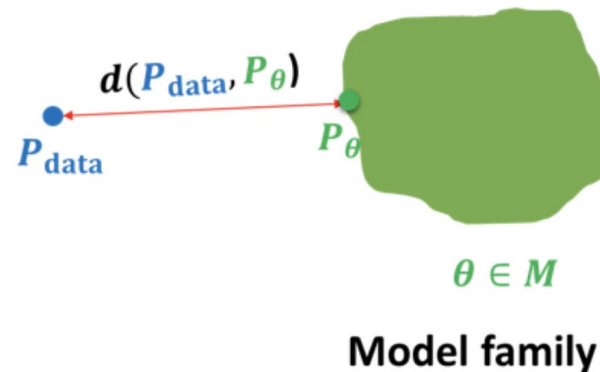
- Background and Motivation
  - Single-cell RNA Sequencing
  - Single-cell Computational Tasks
  - Overview of DeepSEM
- Methods
  - Variational Autoencoder (VAE)
  - Structural Equation Model (SEM)
  - DeepSEM: Framework and Implementation
- Model Performance
  - GRN Inference
  - Single-cell Clustering and Embedding
  - scRNA-seq Data Simulation

# Variational Autoencoder (VAE)

- Generative Models



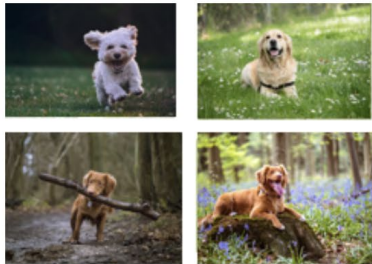
$$\begin{aligned} \mathbf{x}_i &\sim P_{\text{data}} \\ i &= 1, 2, \dots, n \end{aligned}$$



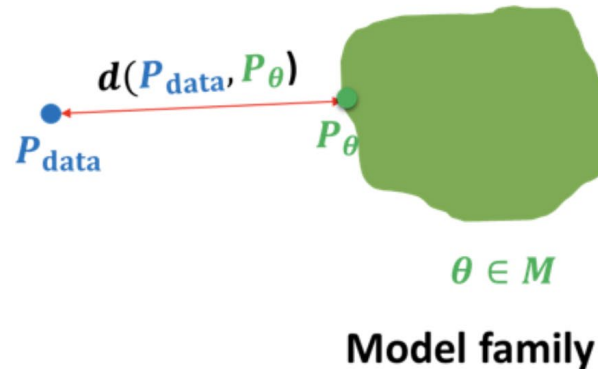
- Goal: Learn a probability distribution  $p(x)$  over  $x$  such that
  - 1. Generation:** If we sample  $x_{\text{new}} \sim p(x)$ ,  $x_{\text{new}}$  should look like a real image
  - 2. Density Estimation:**  $p(x)$  should be high if  $x$  looks like a real image, and low otherwise (anomaly detection)
  - 3. Unsupervised Representation Learning:** We should be able to learn high level features of these images
- Key questions:** how to construct and learn  $p(x)$ ?

# Variational Autoencoder (VAE)

- Learning as Density Estimation
  - Goal: construct  $p_\theta$  as *close* as possible to  $p_{\text{data}}$  (where we assume the dataset  $\mathcal{D}$  come from)



$\mathbf{x}_i \sim P_{\text{data}}$   
 $i = 1, 2, \dots, n$



# Variational Autoencoder (VAE)

---

- Learning as Density Estimation
  - Goal: construct  $p_\theta$  as *close* as possible to  $p_{\text{data}}$  (where we assume the dataset  $\mathcal{D}$  come from)
  - Measurement of closeness: **KL-divergence**

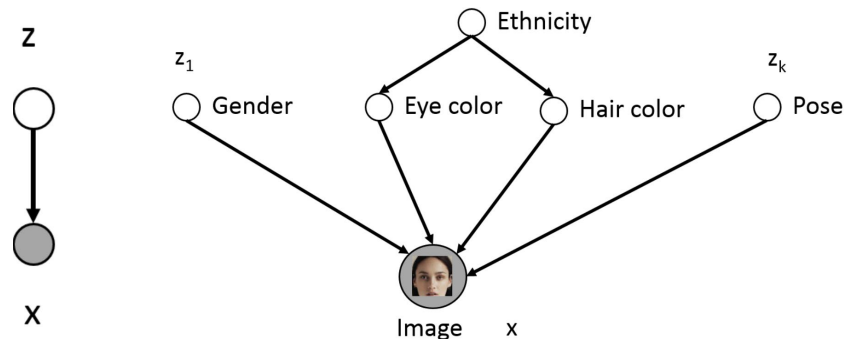
$$\begin{aligned}\text{KL}(p_{\text{data}} \| p_\theta) &= \mathbb{E}_{x \sim p_{\text{data}}} \left( \log \frac{p_{\text{data}}(x)}{p_\theta(x)} \right) \\ &= \sum_x p_{\text{data}}(x) \log \frac{p_{\text{data}}(x)}{p_\theta(x)}\end{aligned}$$

- Minimizing KL-divergence is equivalent to maximizing the **expected log-likelihood**

$$\arg \min_{p_\theta} \text{KL}(p_{\text{data}} \| p_\theta) = \arg \max_{p_\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_\theta(x)$$

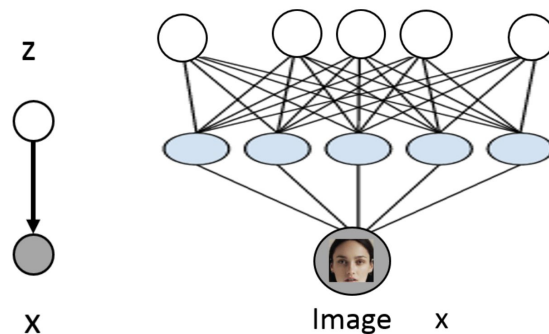
# Variational Autoencoder (VAE)

- Latent Variable Models



- Latent variables  $z$  corresponds to high-level features,  $p(x|z)$  could be much simpler than  $p(x)$  if  $z$  chosen properly

- Deep Latent Variable Models



- Learn  $p(x|z)$  via deep neural networks

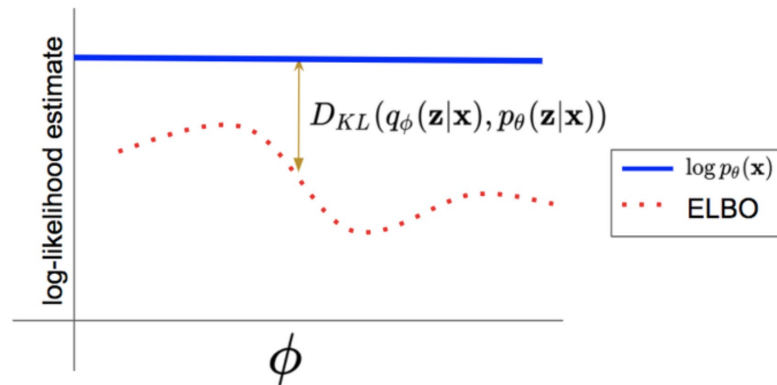


# Variational Autoencoder (VAE)

- Variational Inference for Latent Variable Models
  - Evidence Lower Bound (ELBO)

$$\begin{aligned}\log p_{\theta}(x) &\geq \mathbb{E}_{z \sim q(z)} \log \frac{p_{\theta}(x, z)}{q(z)} \\ &= \mathbb{E}_{z \sim q(z)} \log p_{\theta}(x, z) - \mathbb{E}_{z \sim q(z)} \log q(z) \\ &= \mathbb{E}_{z \sim q(z)} \log p_{\theta}(x, z) + H(q)\end{aligned}$$

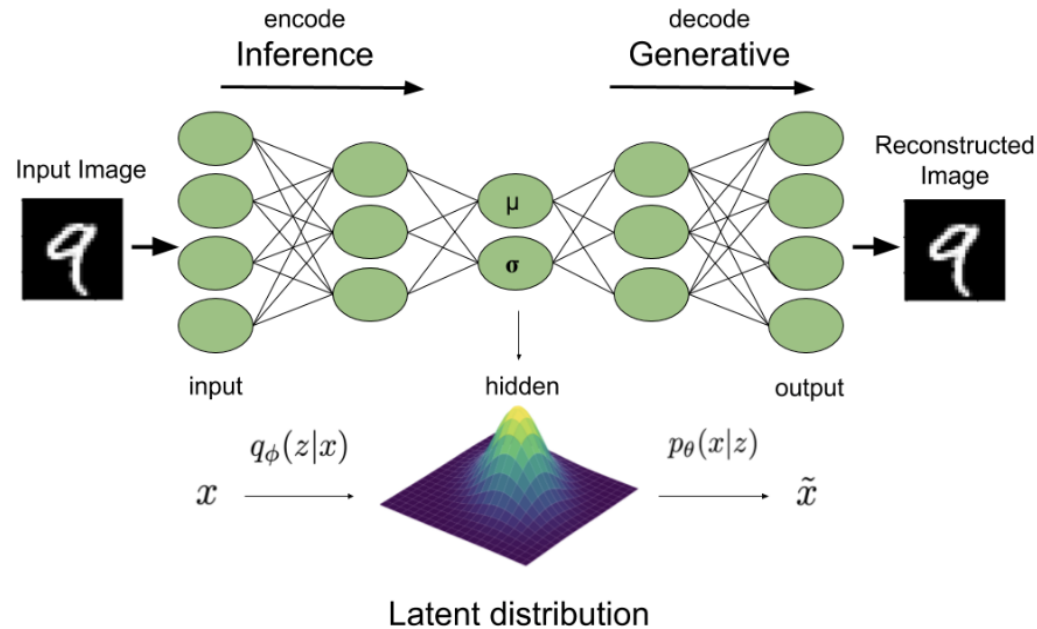
- Maximize the **expected log-likelihood**  $\log p_{\theta}(x)$  by maximizing **ELBO**



$$\begin{aligned}\log p_{\theta}(x) &\geq \int_z q_{\phi}(z|x) \log \frac{p_{\theta}(x, z)}{q_{\phi}(z|x)} = \mathcal{L}(x; \theta, \phi) \\ &= \mathcal{L}(x; \theta, \phi) + \text{KL}(q_{\phi}(z|x) \| p(z|x; \theta))\end{aligned}$$

# Variational Autoencoder (VAE)

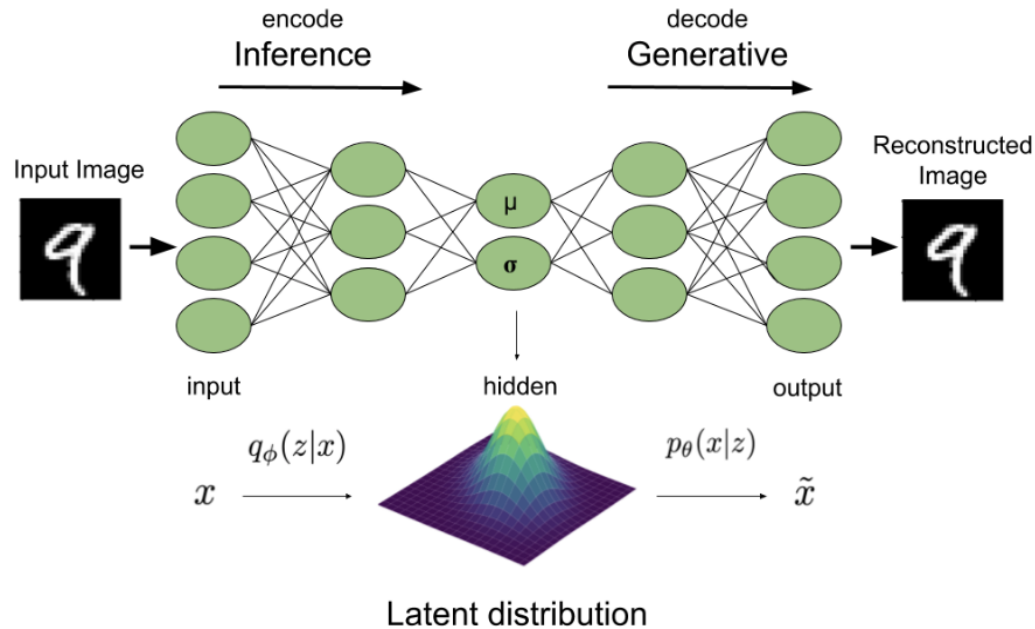
- Variational Autoencoders



$$\begin{aligned}\mathcal{L}(x; \theta, \phi) &= \mathbb{E}_{q_\phi(z|x)} (\log p_\theta(x, z) - \log q_\phi(z|x)) \\ &= \mathbb{E}_{q_\phi(z|x)} (\log p_\theta(x|z) + \log p(z) - \log q_\phi(z|x)) \\ &= \mathbb{E}_{q_\phi(z|x)} \log p(x|z; \theta) - \text{KL}(q_\phi(z|x) || p(z))\end{aligned}$$

# Variational Autoencoder (VAE)

- Intuition of Variational Autoencoders

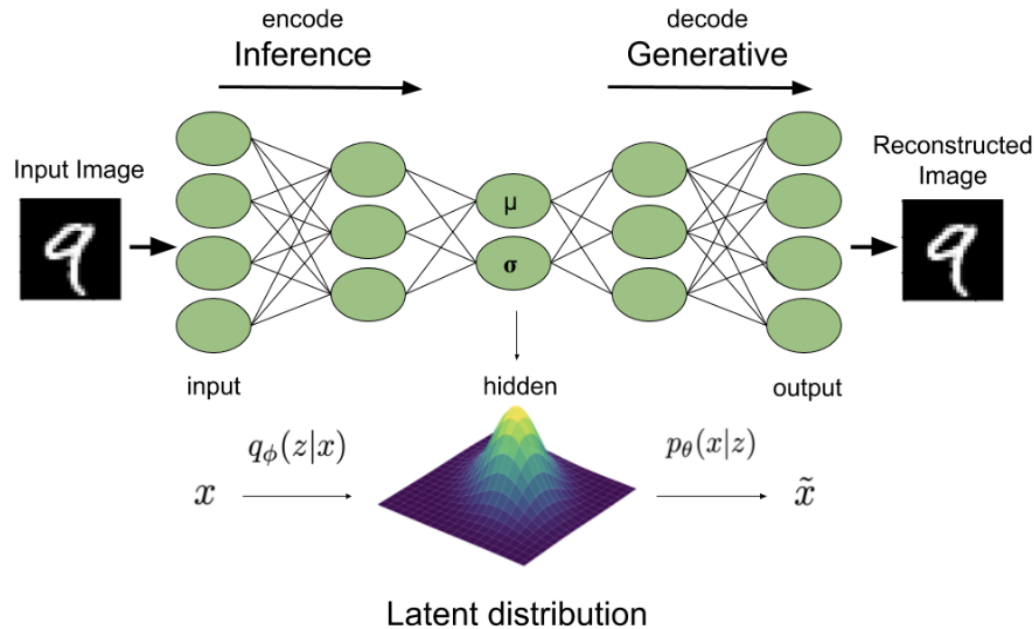


- Take a data point  $x^i$ 
  - Encoder** maps  $x^i$  to  $\hat{z}$  by sampling from  $q_\phi(z|x^i)$
  - Decoder** reconstruct  $\hat{x}$  by sampling from  $p_\theta(x|\hat{z}; \theta)$



# Variational Autoencoder (VAE)

- Intuition of Variational Autoencoders

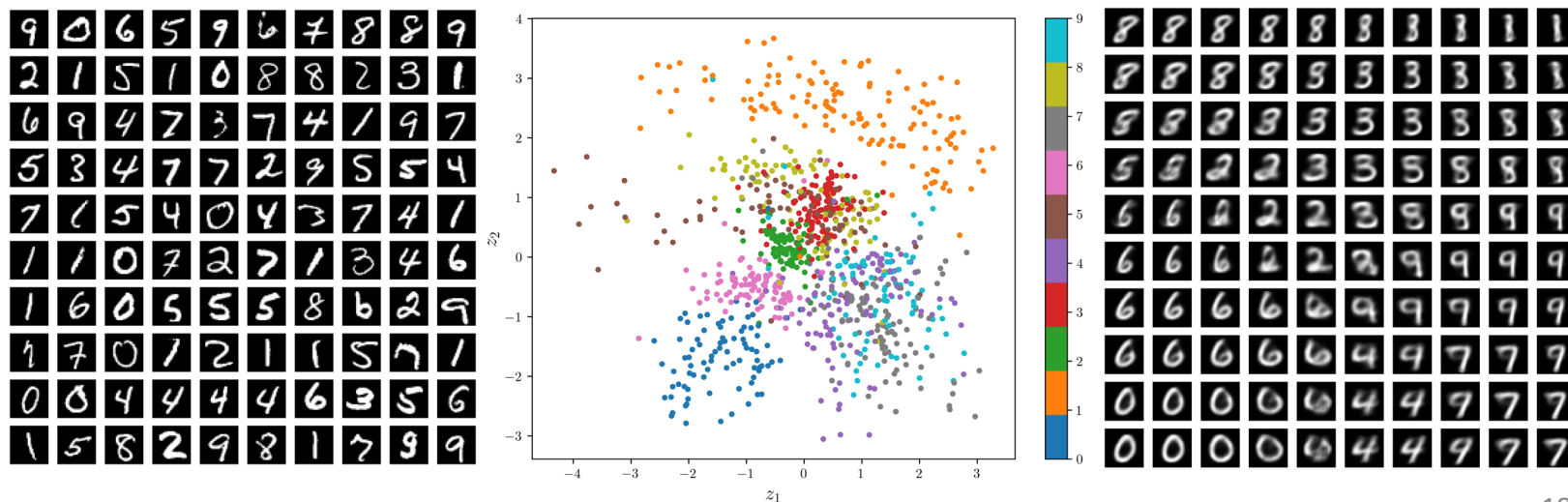
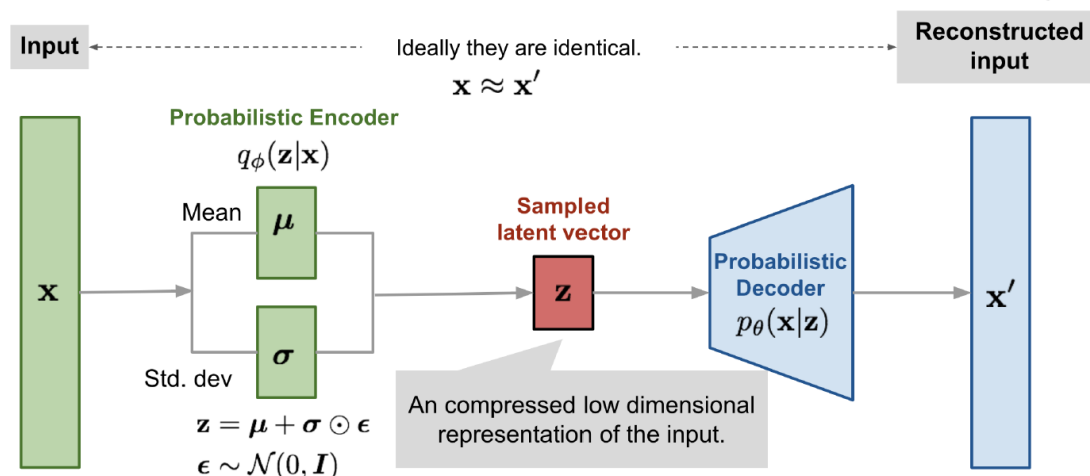


$$\mathcal{L}(x; \theta, \phi) = \mathbb{E}_{q_\phi(z|x)} \log p(x|z; \theta) - \text{KL}(q_\phi(z|x) || p(z))$$

- First term encourages  $\hat{x} \approx x^i$  ( $x^i$  likely under  $p(x|\hat{z}; \theta)$ )
- Second term encourages  $z$  to be likely under the prior  $p(z)$

# Variational Autoencoder (VAE)

- Example: Inference and Generation of MNIST Dataset using VAE





# Structural Equation Model (SEM)

---

- Structural Equation Model
  - Gaussian linear latent variable model
    - **Linearity** of the relations between the variables
    - **Normality**, Gaussian distribution for the residuals of the latent variables and the residuals of the observed variables
  - Linear **measurement** model: relate the latent variables  $\mathbf{X}$  to the observed variables  $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \boldsymbol{\varepsilon}$$

- Linear **structural** model: relate the latent variables  $\mathbf{X}$  to each other (self-regression problem)

$$\mathbf{X} = \mathbf{W}^T \mathbf{X} + \mathbf{Z}$$

which gives

$$\mathbf{X} = (\mathbf{I} - \mathbf{W}^T)^{-1} \mathbf{Z}$$

$$\mathbf{Z} = (\mathbf{I} - \mathbf{W}^T) \mathbf{X}$$

- $\boldsymbol{\varepsilon}, \mathbf{Z}$  are **noise matrices** of multivariate Gaussian distribution



# Structural Equation Model (SEM)

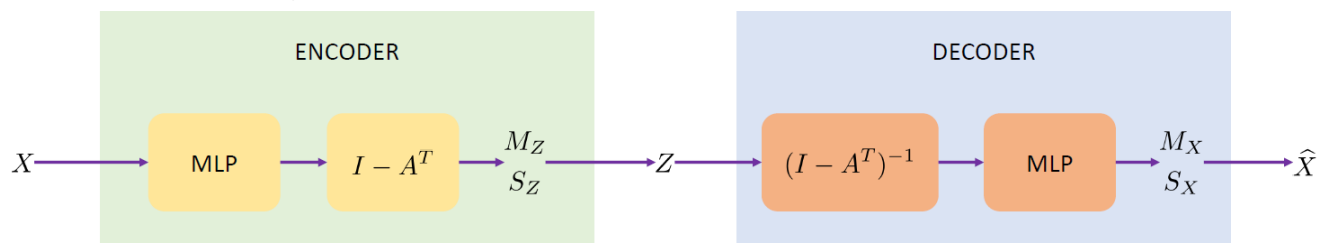
---

- **Deep** Structural Equation Model
  - Gaussian linear latent variable model
    - **Linearity** of the relations between the variables
    - **Normality**, Gaussian distribution for the residuals of the latent variables and the residuals of the observed variables
  - **Deep measurement** model: relate the latent variables  $\mathbf{X}$  to the observed variables  $\mathbf{Y}$  **with deep neural networks (e.g. MLP)**
  - **Deep structural** model: relate the latent variables  $\mathbf{X}$  to each other (self-regression problem)  
**multilayer neural network**  $f_1, f_2$

$$\begin{aligned}\mathbf{X} &= f_1((\mathbf{I} - \mathbf{W}^T)^{-1} \mathbf{Z}) \\ \mathbf{Z} &= (\mathbf{I} - \mathbf{W}^T) f_2(\mathbf{X})\end{aligned}$$

# Structural Equation Model (SEM)

- **Deep** Structural Equation Model → **VAE**

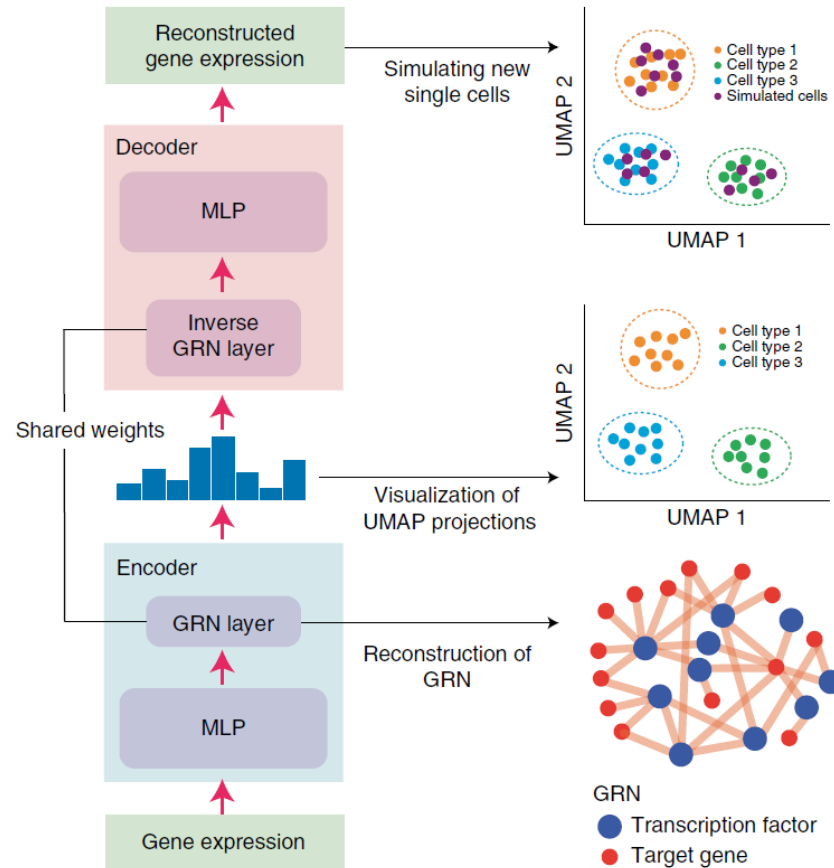


- **Deep measurement** model: relate the latent variables  $\mathbf{X}$  to the observed variables  $\mathbf{Y}$  with deep neural networks (e.g. MLP)
- **Deep structural** model: relate the latent variables  $\mathbf{X}$  to each other (self-regression problem)  
multilayer neural network  $f_1, f_2$  as Decoder and Encoder

$$\begin{aligned}\mathbf{X} &= f_1((\mathbf{I} - \mathbf{W}^T)^{-1} \mathbf{Z}) \\ \mathbf{Z} &= (\mathbf{I} - \mathbf{W}^T) f_2(\mathbf{X})\end{aligned}$$

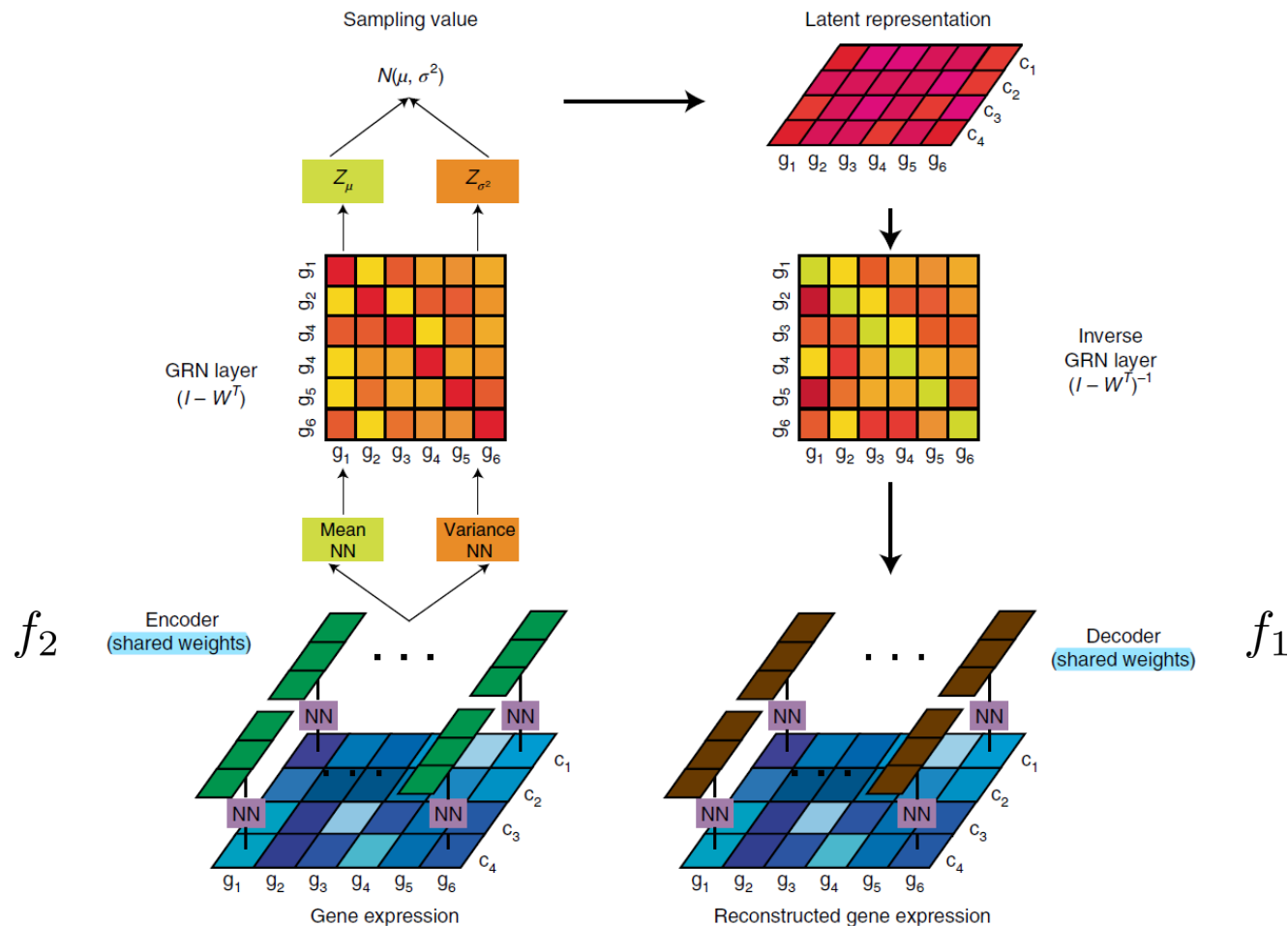
# DeepSEM: Framework and Implementation

- Framework of DeepSEM



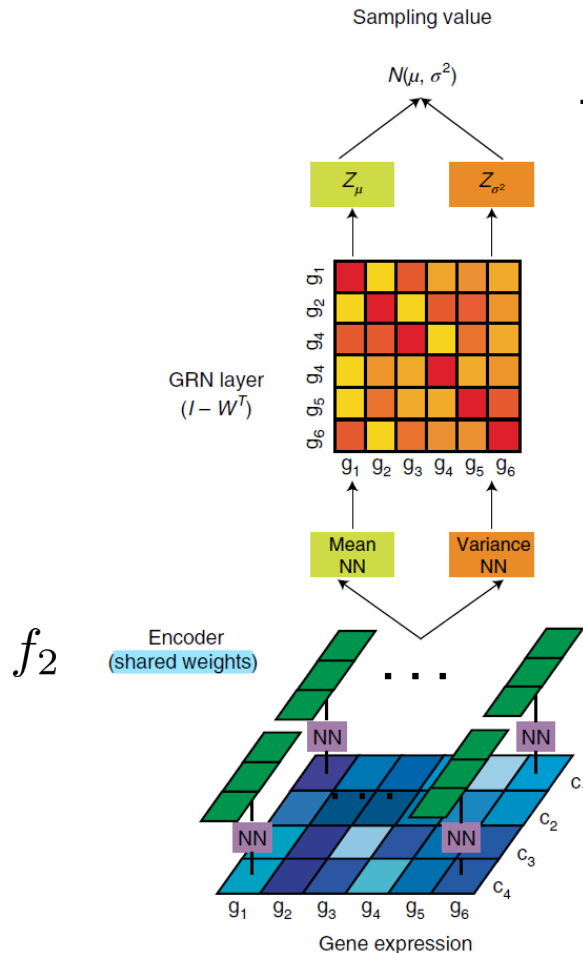
# DeepSEM: Framework and Implementation

- Framework of DeepSEM



# DeepSEM: Framework and Implementation

- Framework of DeepSEM



$$\mathbf{Z} = \text{Reparameter} \left( (\mathbf{I} - \mathbf{W}^T) \mathbf{H}_X \right)$$

$$\mathbf{Z} \in \mathbb{R}^{n \times m}$$

$$\mathbf{H}_X = [f_2(\mathbf{x}_0); f_2(\mathbf{x}_1); \dots; f_2(\mathbf{x}_m)]$$

$$f_2(\mathbf{x}_i) = \tanh \left( \tanh \left( \tanh (\mathbf{x}_i \mathbf{W}_4^T) \mathbf{W}_5^T \right) \mathbf{W}_6^T \right)$$

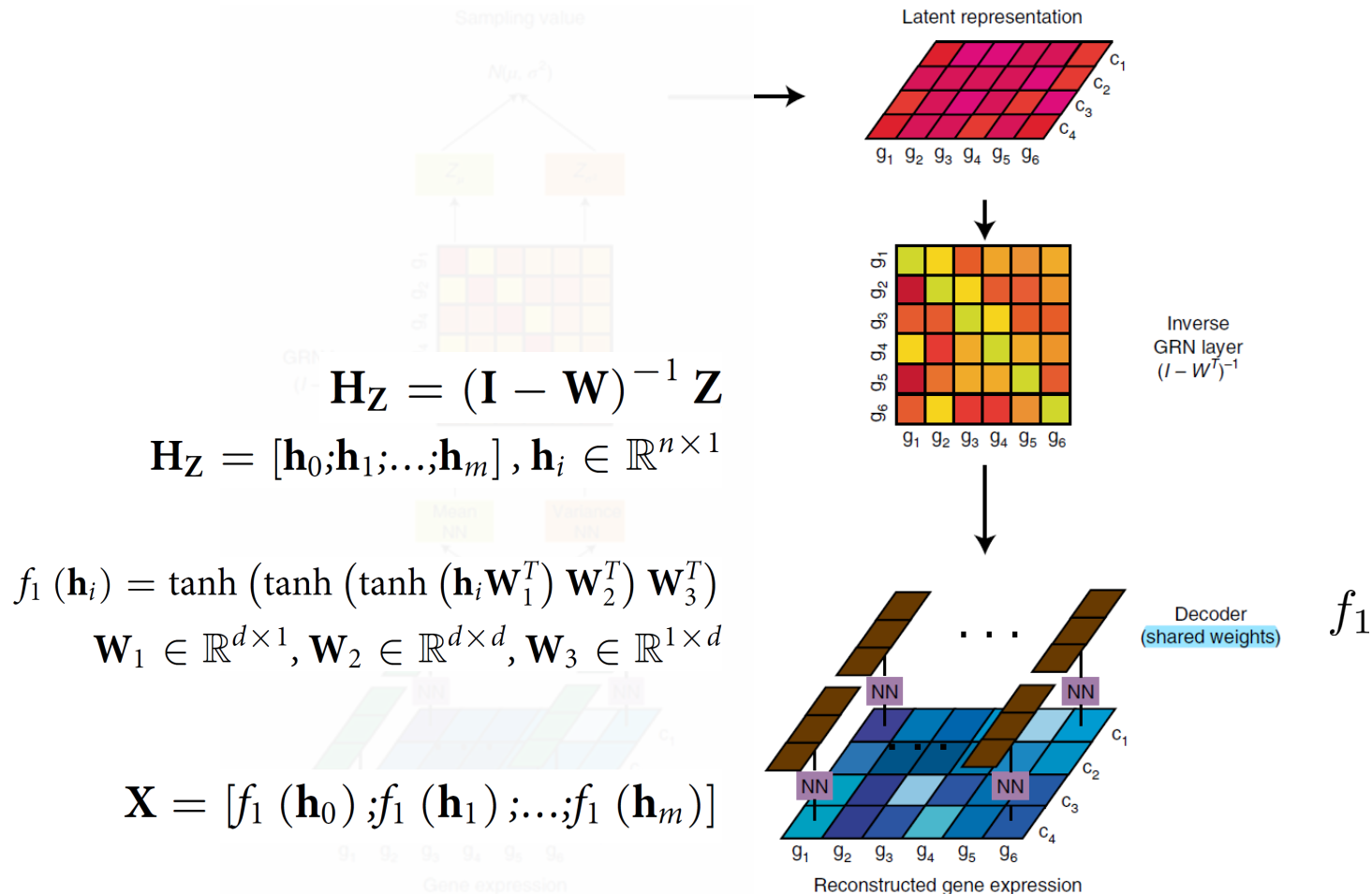
$$\mathbf{W}_4 \in \mathbb{R}^{d \times 1}, \mathbf{W}_5 \in \mathbb{R}^{d \times d}, \mathbf{W}_6 \in \mathbb{R}^{2 \times d}$$

$$\mathbf{X} = [\mathbf{x}_0; \mathbf{x}_1; \dots; \mathbf{x}_m], \mathbf{x}_i \in \mathbb{R}^{n \times 1}$$



# DeepSEM: Framework and Implementation

- Framework of DeepSEM





# DeepSEM: Framework and Implementation

- Implementation Details

- **$\beta$ -VAE**: a simple variant of vanilla VAE

$$\mathcal{L} = -E_{q(\mathbf{X})}[\log p(\mathbf{X}|\mathbf{Z})] + \beta \text{KL}(q(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z})) + \alpha \|\mathbf{W}\|_1$$

$$\mathcal{L}(x; \theta, \phi) = \mathbb{E}_{q_\phi(z|x)} \log p(x|z; \theta) - \text{KL}(q_\phi(z|x)||p(z))$$

- First term encourages  $\hat{x} \approx x^i$  ( $x^i$  likely under  $p(x|\hat{z}; \theta)$ )
  - Second term encourages  $z$  to be likely under the prior  $p(z)$
- Input: log-transformed scRNA-seq expression data (z-normalized)
  - Model training: RMSprop to avoid unstable behavior of Adam; optimize MLPs and  $\mathbf{W}$  alternately
  - Initialization: matrix diagonal of  $\mathbf{W}$  are set as zeros (no self-interactions)

Shu, H., Zhou, J., Lian, Q., Li, H., Zhao, D., Zeng, J., & Ma, J. (2021). Modeling gene regulatory networks using neural network architectures. *Nature Computational Science*, 1(7), 491-501.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., ... & Lerchner, A. (2017, April). beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*.

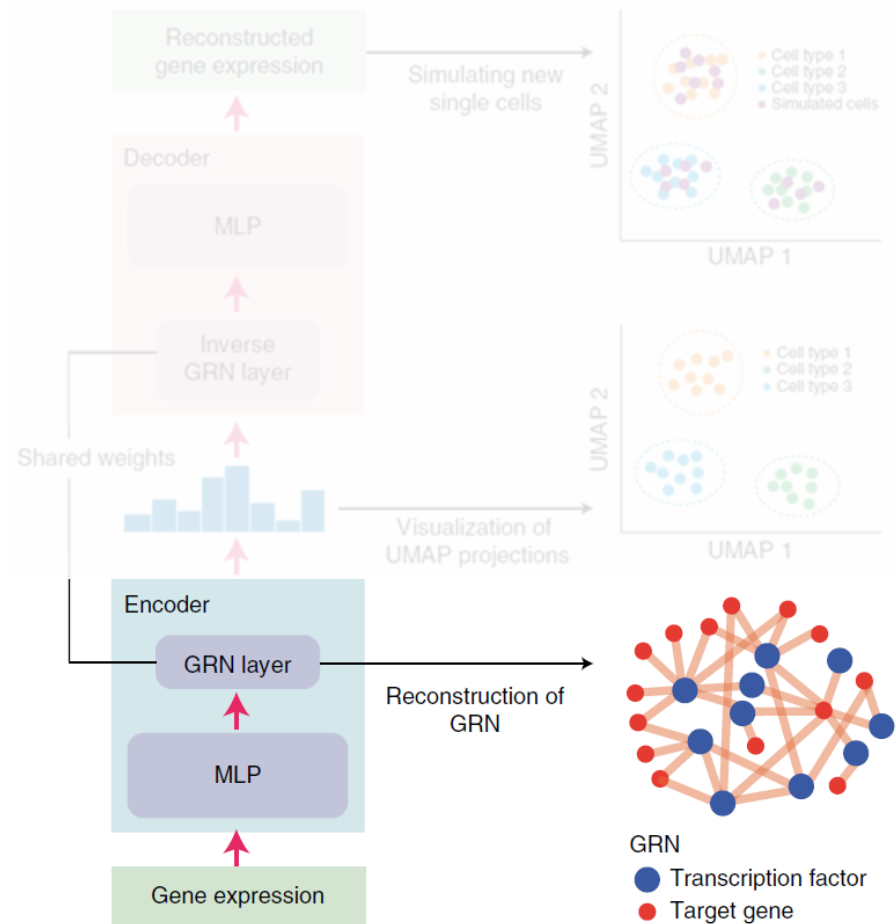
# Outline

---

- Background and Motivation
  - Single-cell RNA Sequencing
  - Single-cell Computational Tasks
  - Overview of DeepSEM
- Methods
  - Variational Autoencoder (VAE)
  - Structural Equation Model (SEM)
  - DeepSEM: Framework and Implementation
- Model Performance
  - GRN Inference
  - Single-cell Clustering and Embedding
  - scRNA-seq Data Simulation

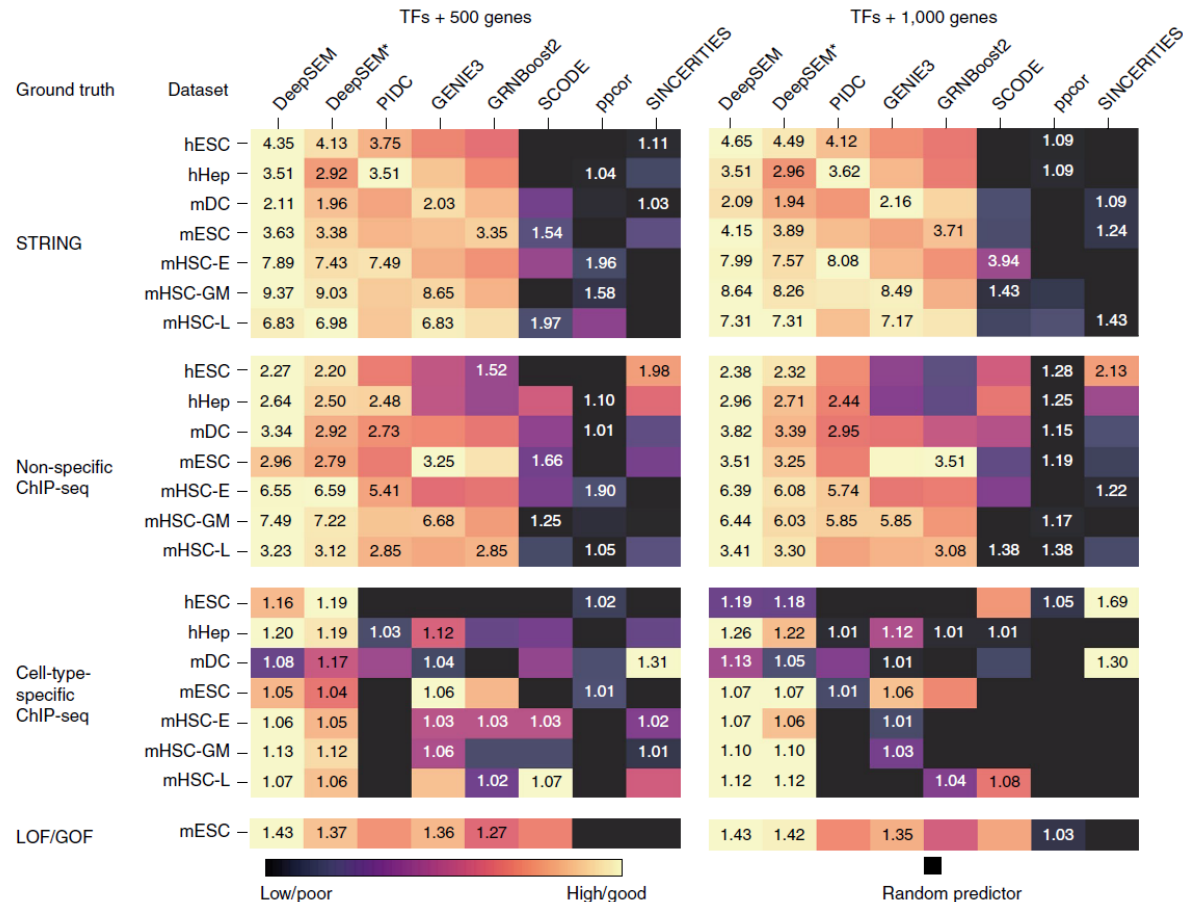
# GRN Inference

- Performance of GRN Inference



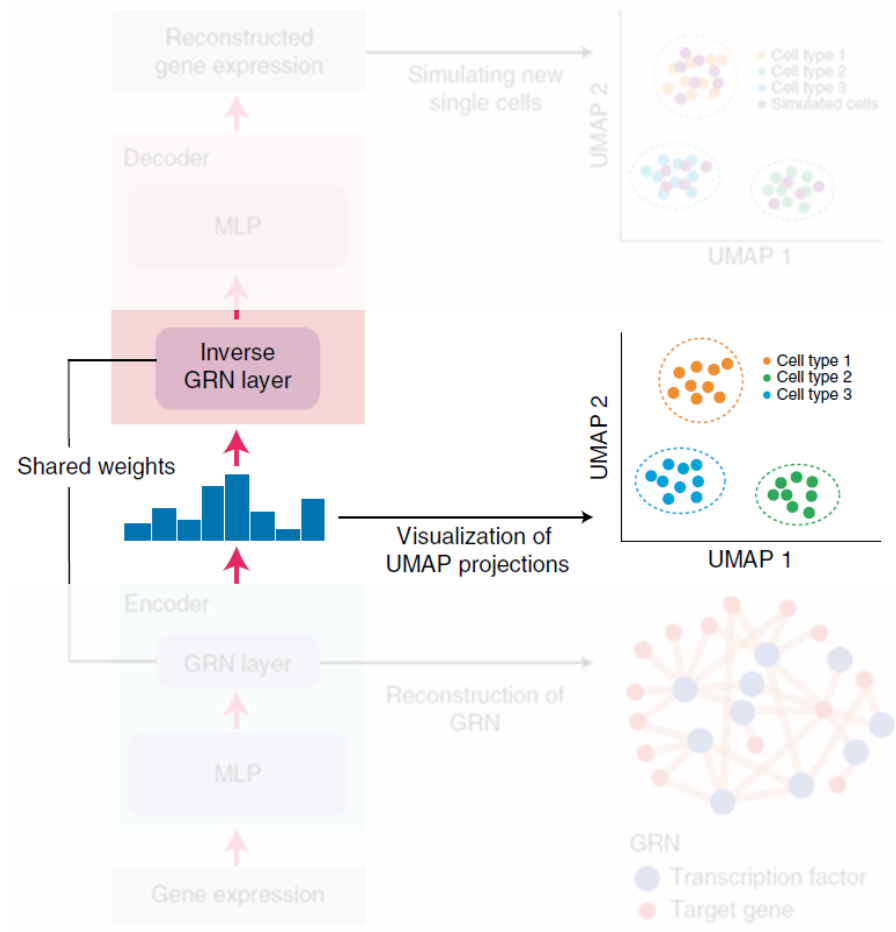
# GRN Inference

- Performance of GRN Inference
  - Metric: EPR (Early precision ratio)



# Single-cell Clustering and Embedding

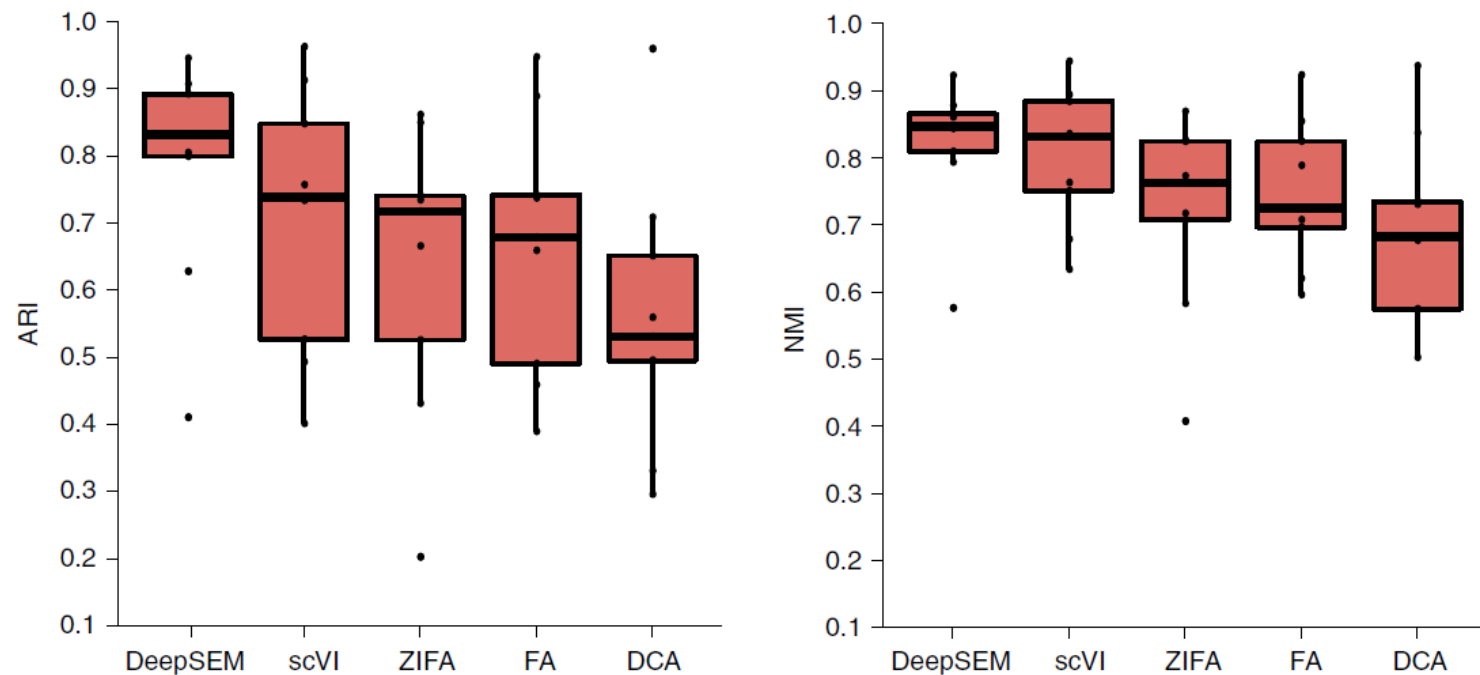
- Performance of Single-cell clustering and embedding





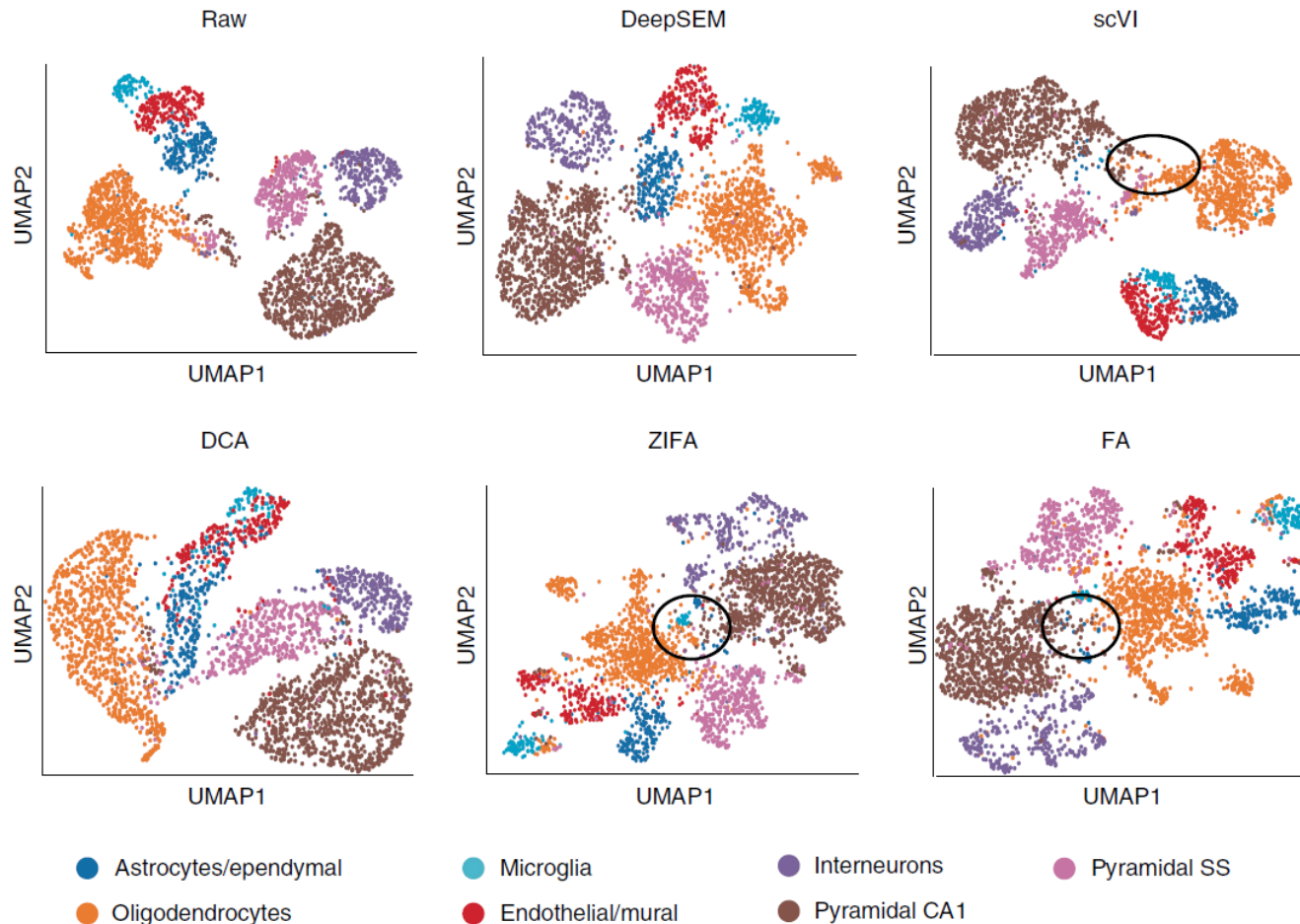
# Single-cell Clustering and Embedding

- Performance of Single-cell clustering and embedding



# Single-cell Clustering and Embedding

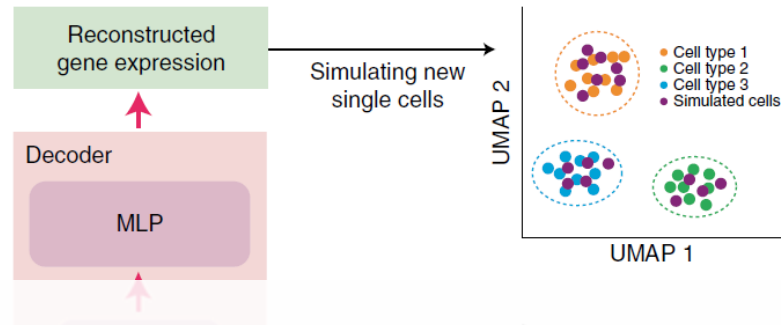
- Performance of Single-cell clustering and embedding





# scRNA-seq Data Simulation

- Simulation Performance



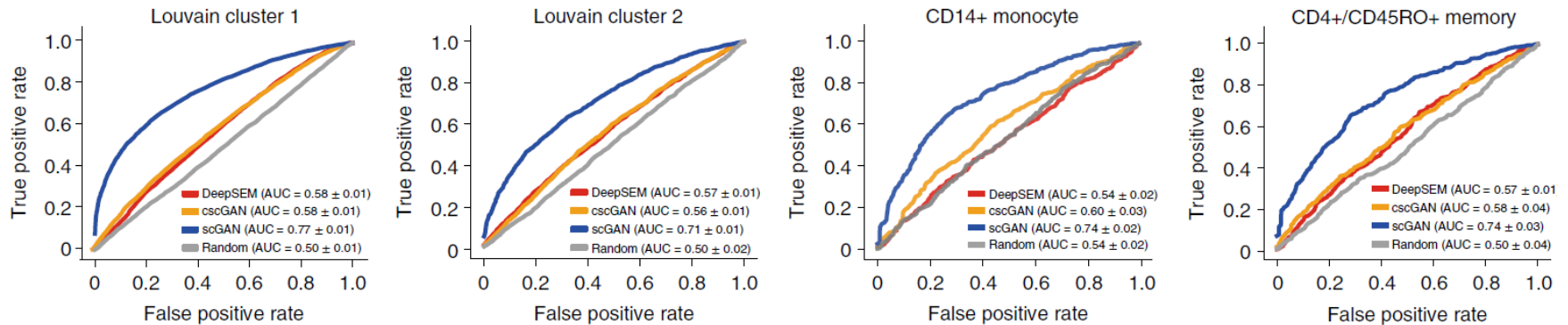
- Perturb the hidden vector  $\mathbf{Z}$  with **Gaussian noise**

$$\hat{\mathbf{Z}} = \mu + n\sigma$$

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

# scRNA-seq Data Simulation

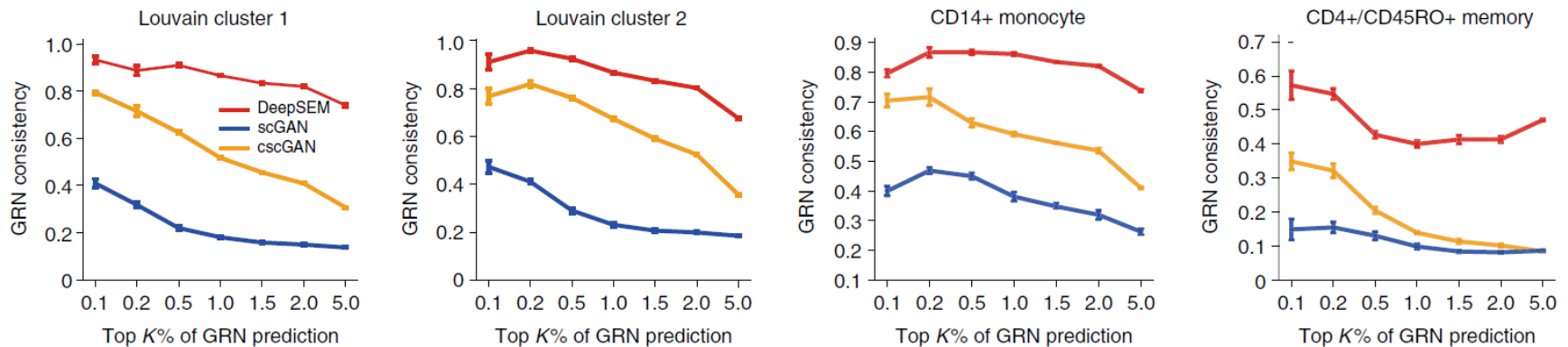
- Simulation Performance





# scRNA-seq Data Simulation

- Simulation Performance

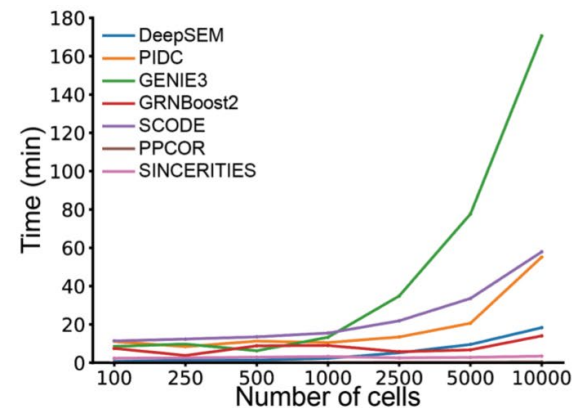
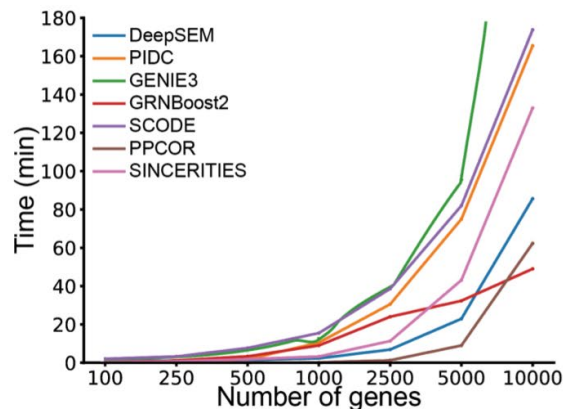


$$\text{GRN consistency} = \frac{\text{Number of overlap edges in top } N \text{ predicted edges between real and simulated cells}}{N}$$

$$N = K\% \times \text{number of predicted GRN in real cells}$$

# Discussion: Inspirations and Critics

- Inspirations
  - Utilize deep neural networks for enhanced model capacity
  - One model for various single-cell computational tasks
  - Combination of deep learning and causal inference gives explainability
- Critics
  - The model need to be trained from scratch for each set of scRNA-seq data, and the training procedure is (notoriously) unstable, though the training speed is comparably acceptable



- TFs are omitted in GRN, thus he predicted regulatory interactions are suspicious

# Thank you

---

**Yuzhe Wang<sup>1</sup>, Shan Liu<sup>2</sup>**

<sup>1</sup> Peking University

<sup>2</sup> Beijing Computational Science Research Center