Latent Representation through Identifiable Variational Autoencoder without Side Information

Ping-Han Hsieh

2021-09-30

Outline

- Dimension reduction
- Disentangled representation
- Variational autoencoder
- Identifiability of generative model
- More on Deep Generative Models
- Discussion and Future Works

Dimension Reduction

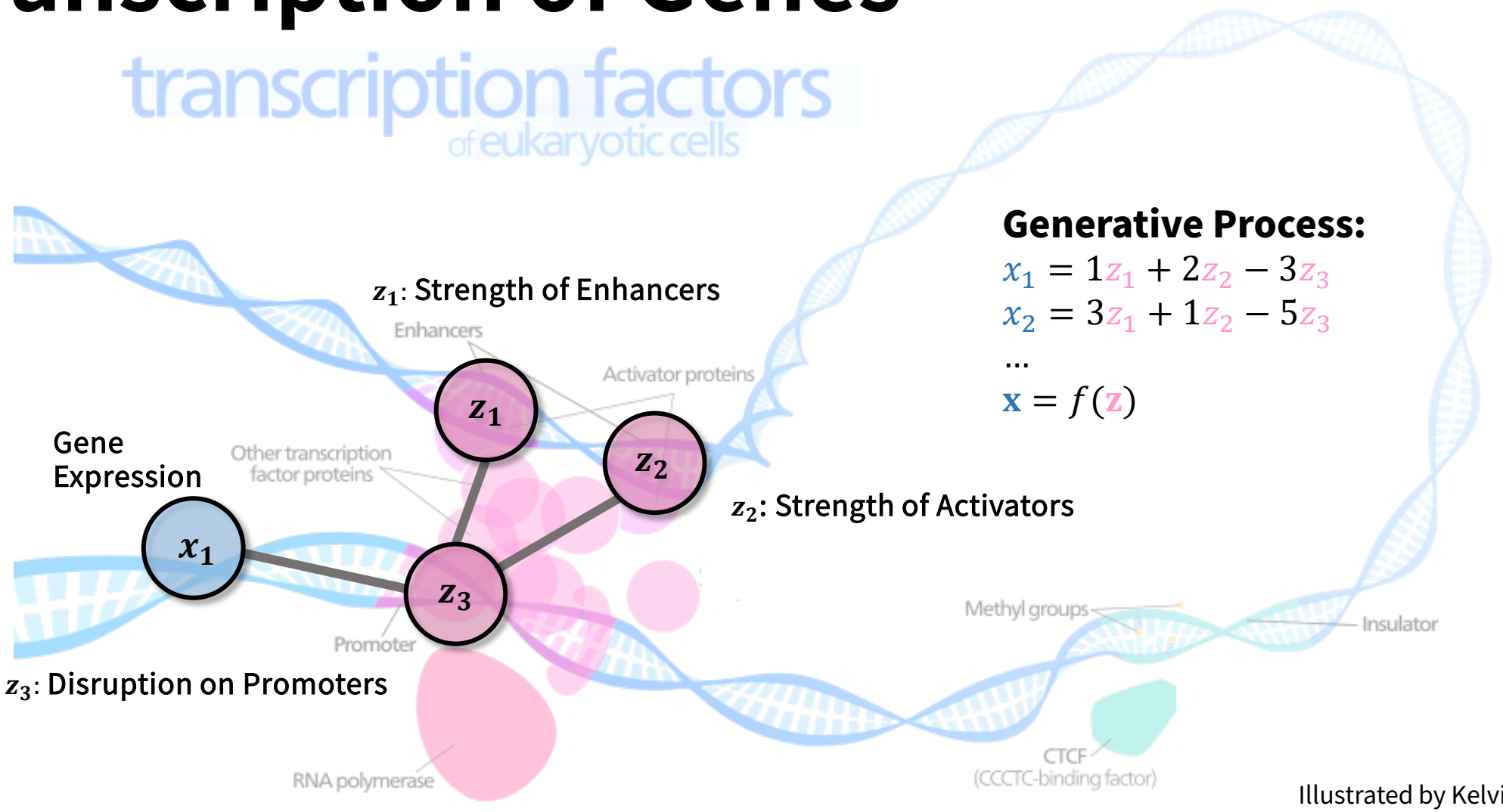
Dimension Reduction (1)

- From a computational point of view
  - A process to embed high-dimensional data into lower-dimensional latent representation.

- Applications
  - Data compression
  - Exploratory data analysis
  - Facilitate downstream analysis (avoid curse of dimensionality)
  - (Latent) Feature extraction
  - Manifold learning (distance between samples)
  - **Generative process**
    - Suppose our **observation data** (e.g. RNA-Seq) is the result of a **biological generative process** (e.g. transcription)
    - Dimension reduction help us **reconstruct the generative process** and understand what can be the important **latent factors** take part in this process (e.g. transcription factor)

Generative Process Example

Transcription of Genes



z_1: Strength of Enhancers

Gene Expression

z_2: Strength of Activators

z_3: Disruption on Promoters

Generative Process:

$x_1 = 1z_1 + 2z_2 - 3z_3$

$x_2 = 3z_1 + 1z_2 - 5z_3$

...

$\mathbf{x} = f(\mathbf{z})$

Illustrated by Kelvin Ma

Dimension Reduction (2)

- The choice of method to perform dimension reduction:
  - Depend on the desired property of the latent factors:
    1. Preserve **distance between samples** (e.g. similarity in gene expression)
       - IsoMap, MDS, tSNE, UMAP
    2. Extract **disentangled** representation (interpretation of the generative model):
       - Enforce **sparsity**
       - Extract **independent** factors
         - Independent Component Analysis (ICA), β-Variational Autoencoder
       - Extract **orthogonal** representation
         - Principal Component Analysis (PCA), Factor Analysis (FA)
    3. Reconstruct **generative process**
       - Linear generative process
         - Probabilistic PCA
       - Non-linear generative process
         - Variational Autoencoder, Non-linear ICA

Disentangled Representation

Disentangling Representation

Chen et al., NeurIPS 2018

Generative Process:

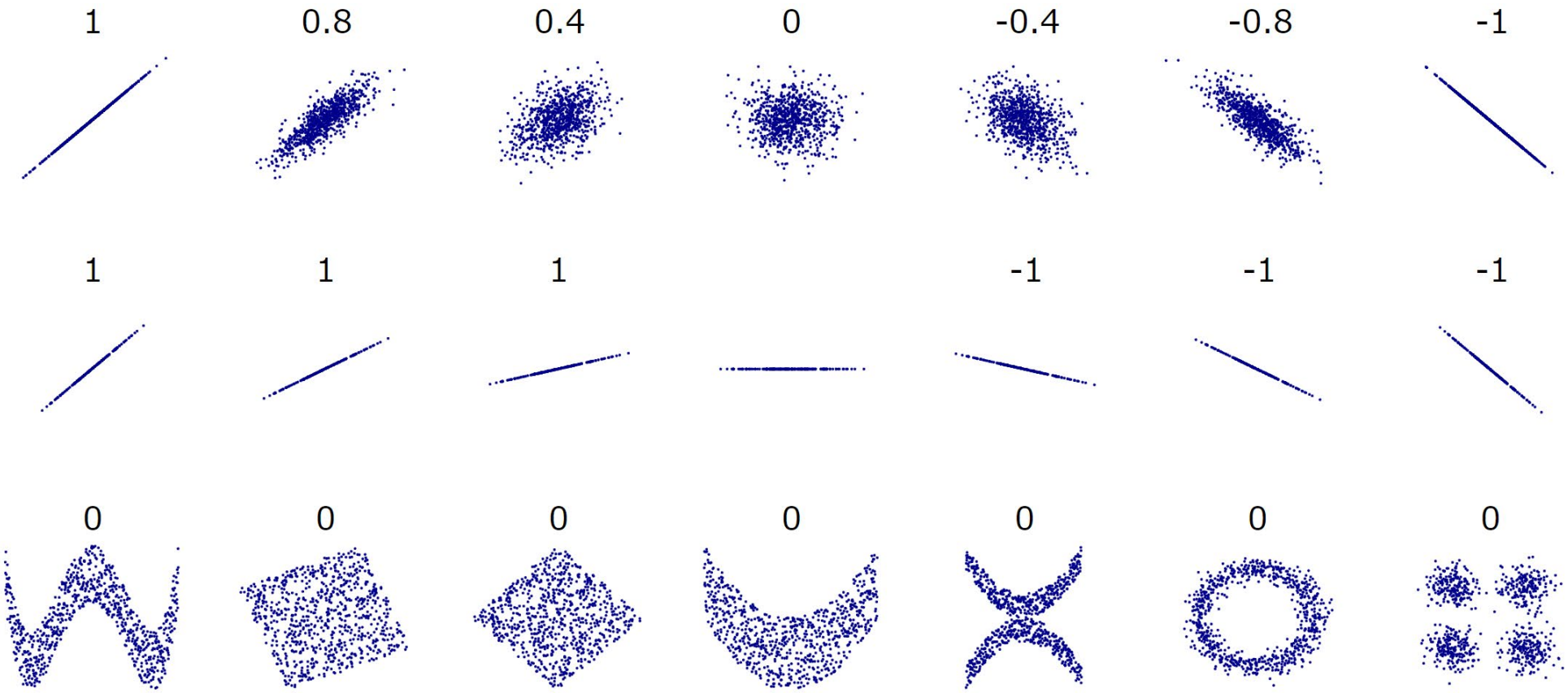
$x_1 = f_1(z_1) + f_2(z_2) + f_3(z_3) + f_4(z_4) + f_5(z_5) + f_6(z_6)$

$\mathbf{x} = f(\mathbf{z})$

- By controlling the latent factor, we can control the generated data (e.g. image) according to the features that we want.
- Disentangling means that **one factor only control one (few) aspect of the generative process** (e.g. changing z_2 will not change the hair color).
- The more disentangling the factors are, the easier it is for us to control (understand) the generative process.

Different Assumptions for Disentangling Factors

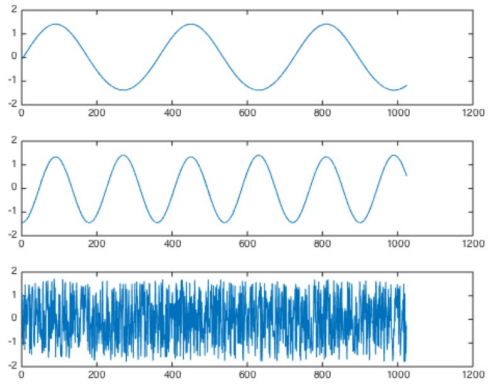
Orthogonal and Independent



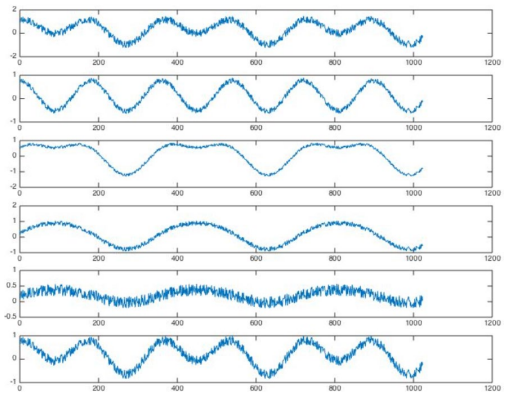
Wikipedia “Correlation”

Linear Independent Component Analysis

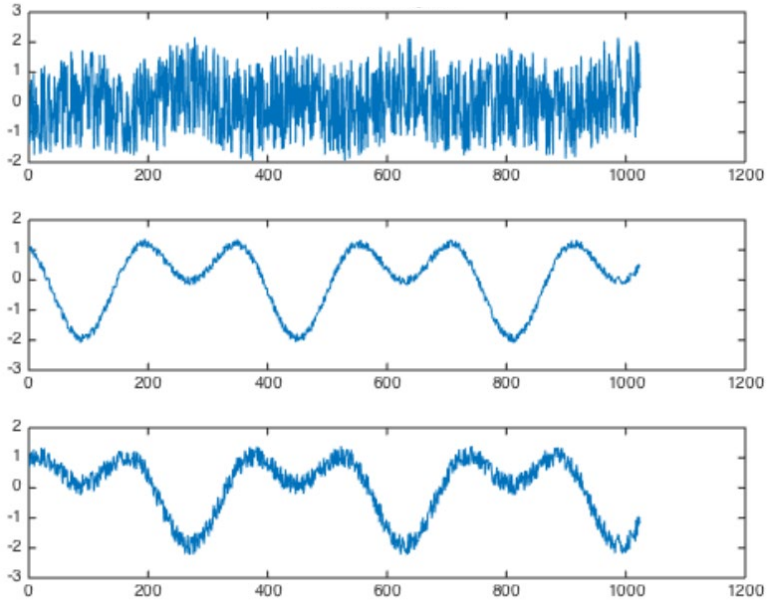
True Source



Observation



Principal Component Analysis

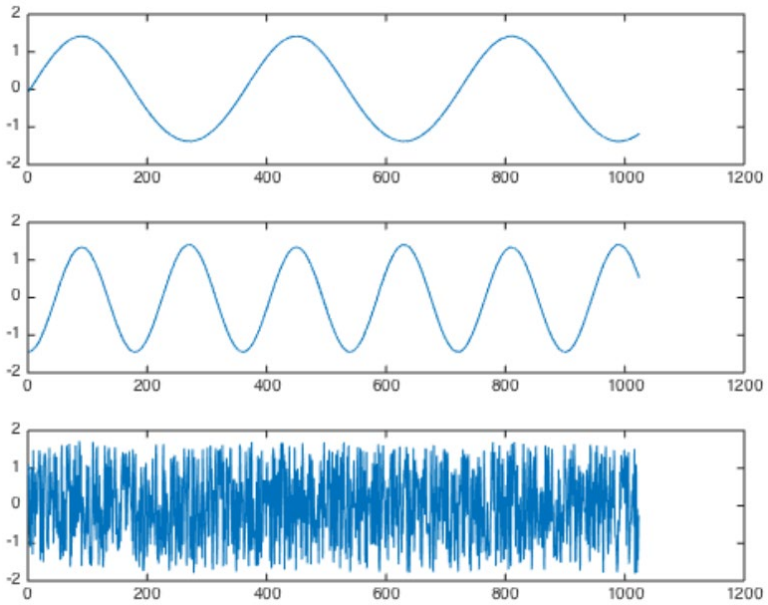


$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{-1}$$

$$\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{I} \quad \mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I}$$

find the largest projected variance

Independent Component Analysis



$$\mathbf{X} = \mathbf{W}^{-1}\mathbf{S}$$

$$p(\mathbf{X}) = p(\mathbf{S})|det\frac{\partial W}{\partial s}| \qquad p(\mathbf{S}) = \prod_{i=1}^{k} p(\mathbf{s_i})$$

find the most non-gaussian factor

Variational Autoencoder

Maximize Likelihood Estimation (1)

- Linear Regression
  - We want to find a generative process, such that $\hat{y} = Xw + b$.
  - We can write the likelihood of this function as $p(\hat{y} \mid X, y)$ with parameters w, b
- The implicit assumption of linear regression
  - We assume the choice of all possible value for w is the same (**uniform prior for w**).
  - We assume our proposed generative distribution is a **Gaussian distribution** given that we observe X (if we use mean squared error as the loss function)
  - $p(\hat{y} \mid X, y)$ is a probability density of Gaussian distribution with a **fixed constant standard deviation** (that is not related to the input) and mean $Xw + b$.

Log-Probability Density Function

$$\log p(\hat{\mathbf{y}}|\mathbf{y}, \mathbf{X}) = -\frac{N}{2}\log(2\pi) - N\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y-\mu)^2$$

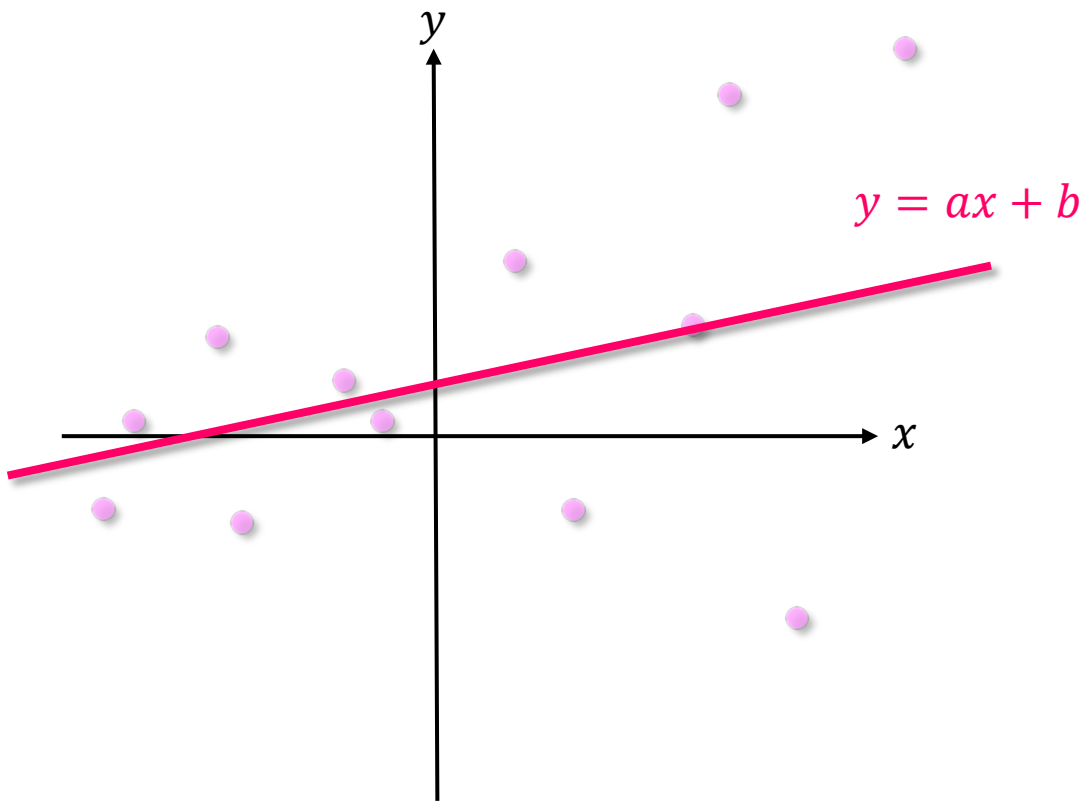
Mean Squared Error

$$\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2$$

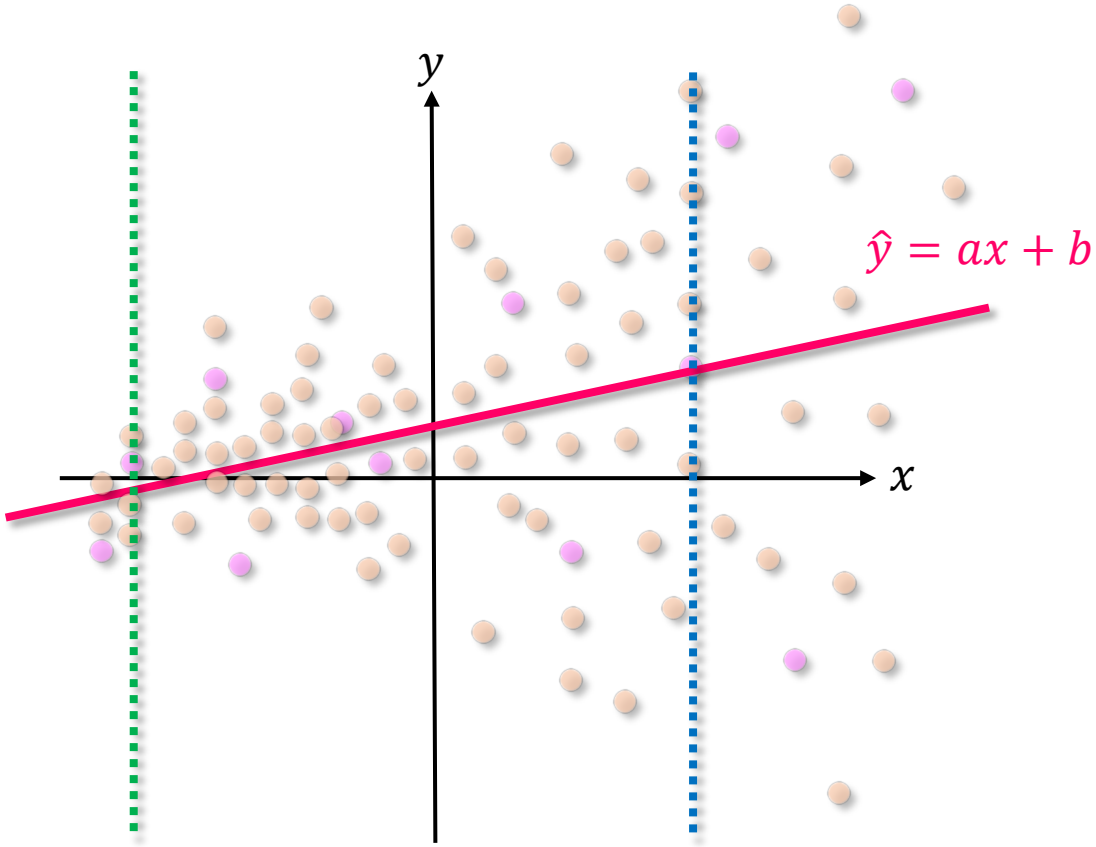
$$where \quad \mu = \hat{\mathbf{y}} = \mathbf{XW} + b$$ **Mean becomes a function of input**

- Given our observations X, y, and a proposed model $p_w(\cdot \mid X)$.
- When we fit the linear regression model with our observed data , we are actually finding a set of parameters w that maximize the likelihood.

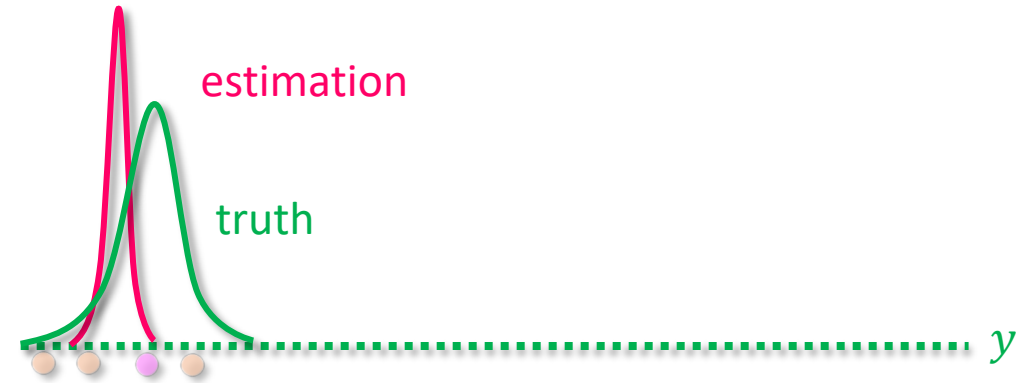
Maximize Likelihood Estimation (2)

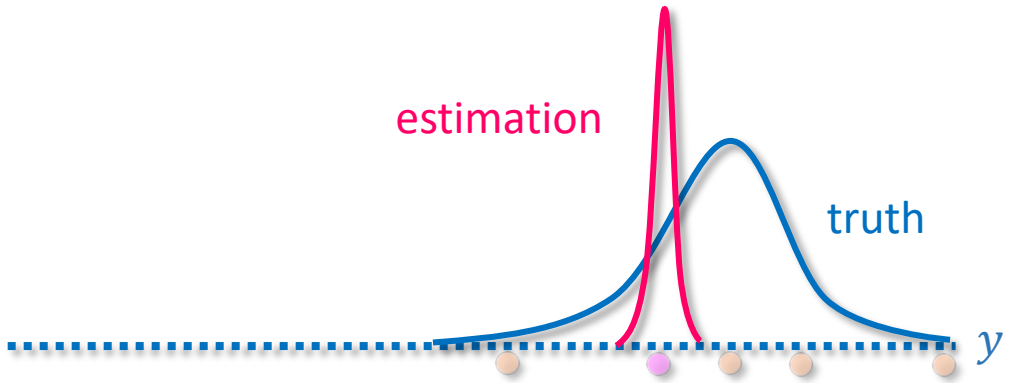


Maximize Likelihood Estimation (2)



observed data unobserved data





Maximize Likelihood Estimation (3)

- What does it mean for a prior:
  - The a priori assumption about the distribution of our parameters.
- For instance, if we use L1-regularization:

Log-Probability Density Function

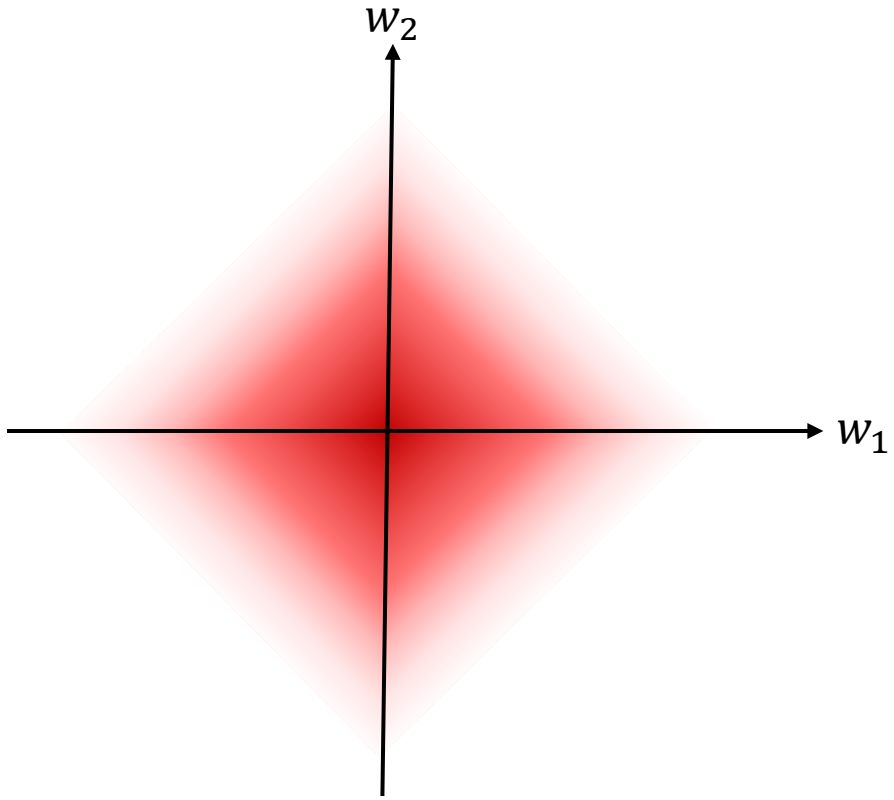
$$let\ \log p(\mathbf{W}) = -\beta|\mathbf{W}|$$

$$\log p(\hat{\mathbf{y}}|\mathbf{y}, \mathbf{X}) + \log p(\mathbf{W}) = -\frac{N}{2}\log(2\pi) - N\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y-\mu)^2 - \beta|\mathbf{W}|$$

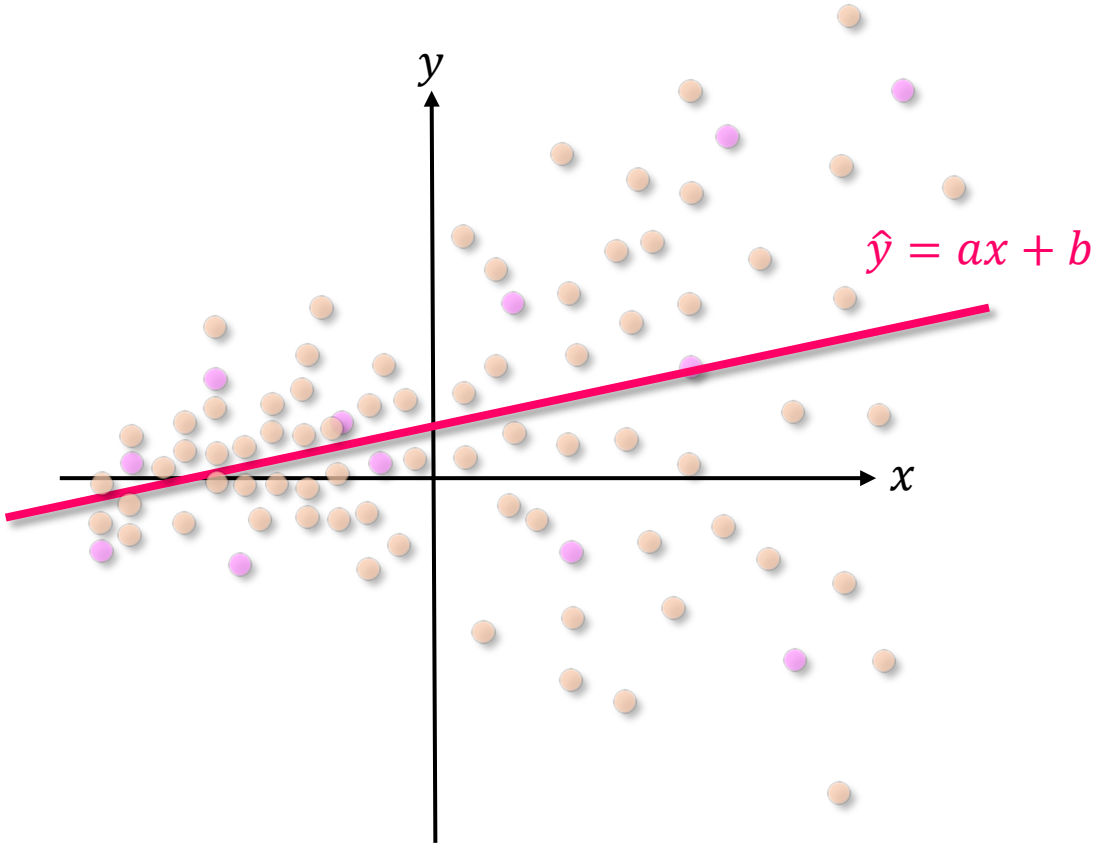
$$\log p(\hat{\mathbf{y}}|\mathbf{y}, \mathbf{X})p(\mathbf{W}) = -\frac{N}{2}\log(2\pi) - N\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y-\mu)^2 - \beta|\mathbf{W}|$$

Mean Squared Error

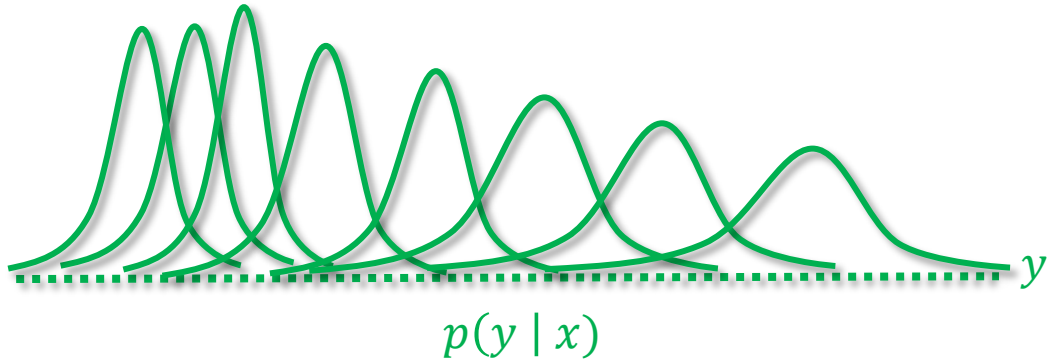
$$\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2 + \beta|\mathbf{W}|$$



Maximize Likelihood Estimation (4)



observed data unobserved data



We have infinite possibility of choosing μ and σ to approximate the true conditional.

Variational Inference

- For reconstructing a generative process. We are interested in the posterior of our latent factors.
  - In the linear regression example, the distribution of our w.
- However, not all distributions can be easily computed. For instance, given a 1-dimensional Gaussian mixture model with 3 components:

$$N_1(\mu_1, \sigma_1), N_2(\mu_2, \sigma_2), N_3(\mu_3, \sigma_3) \qquad \mathcal{Z} \sim Cat(\pi) \quad \mu \sim N(0, \tau^{-1}) \quad \sigma \sim \Gamma(a, b)$$

$$p(\mu_{1:3}, \sigma_{1:3}, z_{1:N} | \mathbf{x}) = \frac{\prod_{j=1}^{3} p(\mu_k)p(\sigma_k) \prod_{i=1}^{N} p(z_i)p(x_i|z_i,\mu_j,\sigma_j)}{\int_\sigma \int_\mu \sum_z \prod_{j=1}^{3} p(\mu_k)p(\sigma_k) \prod_{i=1}^{N} p(z_i)p(x_i|z_i,\mu_j,\sigma_j)}$$

- It will be very difficult to compute $p(z \mid x)$. So instead, we use a machine learning model to derive a posterior distribution $q(z \mid x)$ such that $q(z \mid x) \approx p(z \mid x)$.
  - We do this by minimizing the KL divergence between the two.

$$D_{KL}(q(z|x) \,||\, p(z|x))$$

Variational Autoencoder (1)

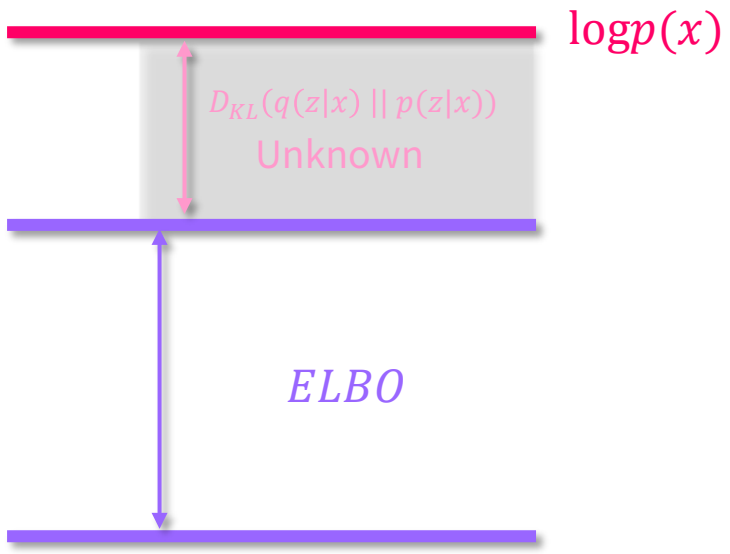
- We cannot directly maximize the log-likelihood:

$$\begin{aligned}
\log p(x) &= \log p(x) \int_z q(z|x) dz \\
&= \int_z q(z|x) \log p(x) dz \\
&= \int_z q(z|x) \log \left(\frac{p(x,z)}{p(z|x)}\right) dz \\
&= \int_z q(z|x) \log \left(\frac{p(x,z)}{q(z|x)} \frac{q(z|x)}{p(z|x)}\right) dz \\
&= \underbrace{\int_z q(z|x) \log \left(\frac{p(x,z)}{q(z|x)}\right) dz}_{L_b} + \underbrace{\int_z q(z|x) \log \left(\frac{q(z|x)}{p(z|x)}\right) dz}_{D_{KL}(q(z|x) \,||\, p(z|x)) \geq 0}
\end{aligned}$$

$$\log p(x) = L_b + D_{KL}(q(z|x) \,||\, p(z|x)) \geq L_b + 0 = L_b$$

- Instead, we maximize evidence lower bound:

$$\begin{aligned}
L_b &= \int_z q(z|x) \log \left(\frac{p(x,z)}{q(z|x)}\right) dz \\
&= \int_z q(z|x) \log \left(\frac{p(x|z)p(z)}{q(z|x)}\right) dz \\
&= \int_z q(z|x) \log p(x|z) dz + \int_z q(z|x) \log \left(\frac{p(z)}{q(z|x)}\right) dz \\
&= \int_z q(z|x) \log p(x|z) dz - D_{KL}(q(z|x) \,||\, p(z))
\end{aligned}$$



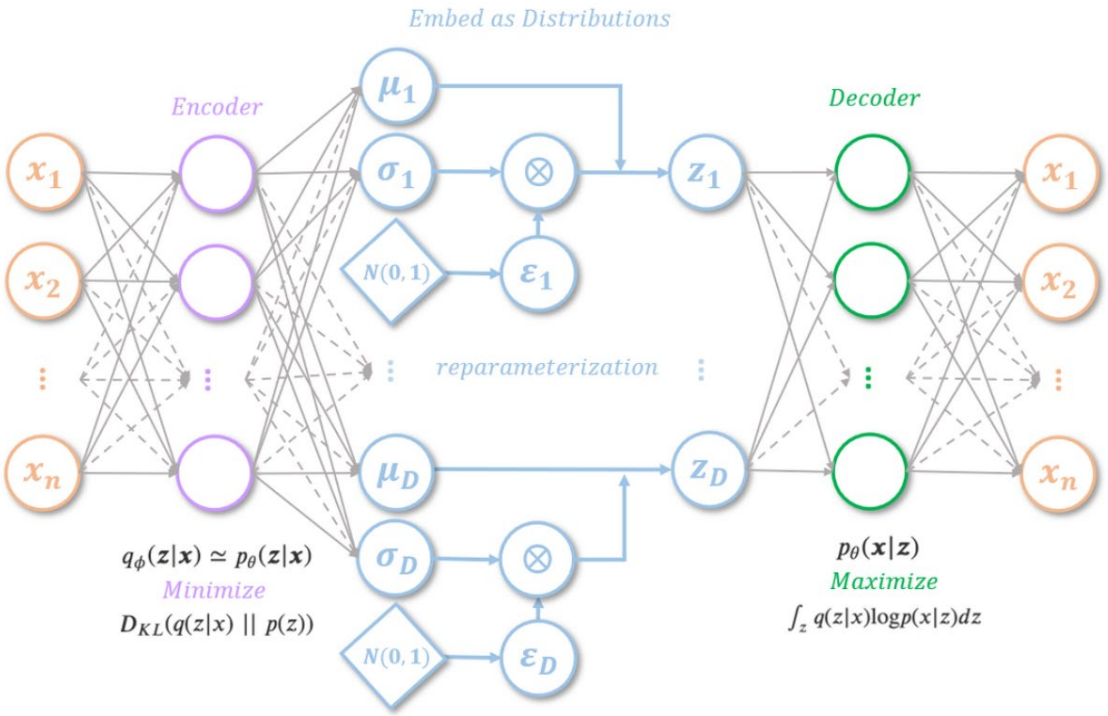
Kingma et al., ICLR 2014

Variational Autoencoder (2)

$$\int_z q(z|x)\log p(x|z)dz - D_{KL}(q(z|x) \,||\, p(z))$$

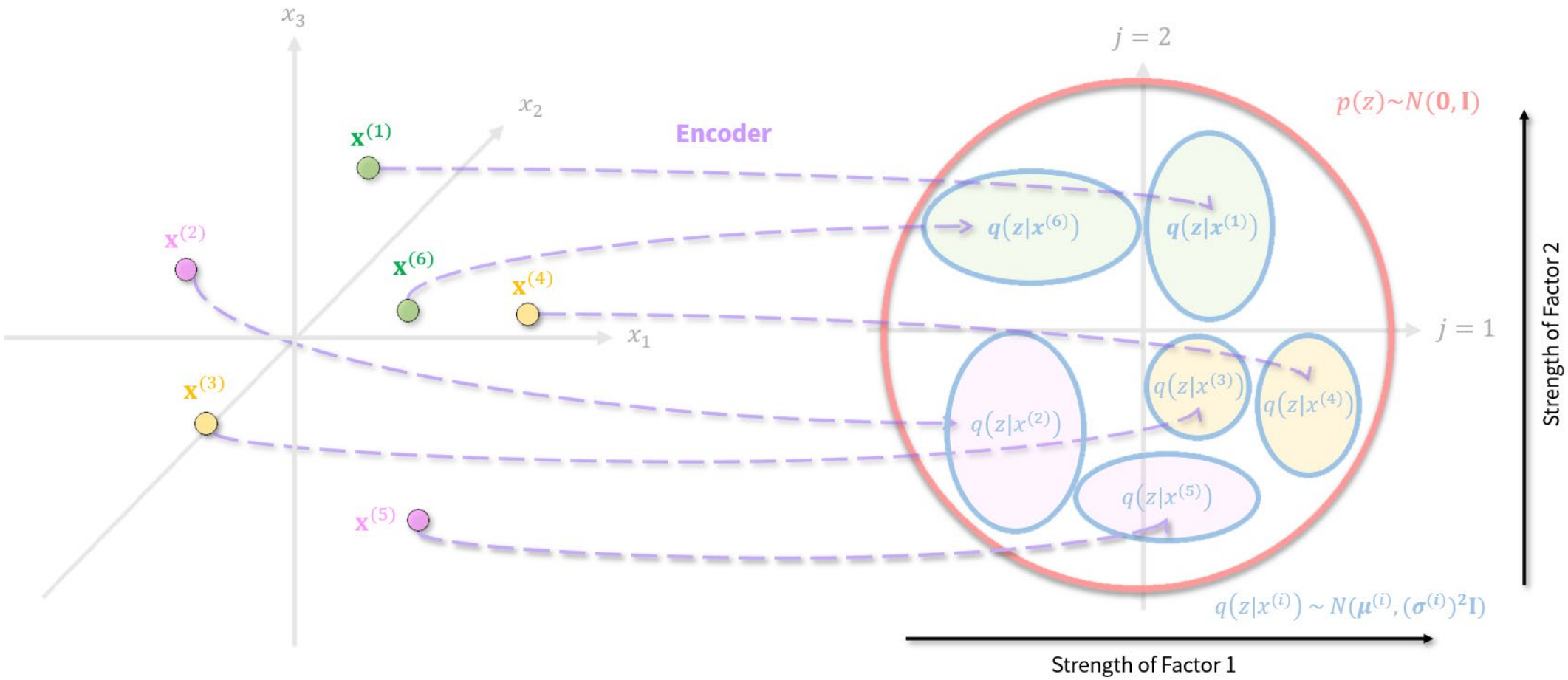
- Encoder network: approximate the posterior distribution $q_\varphi(z \mid x)$
- Decoder network: approximate the prior distribution $p_\theta(x \mid z)$
- Use a reparameterization trick to decouple backpropagation with the sampling process.

$$z_i = \mu_i + \sigma_i \varepsilon_i \qquad \varepsilon \sim N(0,1)$$



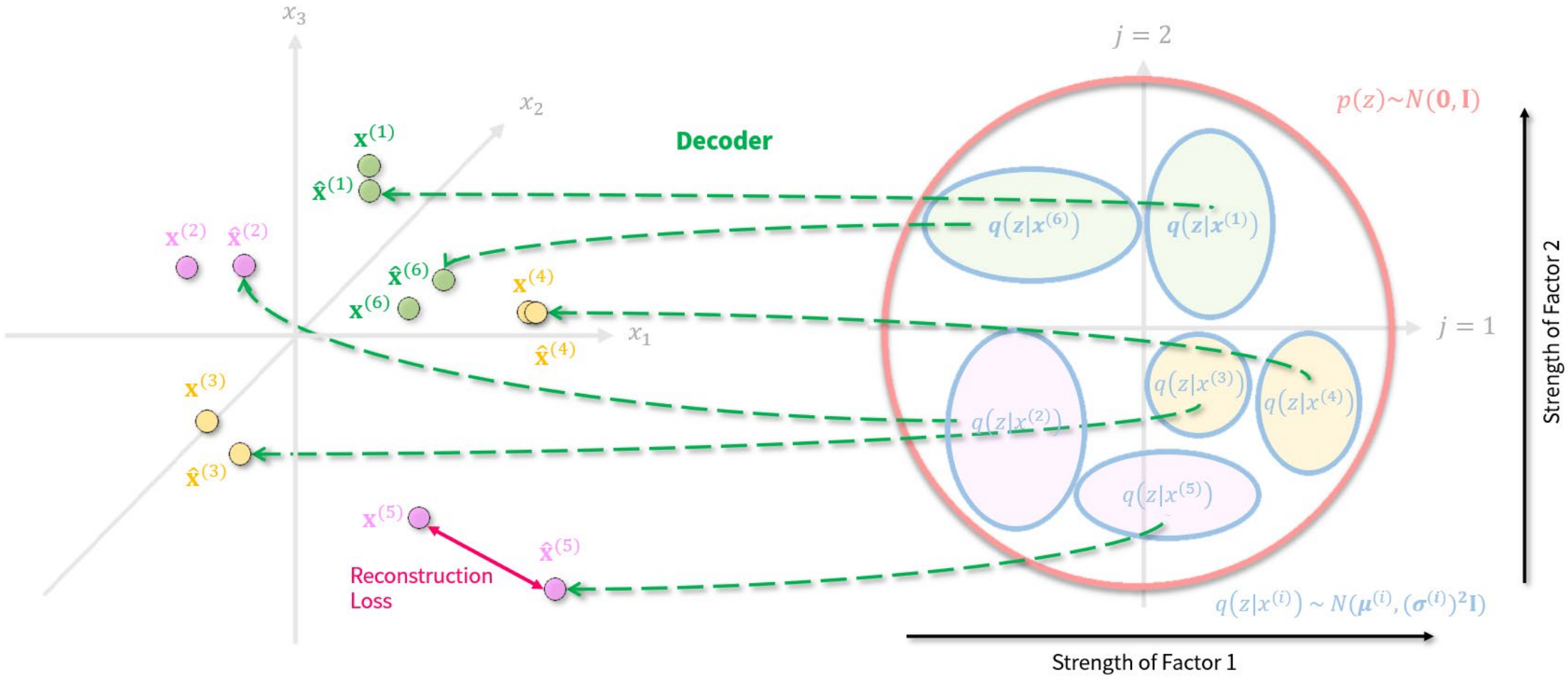
Variational Autoencoder (3)

Encoder (Approximate Posterior)



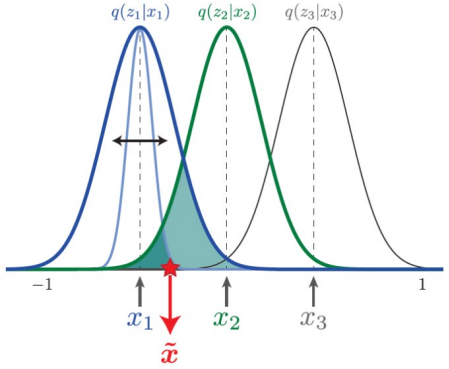
Variational Autoencoder (4)

Decoder (Generative Process)



Disentangling Variational Autoencoder

Burgess et al., NeurIPS 2017



- β-Variational Autoencoder

$$L_b = \int_z q(z|x) \log p(x|z) dz - \beta D_{KL}(q(z|x) \,||\, p(z))$$

Higgins et al., ICLR 2017

- β-Total Correlation Variational Autoencoder

$$D_{KL}(q(z|x) \,||\, p(z)) = D_{KL}(q(z|n)p(n) \,||\, q(z)p(n)) + D_{KL}(q(z) \,||\, \prod_j q(z_j)) + \sum_j D_{KL}(q(z_j) \,||\, p(z_j))$$

$$L_b = \int_z q(z|x) \log p(x|z) dz - D_{KL}(q(z|x) \,||\, p(z)) - (\beta - 1) D_{KL}(q(z) \,||\, \prod_j q(z_j))$$

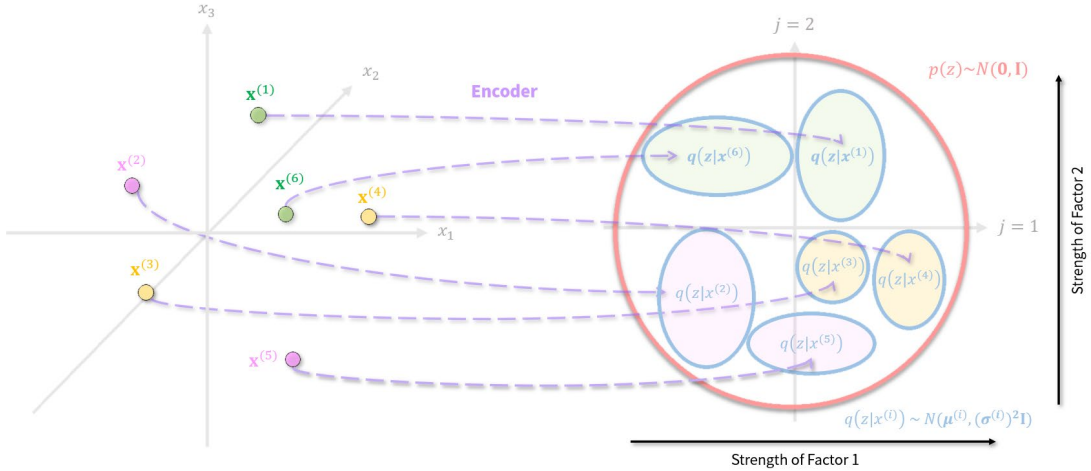
Chen et al., NeurIPS 2018

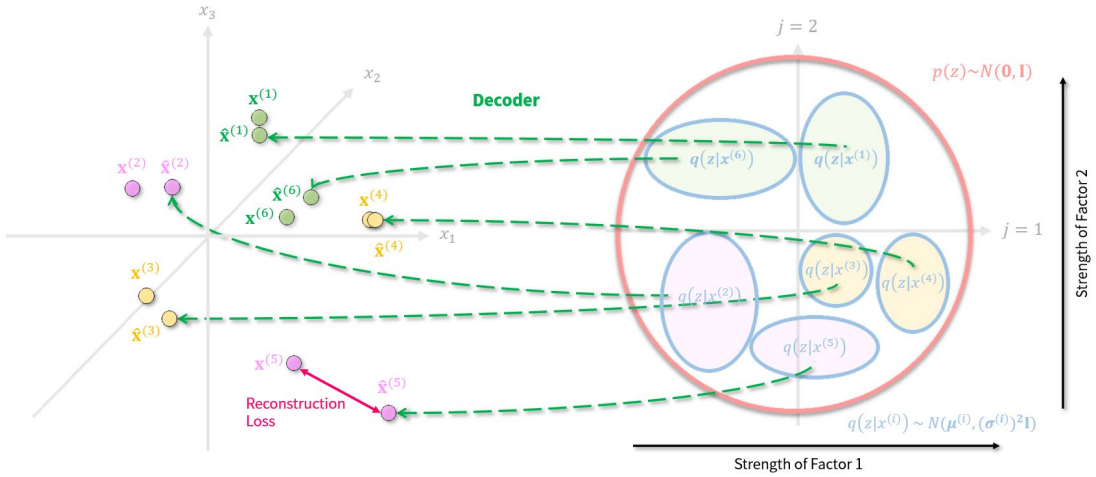
- Factor Variational Autoencoder

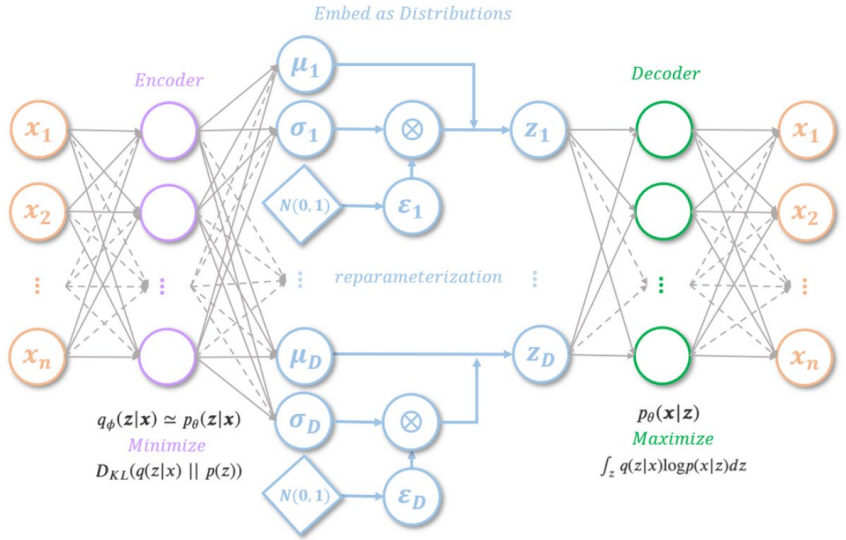
$$TC(z) = D_{KL}(q(z) \,||\, \prod_j q(z_j)) = \mathbb{E}_{q(z)}[\log(\frac{q(z)}{\prod_j q(z_j)})] \sim \mathbb{E}_{q(z)}[\frac{D(z)}{1-D(z)}]$$

Kim and Mihn., PMLR 2018

Variational Autoencoder (3)



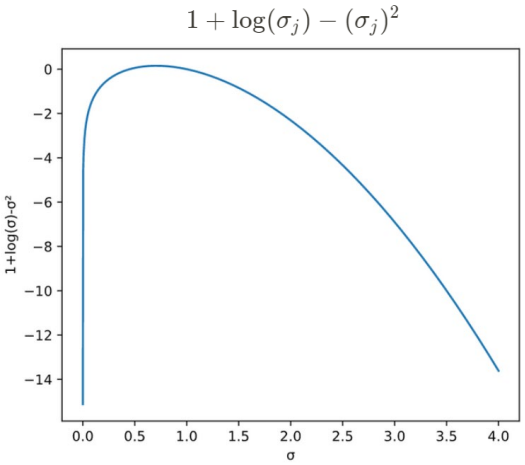




$$
\begin{aligned}
L_b &= \int_z q(z|x)\log(\frac{p(x,z)}{q(z|x)})dz \\
&= \int_z q(z|x)\log(\frac{p(x|z)p(z)}{q(z|x)})dz \\
&= \int_z q(z|x)\log p(x|z)dz + \int_z q(z|x)\log(\frac{p(z)}{q(z|x)})dz \\
&= \int_z q(z|x)\log p(x|z)dz - D_{KL}(q(z|x) \,||\, p(z))
\end{aligned}
$$

Gaussian Case

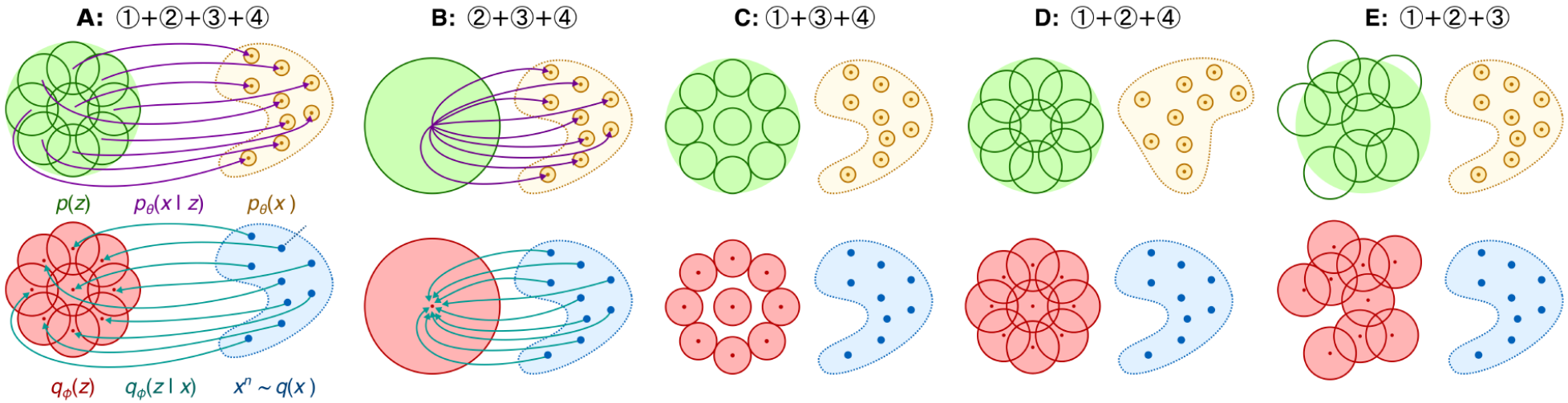
$$-D_{KL}(q(z|x) \,||\, p(z)) = \frac{1}{2}\sum_{j=1}^{D}[1 + \log(\sigma_j) - (\sigma_j)^2 - (\mu_j)^2]$$



More on Decomposition of the ELBO

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\boldsymbol{z}, \boldsymbol{x})} \left[\log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{p_\theta(\boldsymbol{x}) p(\boldsymbol{z})} + \log \frac{q_\phi(\boldsymbol{z}) q(\boldsymbol{x})}{q_\phi(\boldsymbol{z}, \boldsymbol{x})} + \log \frac{p_\theta(\boldsymbol{x})}{q(\boldsymbol{x})} + \log \frac{p(\boldsymbol{z})}{q_\phi(\boldsymbol{z})} \right],$$

$$= \mathbb{E}_{q_\phi(\boldsymbol{z}, \boldsymbol{x})} \Bigg[\underbrace{\log \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z})}{p_\theta(\boldsymbol{x})}}_{①} - \underbrace{\log \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{q_\phi(\boldsymbol{z})}}_{②} \Bigg] - \underbrace{\mathrm{KL}\left(q(\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{x})\right)}_{③} - \underbrace{\mathrm{KL}\left(q_\phi(\boldsymbol{z}) \,||\, p(\boldsymbol{z})\right)}_{④}$$



Esmaeili et al., AISTATS 2018

Identifiability of Generative Model

Impossibility of Disentangling (1)

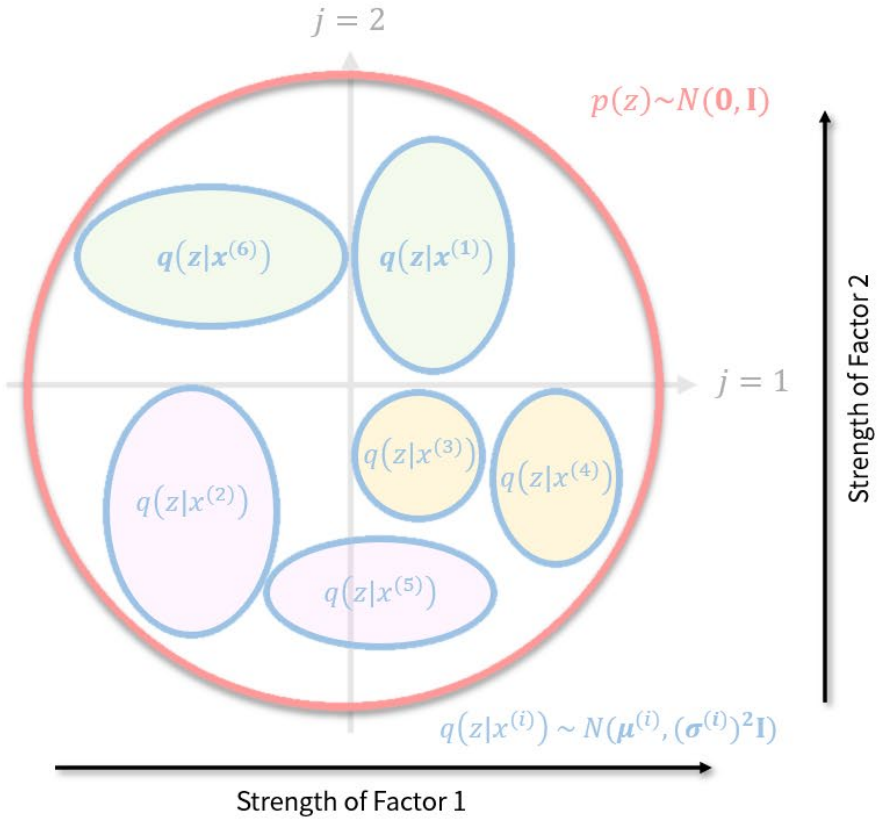
- **Theorem**
  - If $p(z) = \prod_{i=1}^{k} p(z_i)$ is the only constraint in the generative process $p(x \mid z)$.
  - There exists infinite bijective function $f: supp(z) \rightarrow supp(z)$ such that $\frac{\partial f_i(u)}{\partial u_j} \neq 0$ almost everywhere for every i and j.
  - $P(z \leq u) = P(f(z) \leq u)$ for all $u \in supp(z)$.

Locatello et al., AISTATS 2019

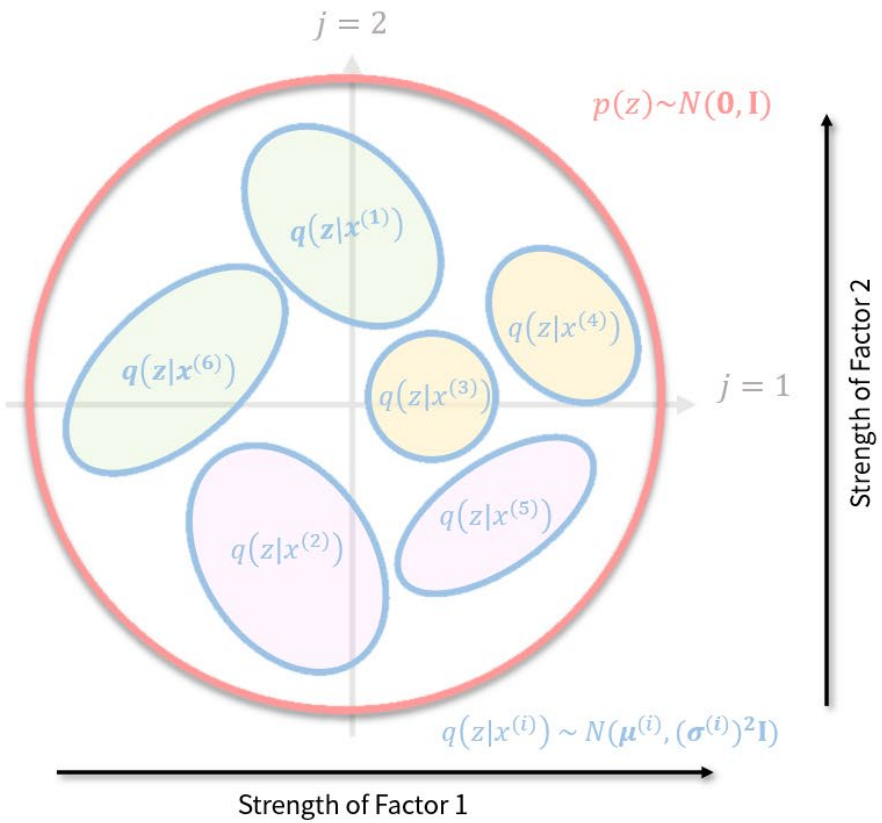
- Theoretically, by simply using the input data, β-Variational autoencoder **cannot** identify the difference between disentangled and entangled representation.

- In practice, the inductive bias on the model and data help VAE to find disentangled representation.

Impossibility of Disentangling (2)

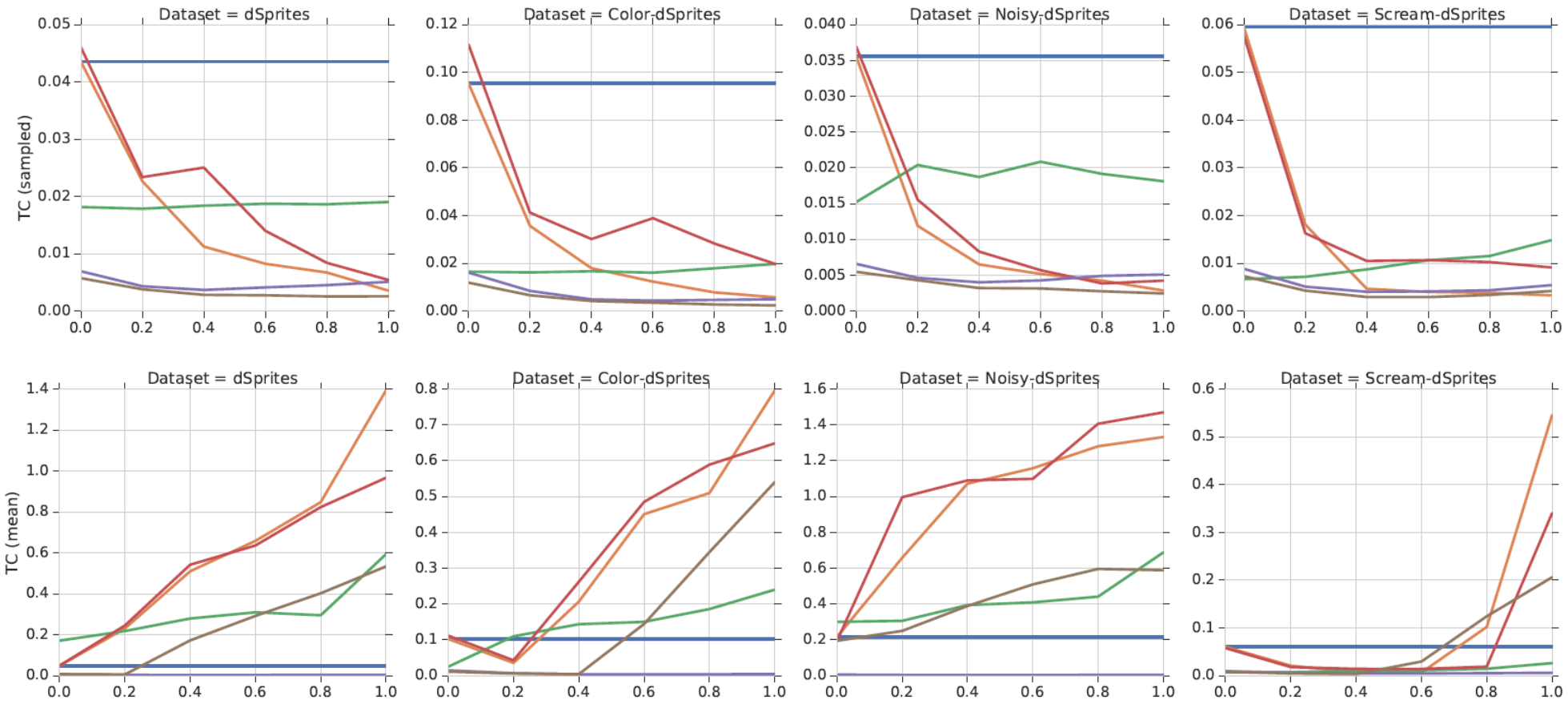


Joint distribution is statistically the same, the drawn conclusion can be completely different.



Disentangled representation imply clustering
Clustering do not imply disentangled representation

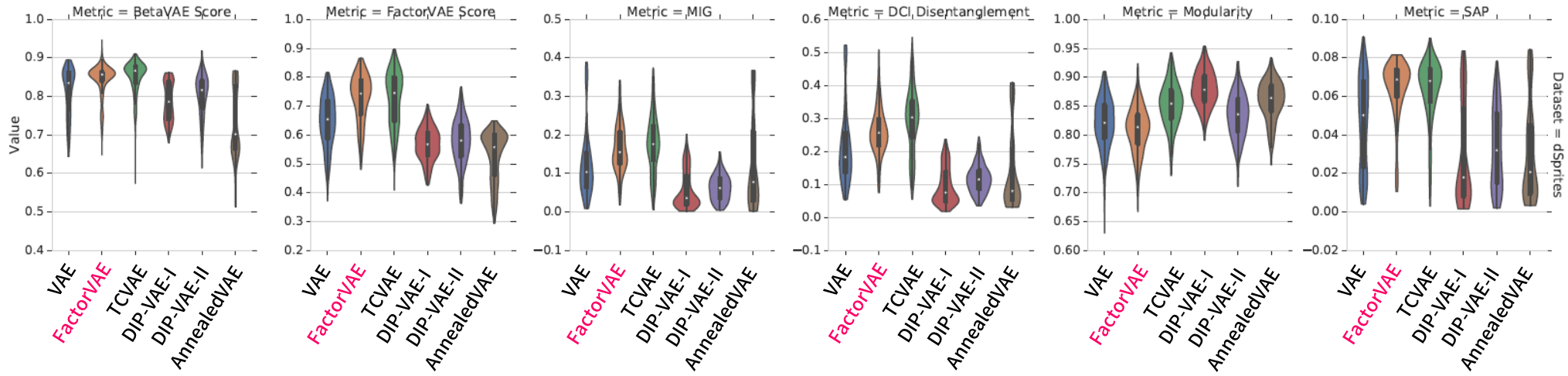
β-VAEs Do Not Disentangle the Posterior Mean



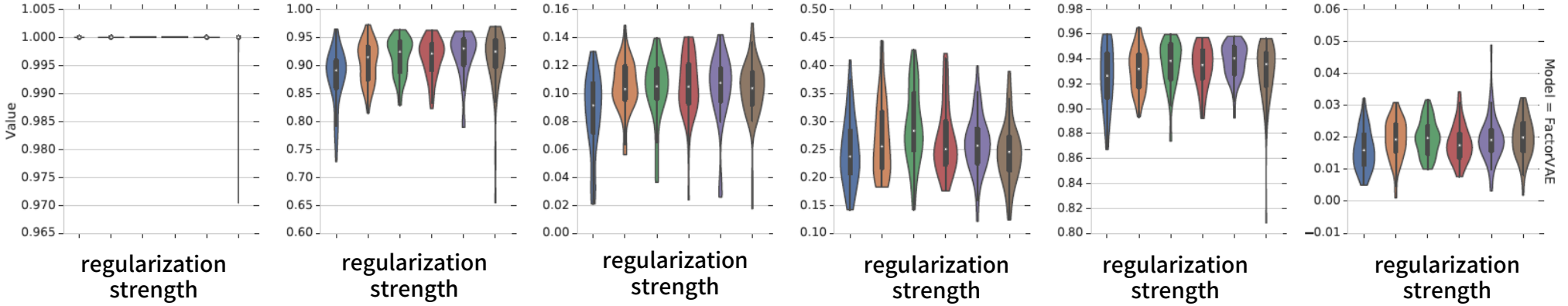
increase regularization strength do not decrease the total correlation of posterior mean

Random Seeds Matter More than the Parameter

Random seed is more relevant than the choice of model

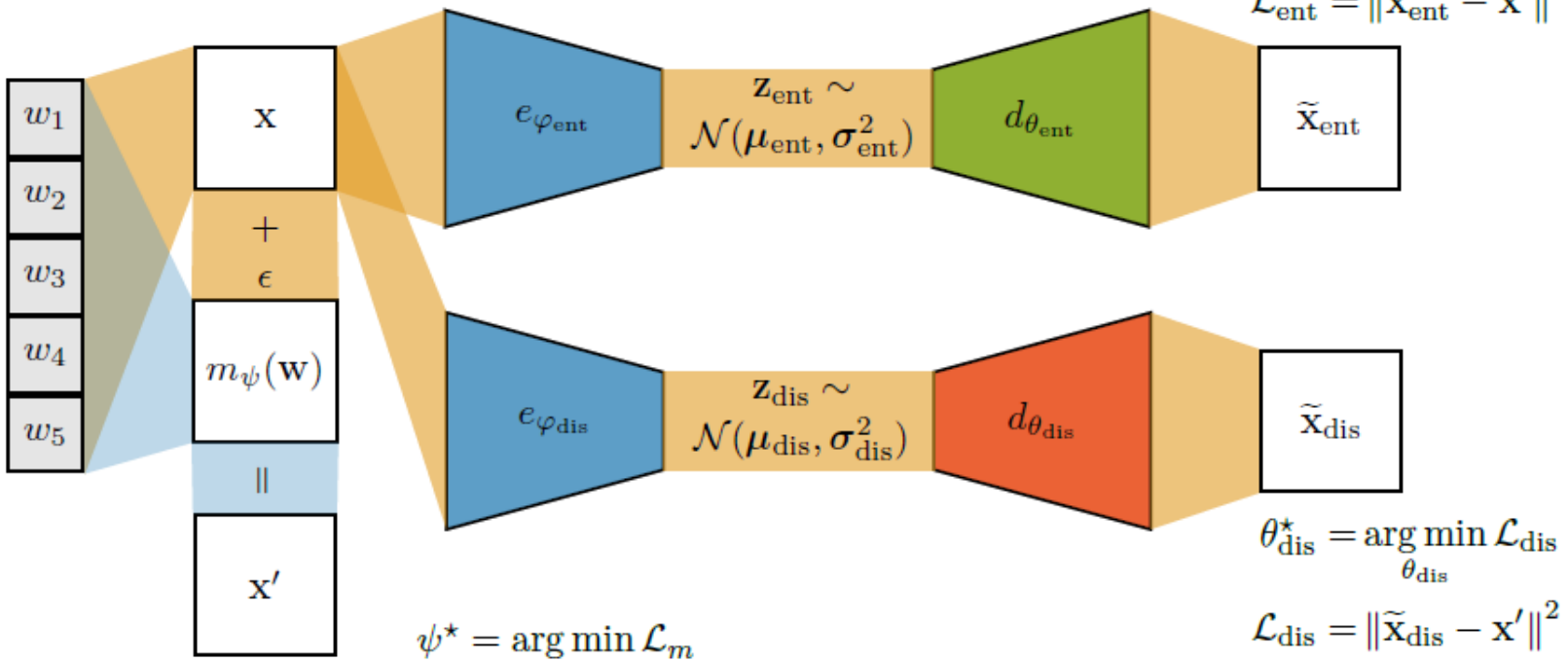


Random seed is more relevant than the parameter (regularization strength)



Inductive Bias from the Data

Modify the decoder so that the entangled representation yield smaller reconstruction error



$$\theta^\star_{\text{ent}} = \arg\min_{\theta_{\text{ent}}} \mathcal{L}_{\text{ent}}$$

$$\mathcal{L}_{\text{ent}} = \|\widetilde{\mathbf{x}}_{\text{ent}} - \mathbf{x}'\|^2$$

$$\psi^\star = \arg\min_{\psi} \mathcal{L}_m$$

$$\mathcal{L}_m = \mathcal{L}_{\text{ent}} - \mathcal{L}_{\text{dis}}$$

$$\theta^\star_{\text{dis}} = \arg\min_{\theta_{\text{dis}}} \mathcal{L}_{\text{dis}}$$

$$\mathcal{L}_{\text{dis}} = \|\widetilde{\mathbf{x}}_{\text{dis}} - \mathbf{x}'\|^2$$

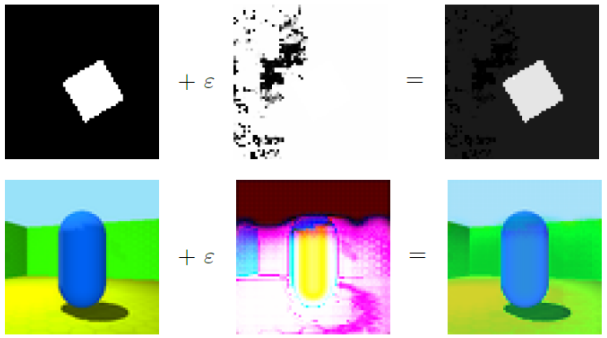
Zietlow et al., arXiv 2021

Inductive Bias from the Data

Modify the datasets:

$$\psi^\star = \arg\min_{\psi} \left(\mathcal{L}_{\text{rec}}^{\text{ent}} \left(\mathbf{x}'^{(i)}, \mathbf{z}^{(i)} \right) - \mathcal{L}_{\text{rec}}^{\text{dis}} \left(\mathbf{x}'^{(i)}, \mathbf{z}^{(i)} \right) \right)$$

Introduce a network to learn the additive manipulation which minimize the reconstruction loss of the entangled VAE and to increase the loss of the disentangled VAE



	dSprites			Shapes3d		
	orig.	**mod.**	**noise**	**orig.**	**mod.**	**noise**
AE	0.09 ± 0.06	–	–	0.06 ± 0.03	–	–
β-VAE	0.23 ± 0.08	0.07 ± 0.09	0.14 ± 0.07	0.60 ± 0.31	0.09 ± 0.14	0.66 ± 0.05
Fac. VAE	0.27 ± 0.11	0.20 ± 0.12	0.16 ± 0.08	0.27 ± 0.18	0.07 ± 0.05	0.33 ± 0.20
TC-β-VAE	0.25 ± 0.08	0.14 ± 0.10	0.20 ± 0.04	0.58 ± 0.20	0.24 ± 0.16	0.60 ± 0.11
Slow-VAE	0.39 ± 0.08	0.27 ± 0.08	0.37 ± 0.09	0.53 ± 0.19	0.13 ± 0.08	0.60 ± 0.10
PCL	0.21 ± 0.03	0.24 ± 0.07	0.24 ± 0.07	0.44 ± 0.06	0.47 ± 0.08	0.40 ± 0.07

Zietlow et al., arXiv 2021

Identifiable Variational Autoencoder

- Identifiable Variational Autoencoder (Non-linear ICA) Khemakhem et al., AISTATS 2020

$$p_{\theta^*}(\mathbf{x}, \mathbf{z}|\mathbf{u}) = p_{\mathbf{f}^*}(\mathbf{x}|\mathbf{z}) p_{\mathbf{T}^*,\lambda^*}(\mathbf{z}|\mathbf{u})$$

$$p_{\mathbf{T},\lambda}(\mathbf{z}|\mathbf{u}) = \prod_i \frac{Q_i(z_i)}{Z_i(\mathbf{u})} \exp[\sum_{j=1}^{d} T_{ij}(z_i)\lambda_{i,j}(\mathbf{u})]$$

* The latent representation is theoretically equivalent to the true factor (given from the condition) up to a linear invertible transformation

* The equivalent to the true factor (given from the condition) can up to a permutation transformation if the parameters for each distribution ≥ 2 (e.g. Independent Multivariate Gaussian)

- Variational Deep Embedding (VaDE)

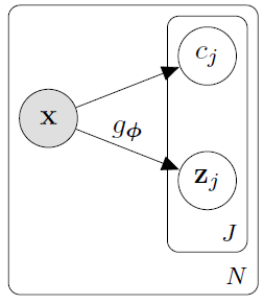
$$\mathcal{L}(\mathbf{x}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z},c|\mathbf{x})}[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}|c) p_\theta(c)}{q_\phi(\mathbf{z}|\mathbf{x}) q_\phi(c|\mathbf{x})}] = \mathbb{E}_{q_\phi(\mathbf{z},c|\mathbf{x})}[\log \frac{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}) p_\theta(c|\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x}) q_\phi(c|\mathbf{x})}]$$

$$\mathcal{L}(\mathbf{x}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{KL}\left[q_\phi(c|\mathbf{x})||p_\theta(c|\mathbf{z})\right]$$

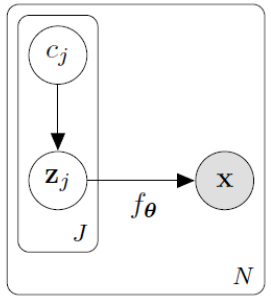
KL Divergence in original VAE. But prior is computed separately for each group

The clustering distribution constraint by the prior that assigning clusters with latent representation

Variational Posterior



Generative Process



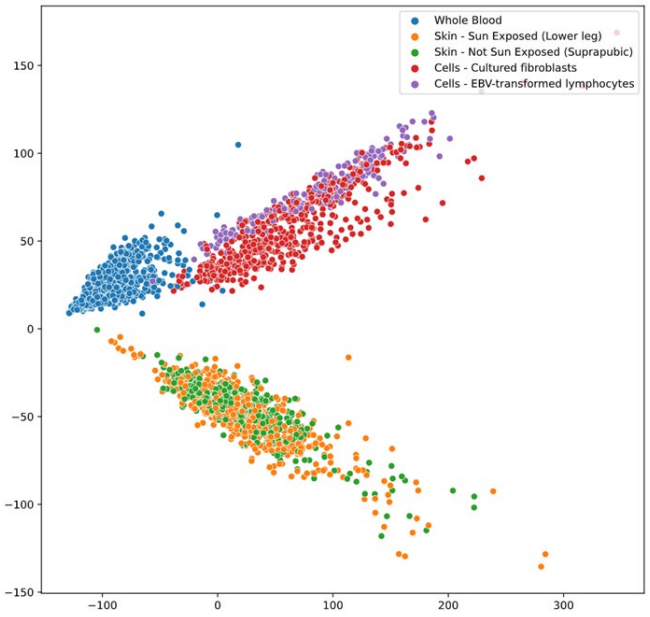
Jiang et al., IJCAI 2017
Willetts et al., arXiv 2021

On GTEx Skin, Whole Blood and Cells

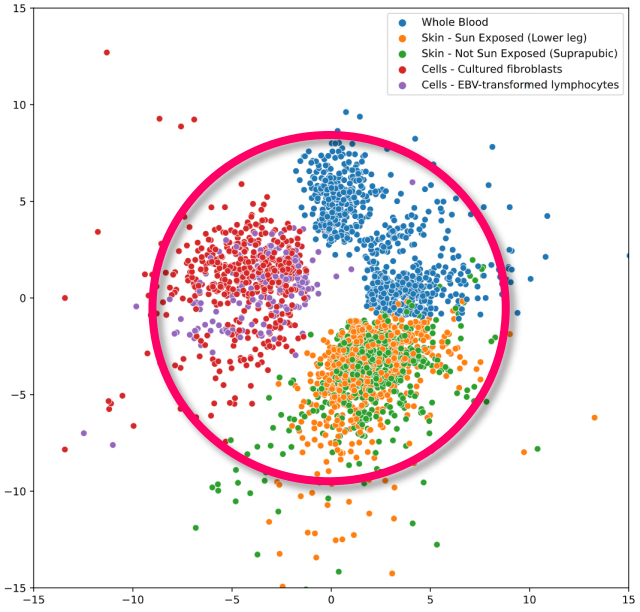
Comparison of Latent Space (1)

*** posterior mean is shown**

Principal Component Analysis

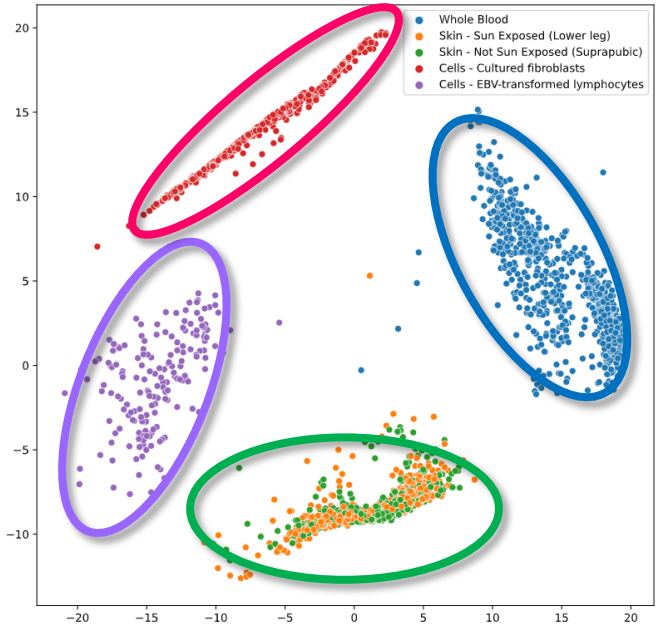


β-Variational Autoencoder ($\beta = 1.5$)



single prior for all groups

VaDE

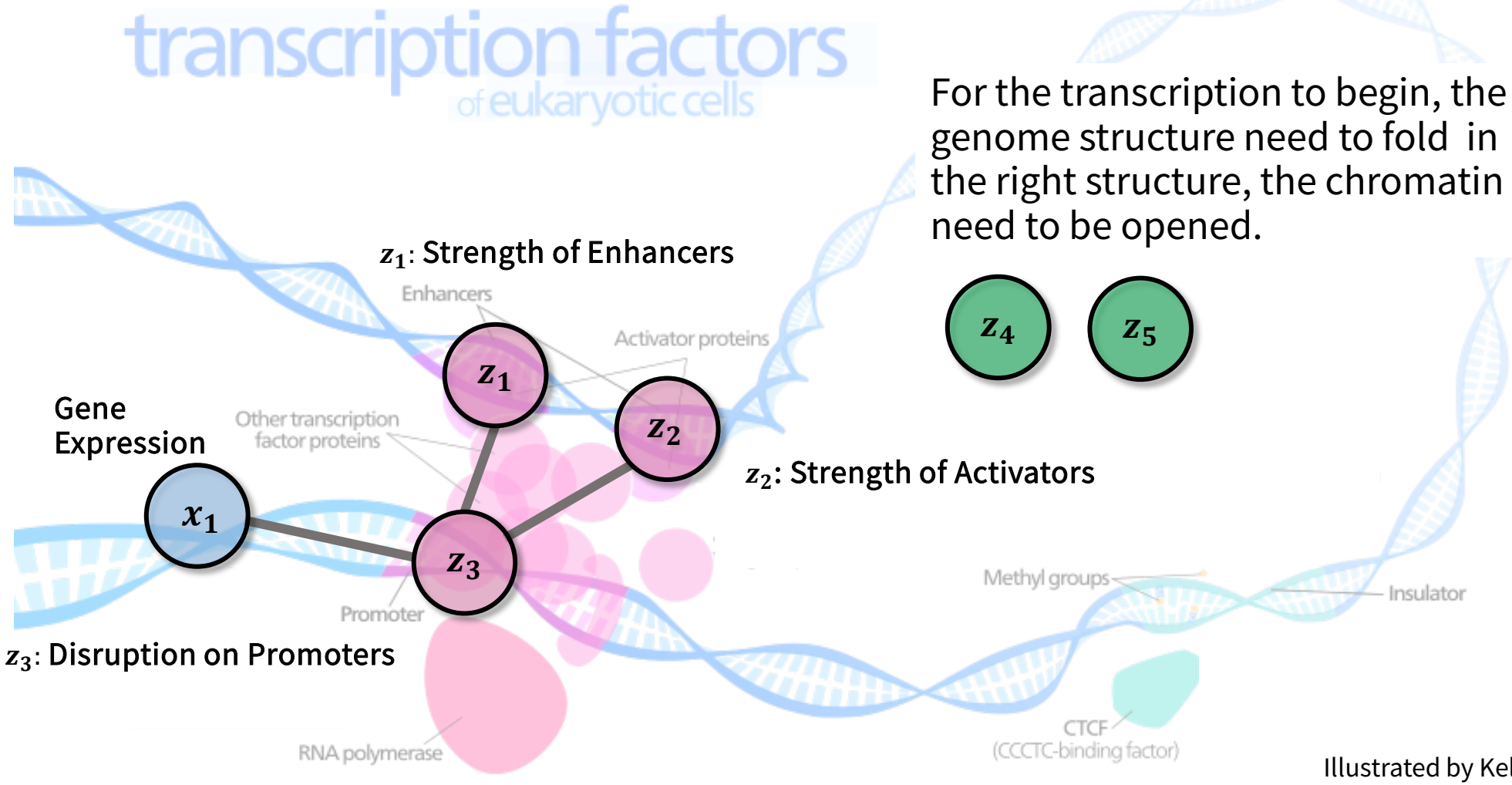


different prior for different groups

*group information is not provided to the model

More on Deep Generative Models

What if There is a Hierarchical Relationship between Latent Factors?



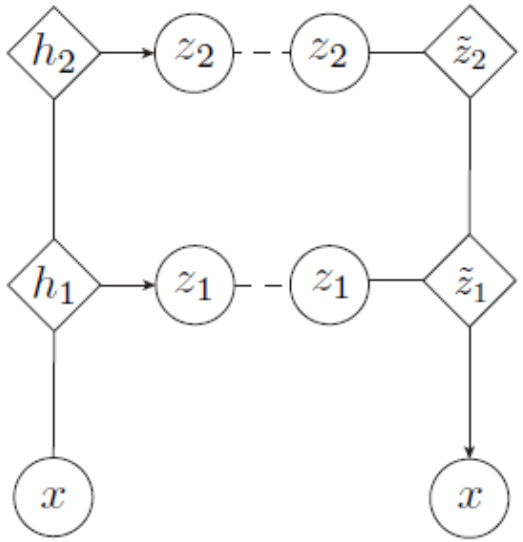
For the transcription to begin, the genome structure need to fold in the right structure, the chromatin need to be opened.

z_4 z_5

Illustrated by Kelvin Ma

Hierarchical Variational Autoencoders (1)

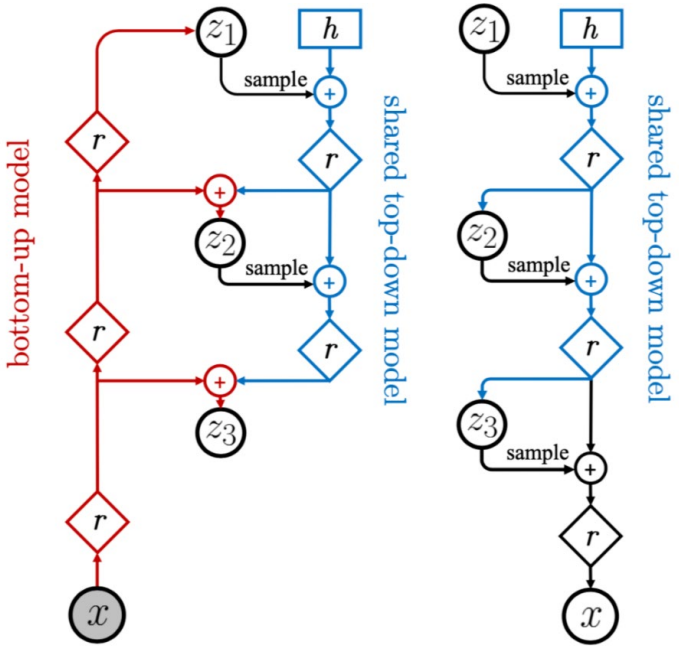
Variational Ladder Autoencoder



Zhao et al., PMLR 2017

$$\mathbf{h}_\ell = \mathbf{g}_\ell(\mathbf{h}_{\ell-1})$$
$$\mathbf{z}_\ell \sim \mathcal{N}(\boldsymbol{\mu}_\ell(\mathbf{h}_\ell), \boldsymbol{\sigma}_\ell(\mathbf{h}_\ell))$$

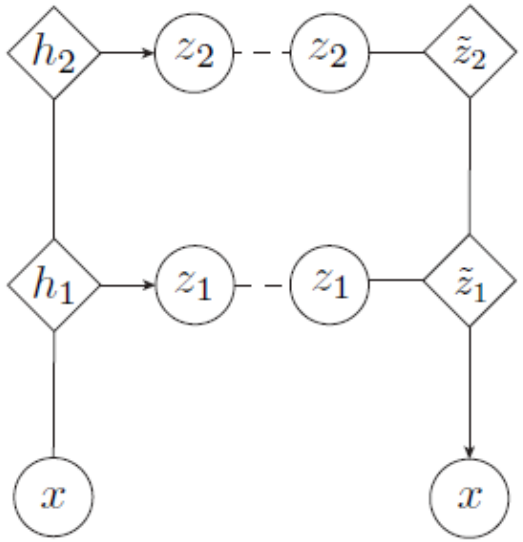
Nouveau Variational Autoencoder (NVAE)



Vahdat and Kautz. NeurIPS 2020

Hierarchical Variational Autoencoders (2)

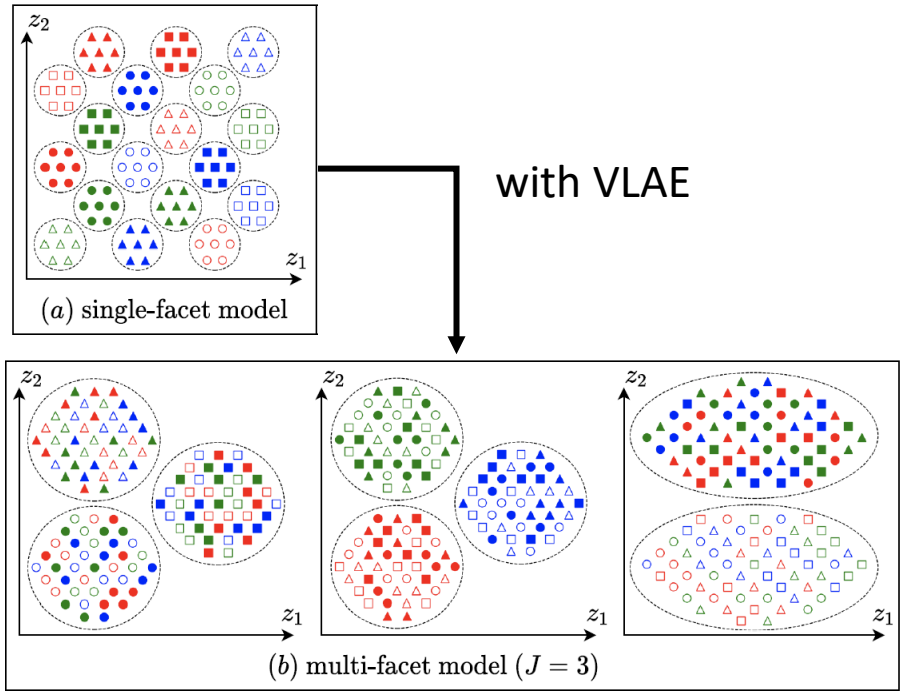
Variational Ladder Autoencoder



Zhao et al., PMLR 2017

$$\mathbf{h}_\ell = \mathbf{g}_\ell(\mathbf{h}_{\ell-1})$$
$$\mathbf{z}_\ell \sim \mathcal{N}(\boldsymbol{\mu}_\ell(\mathbf{h}_\ell), \boldsymbol{\sigma}_\ell(\mathbf{h}_\ell))$$

Multi-Facet Clustering Variational Autoencoder



Falck et al., arXiv 2021

MFC-Variational Autoencoders (1)

Hierarchy 3

clusters

generative images

Hierarchy 2

clusters

generative images

Hierarchy 1

clusters

generative images

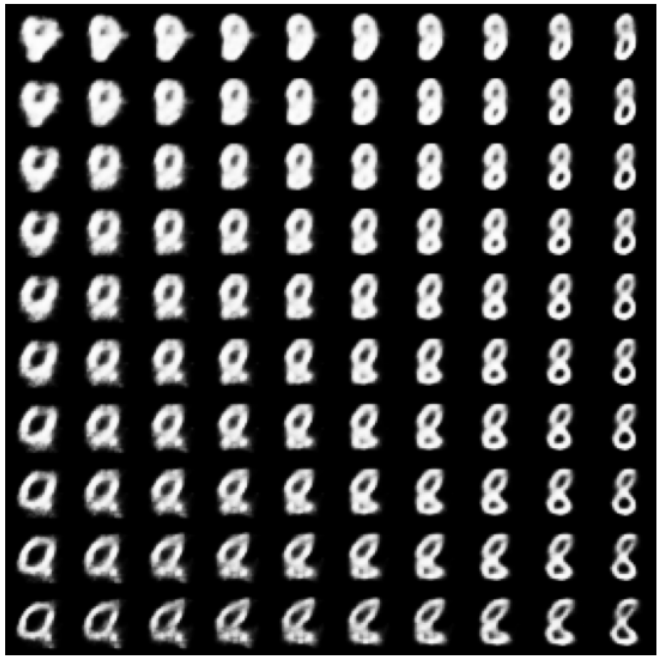
MFC-Variational Autoencoders (2)

Hierarchy 3

digit

digit

Hierarchy 2



stroke width

# of circle

Hierarchy 1

tilt

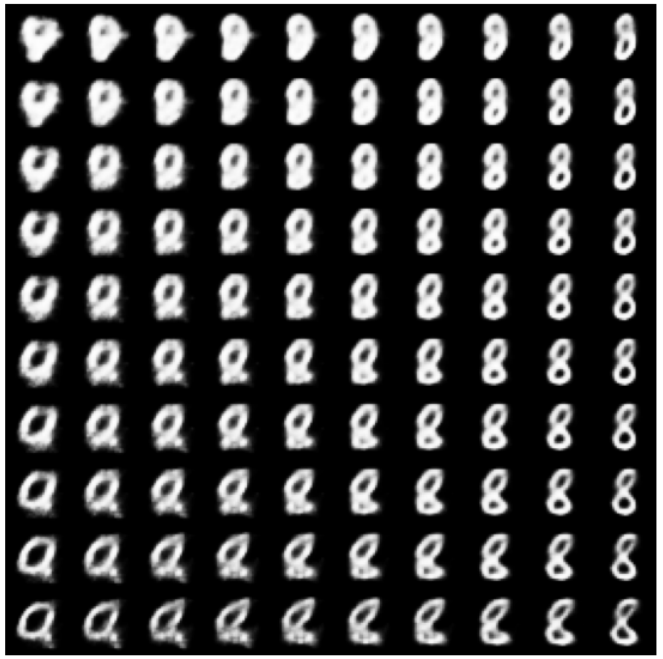
MFC-Variational Autoencoders (3)

Hierarchy 3

digit

digit

Hierarchy 2



stroke width

# of circle

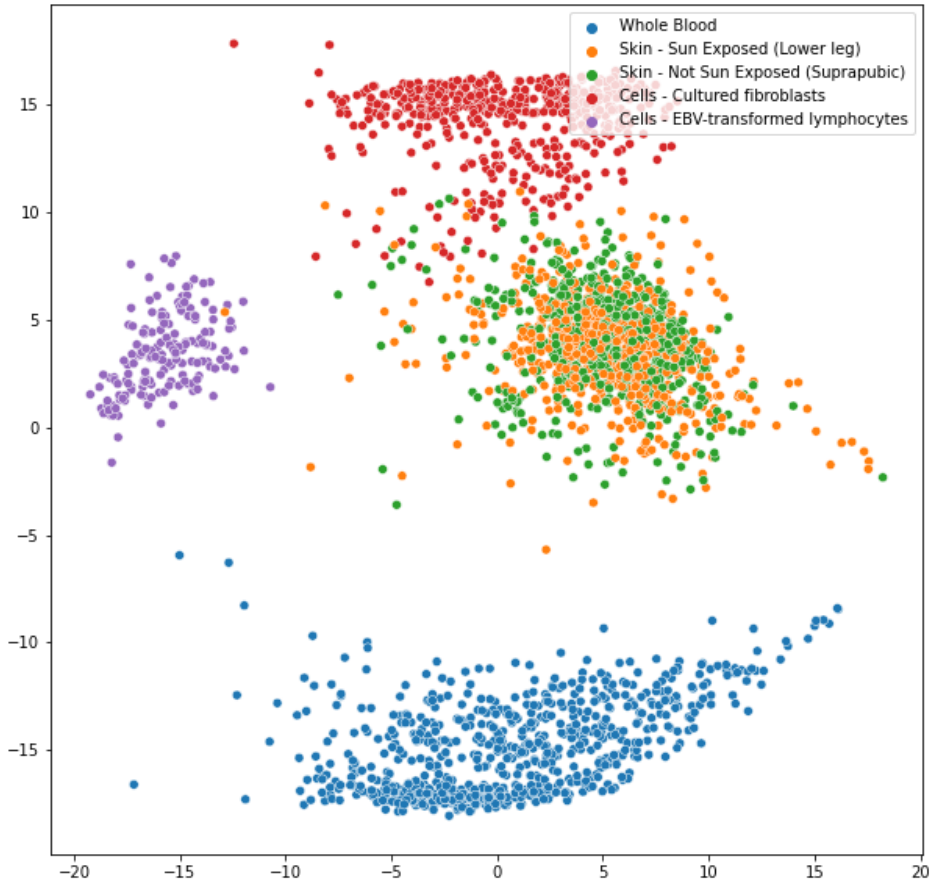
Hierarchy 1

tilt

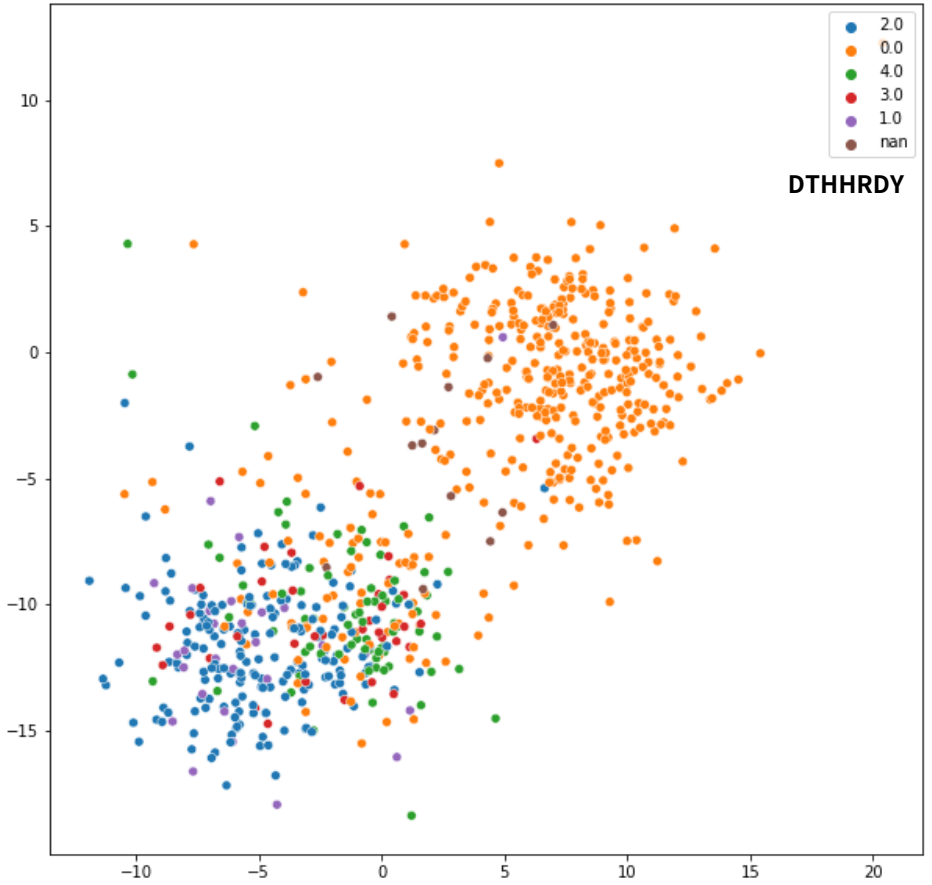
GTEx Preliminary Result

MFC-Variational Autoencoders (2)

Layer 1



Layer 2 – Condition on Whole Blood



Modeling scRNA-Seq

Modeling scRNA with Variational Autoencoder (1)

scVI (Lopez et al., Nature Methods 2018)

Latent representation for biological variability $\mathbf{z}^{(n)} \sim N(0, I)$

Latent representation for library size (**per batch**) $\mathbf{l}^{(n)} \sim N(l_\mu^{(n)}, l_\sigma^{(n)})$

posterior → $q(z|x,s)$, $q(l|x,s)$

Batch corrected estimation for expressed proportion $\rho^{(n)} = f_w(\mathbf{z}^{(n)}, s^{(n)})$ ← **batch annotation**

Weight for the Poisson mean $w_g^{(n)} \sim \text{Gamma}(\rho_g^{(n)}, \theta)$

Estimated read count if not dropped out $y_g^{(n)} \sim \text{Poisson}(l_g^{(n)} w_g^{(n)})$

gene specific dispersion
- **cell specific**
- **shared across batch**
- **shared across cell**

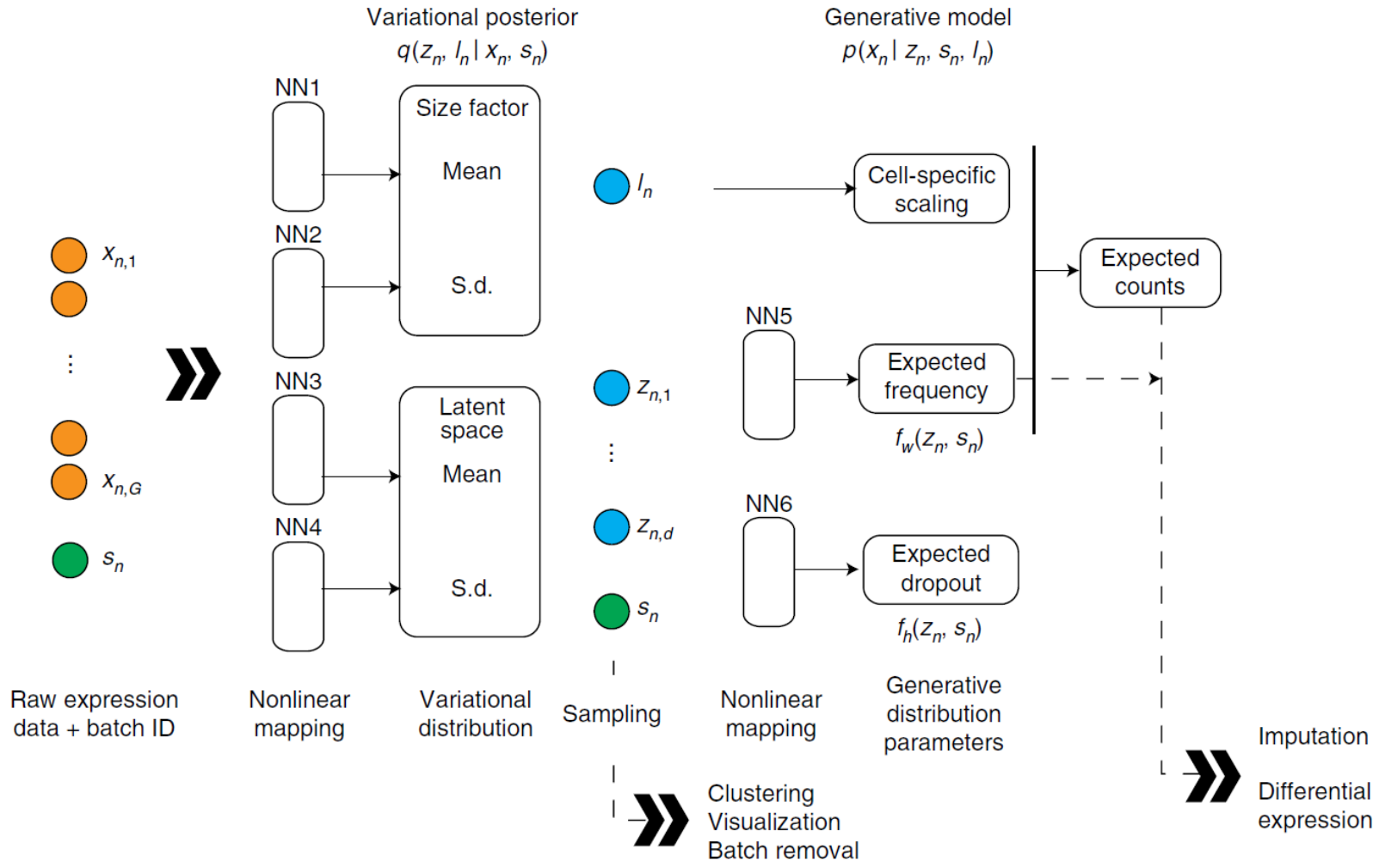
Estimated dropout rate $\pi_g^{(n)} = f_g^{(n)}(\mathbf{z}^{(n)}, \mathbf{s}^{(n)})$

Random variable for dropout rate $h_g^{(n)} \sim \text{Bernoulli}(\pi_g^{(n)})$

Estimated read count $x_g^{(n)} = \begin{cases} y_g^{(n)} & \text{if } h_g^{(n)} = 0 \\ 0 & \text{otherwise} \end{cases}$

Identifiable Model

Modeling scRNA with Variational Autoencoder (2)



Modeling scRNA with Variational Autoencoder (3)

scVAE (Grønbech et al., Bioinformatics 2020)

Inferred cluster (batch) $\mathbf{y}^{(n)} \sim \mathrm{Cat}(y|\gamma)$

Latent representation for variability $\mathbf{z}^{(n)} \sim N(0, I)$

Estimated expressed proportion $\rho^{(n)} = f_w(\mathbf{z}^{(n)})$

Estimated read count if not dropped out $\hat{x}_g^{(n)} \sim \mathrm{Poisson}(l_g^{(n)} \rho_g^{(n)})$ *

Estimated dropout rate $\pi_g^{(n)} = f_g^{(n)}(\mathbf{z}^{(n)}, \mathbf{y}^{(n)})$

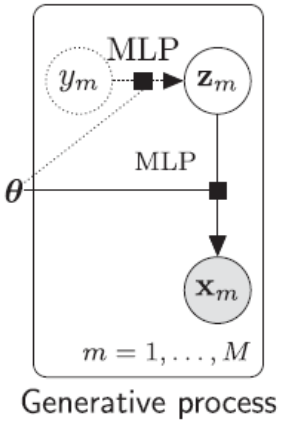
Random variable for dropout rate $h_g^{(n)} \sim \mathrm{Bernoulli}(\pi_g^{(n)})$

Estimated read count $x_g^{(n)} = \begin{cases} \hat{x}_g^{(n)} & \text{if } h_g^{(n)} = 0 \\ 0 & \text{otherwise} \end{cases}$

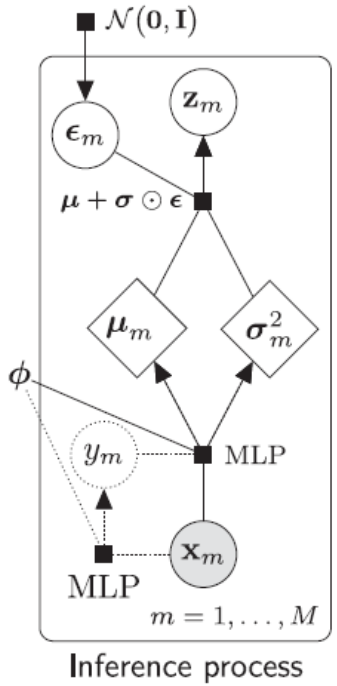
posterior and prior

$q(z|x, y) \sim N(\mu_y, \sigma_y^2)$

$p(x|z) \sim \mathrm{Bernoulli}(x|\mu_x)$



Generative process



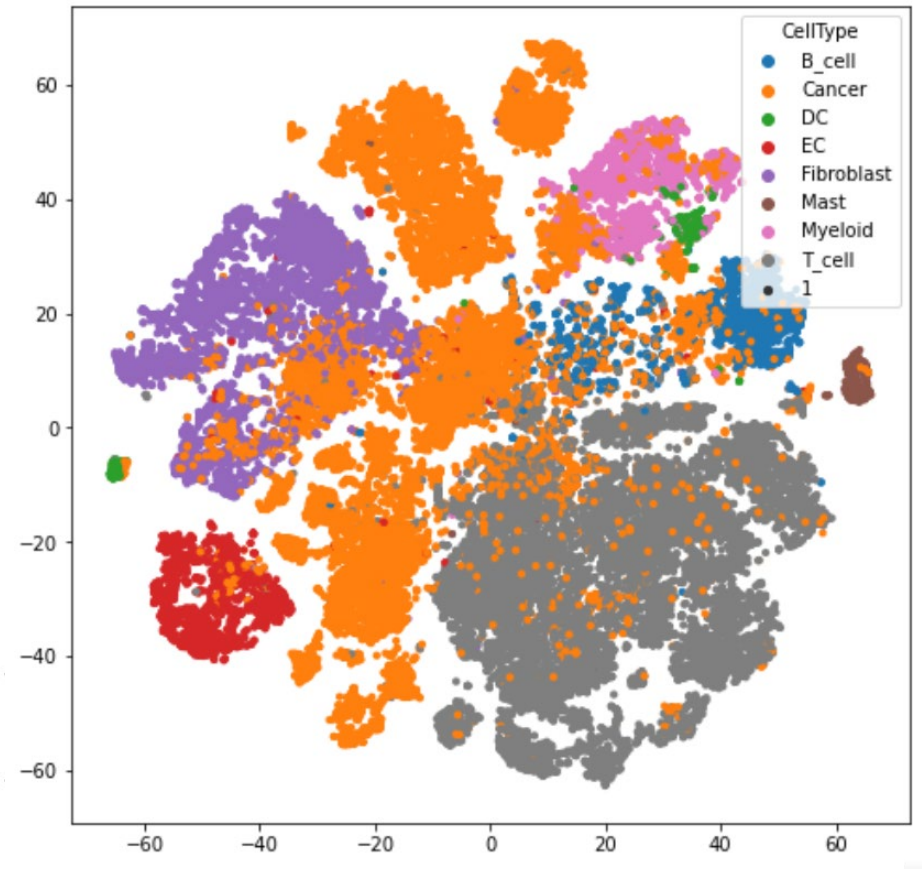
Inference process

Identifiable Model

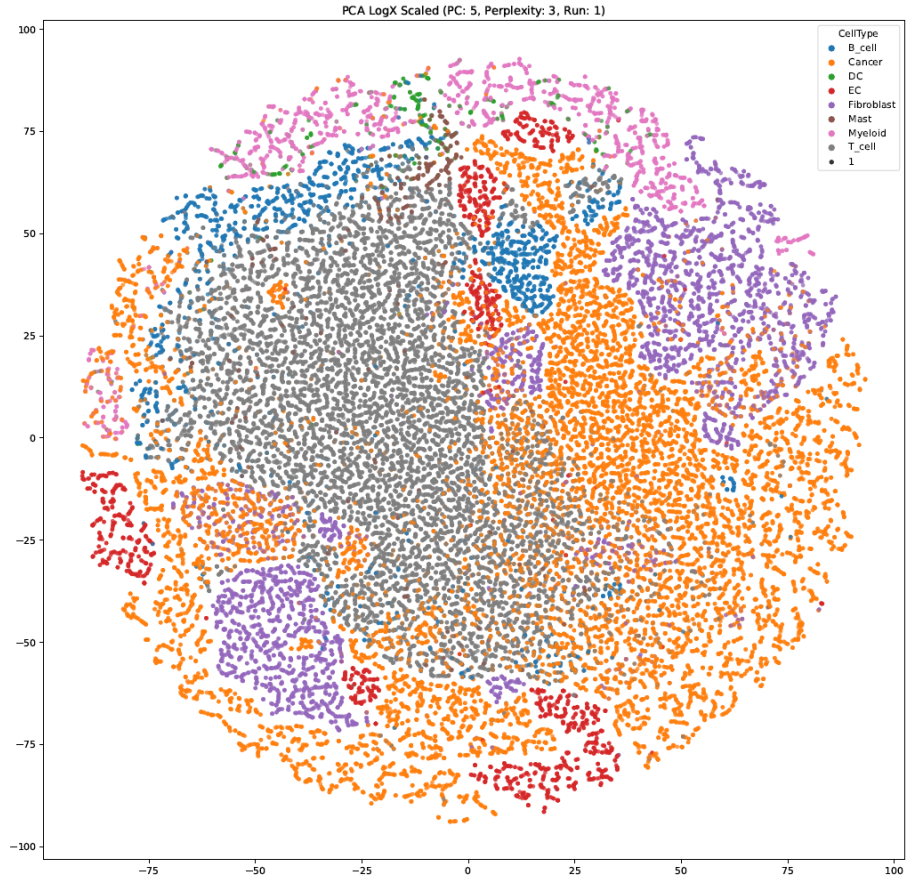
*** Use Poisson in the formulation, but negative binomial gives better result**

Qian et al., 2020 Preliminary Result

MFCVAE Zero Inflated Negative Binomial (3-tSNE)



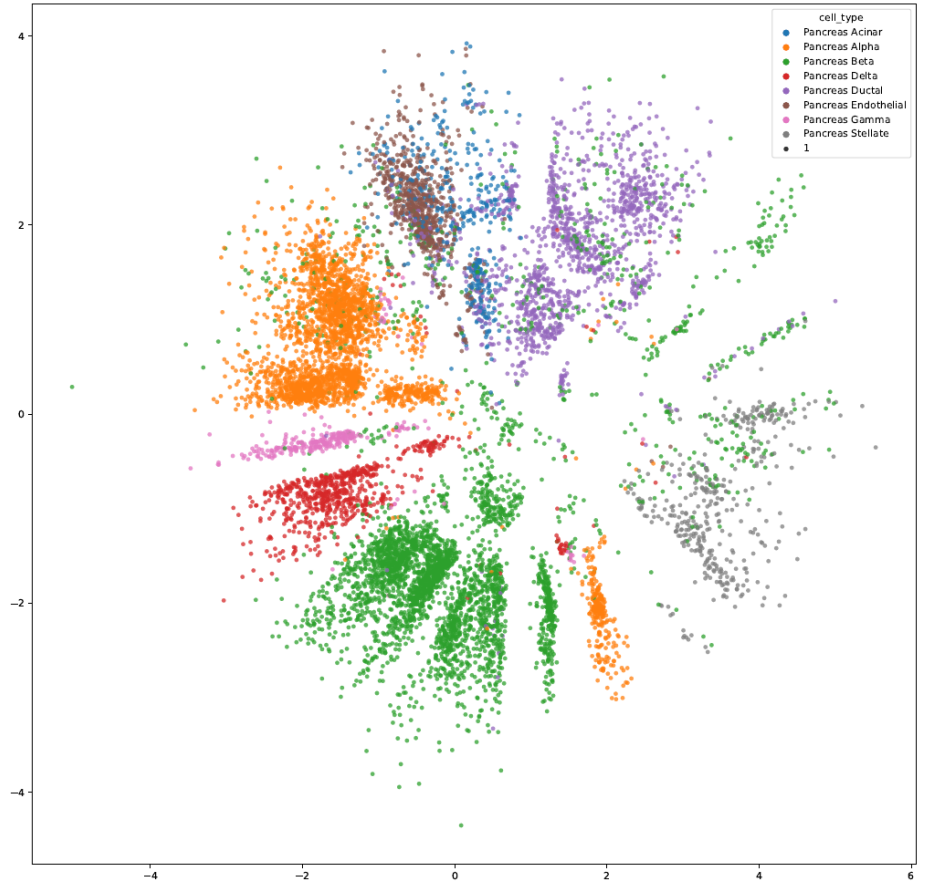
tSNE from Qiang et al., 2020



Pancreas, Preliminary Result (1)

scVI

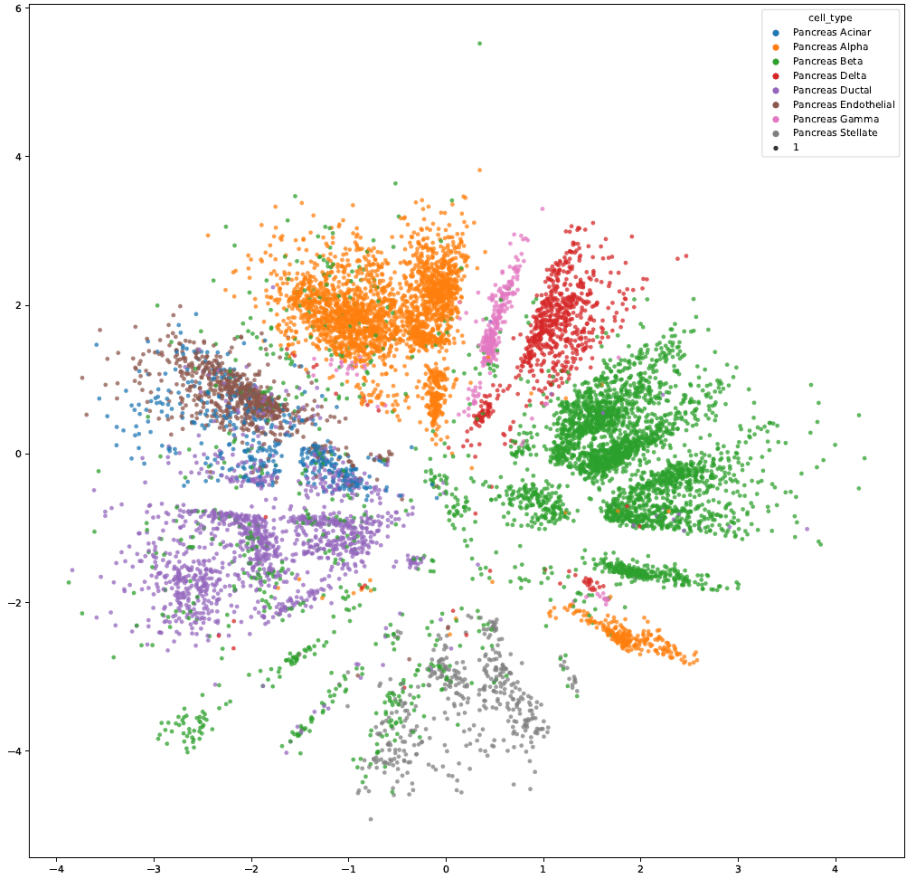
Our Implementation (ZIP)



Pancreas, Preliminary Result (2)

scVI

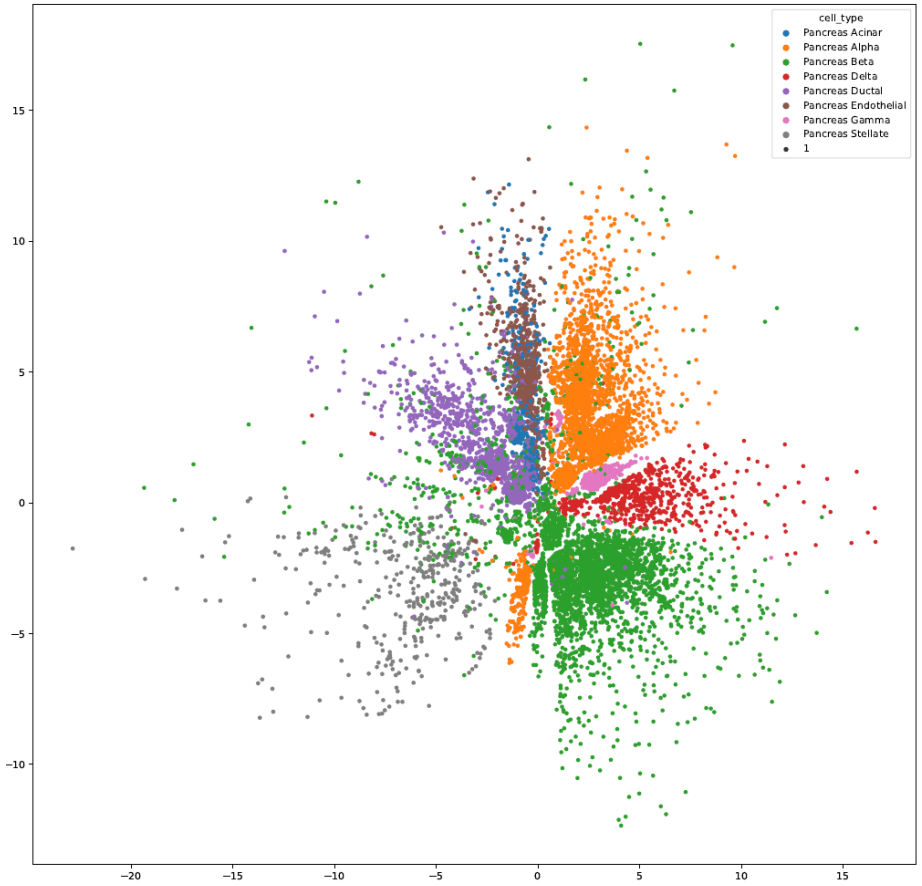
Our Implementation (ZINB)



Pancreas, Preliminary Result (3)

scVI

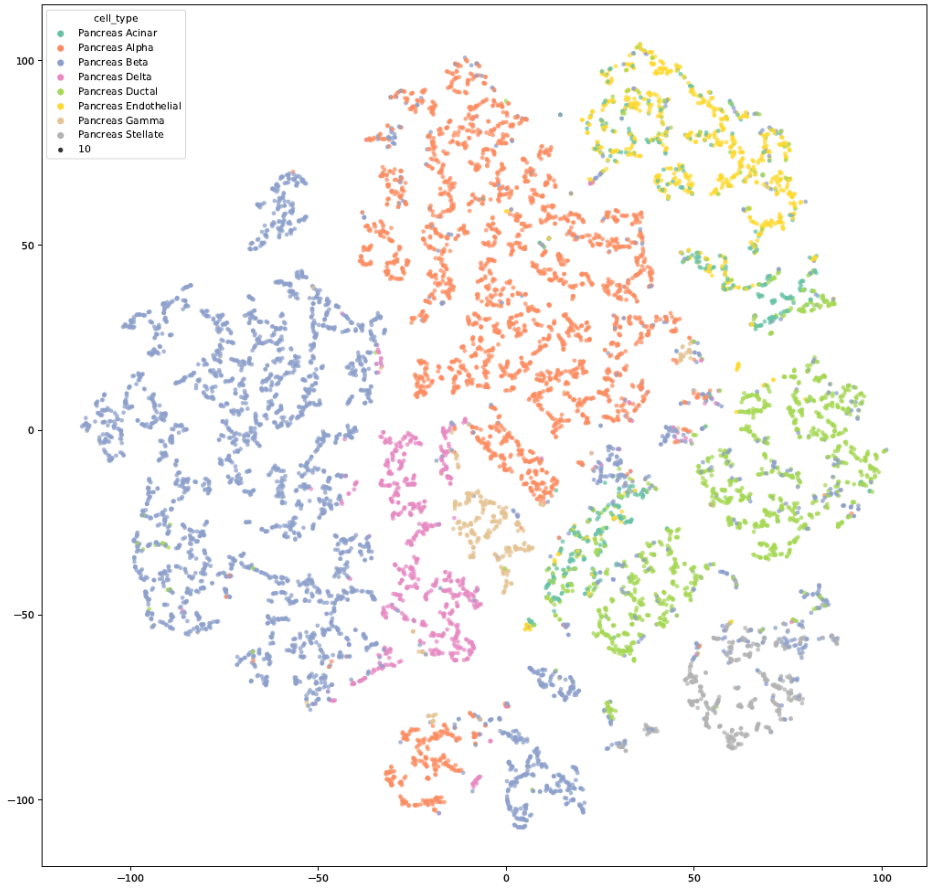
ProMFCVAE (2 Components)



Pancreas, Preliminary Result (4)

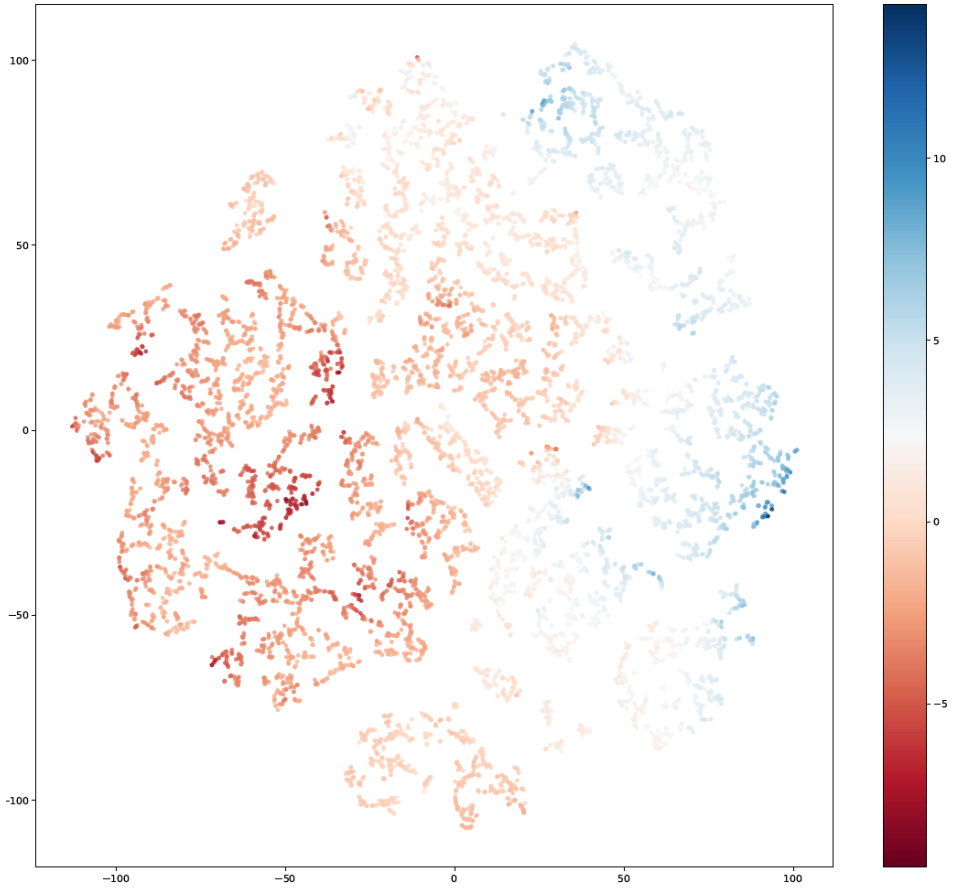
scVI

ProMFCVAE (3 Components, tSNE, Top Hierarchy)

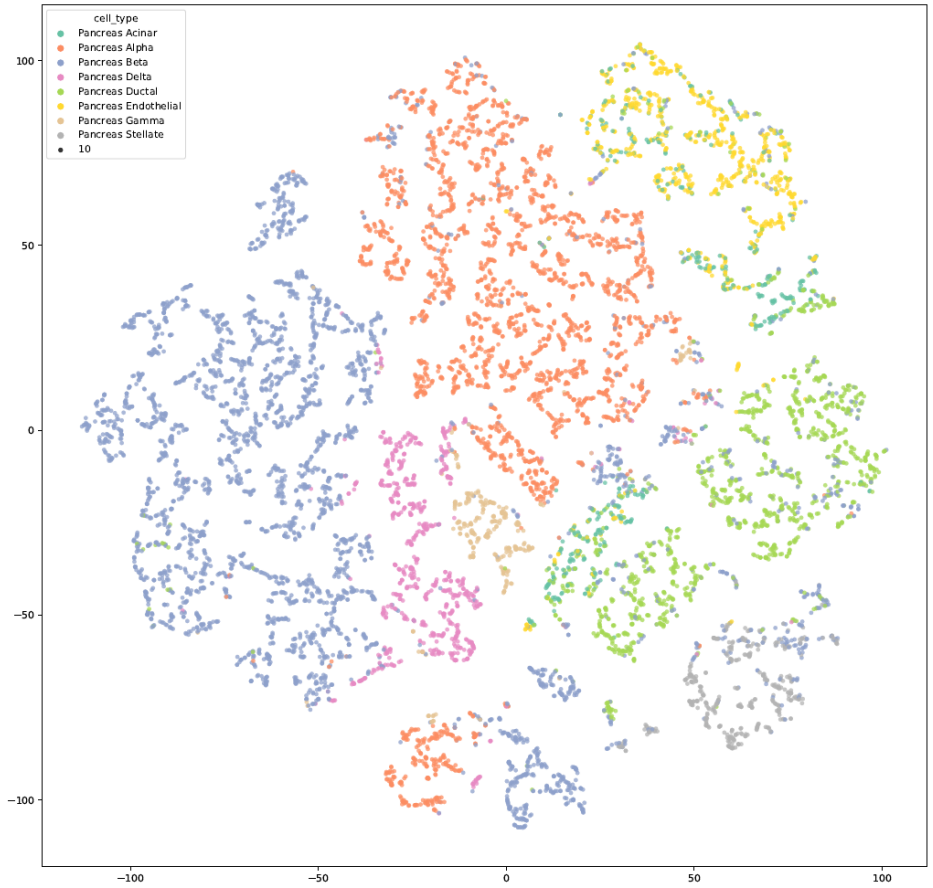


Pancreas, Preliminary Result (5)

Latent Representation H2Z1



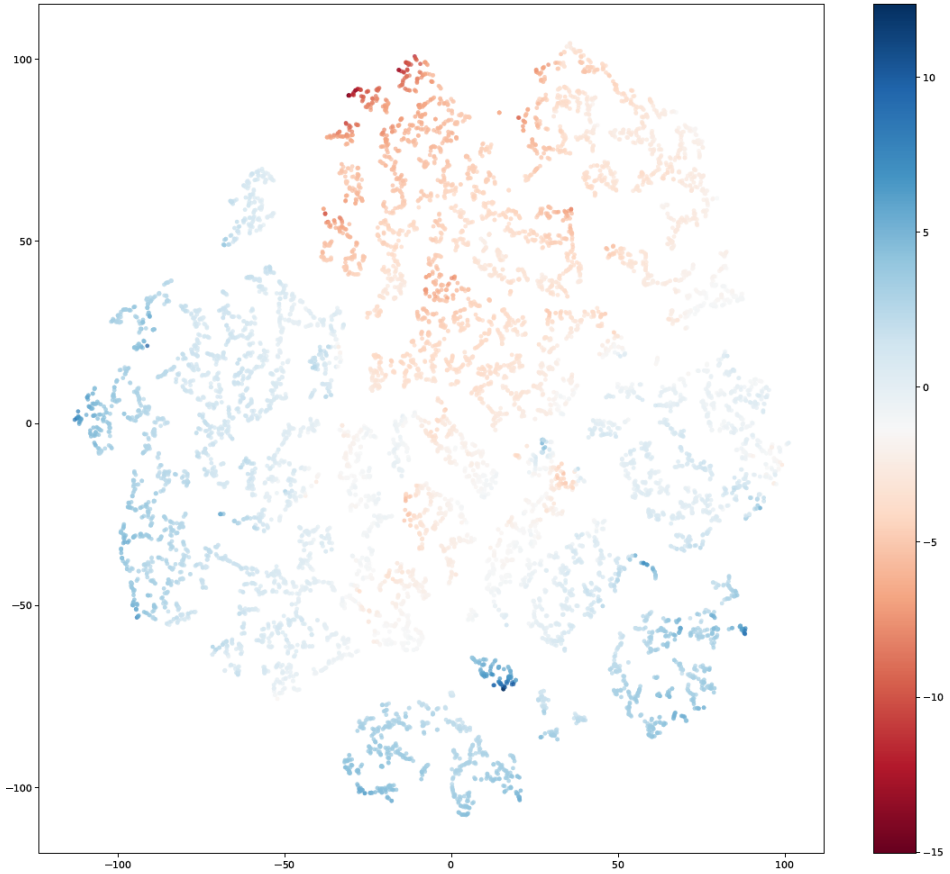
ProMFCVAE (3 Components, tSNE, Top Hierarchy)



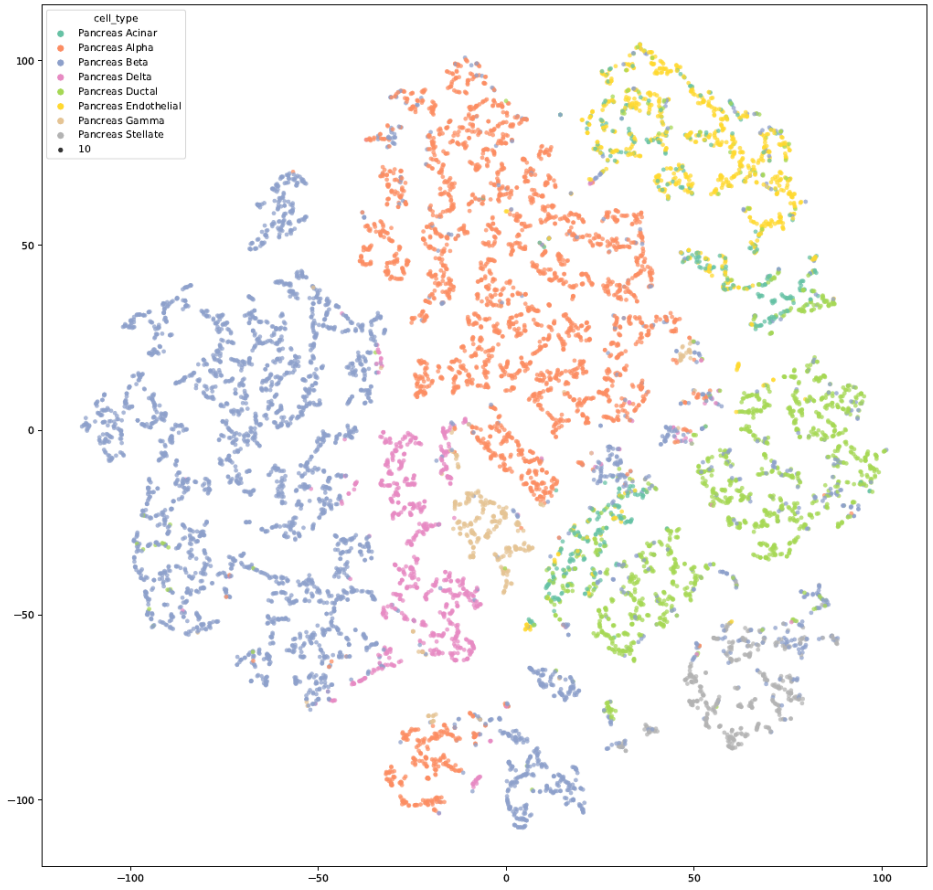
Lower the value, more likely to be Pancreas Beta

Pancreas, Preliminary Result (6)

Latent Representation H2Z2



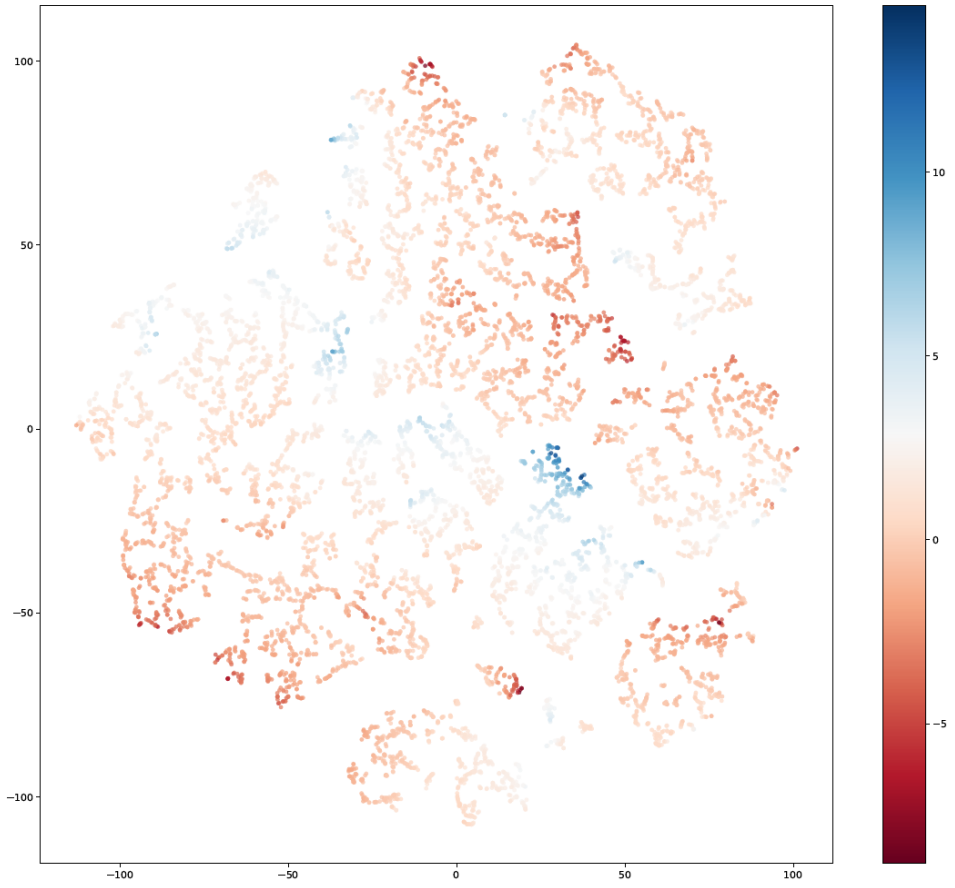
ProMFCVAE (3 Components, tSNE, Top Hierarchy)



Lower the value, more likely to be Pancreas Alpha

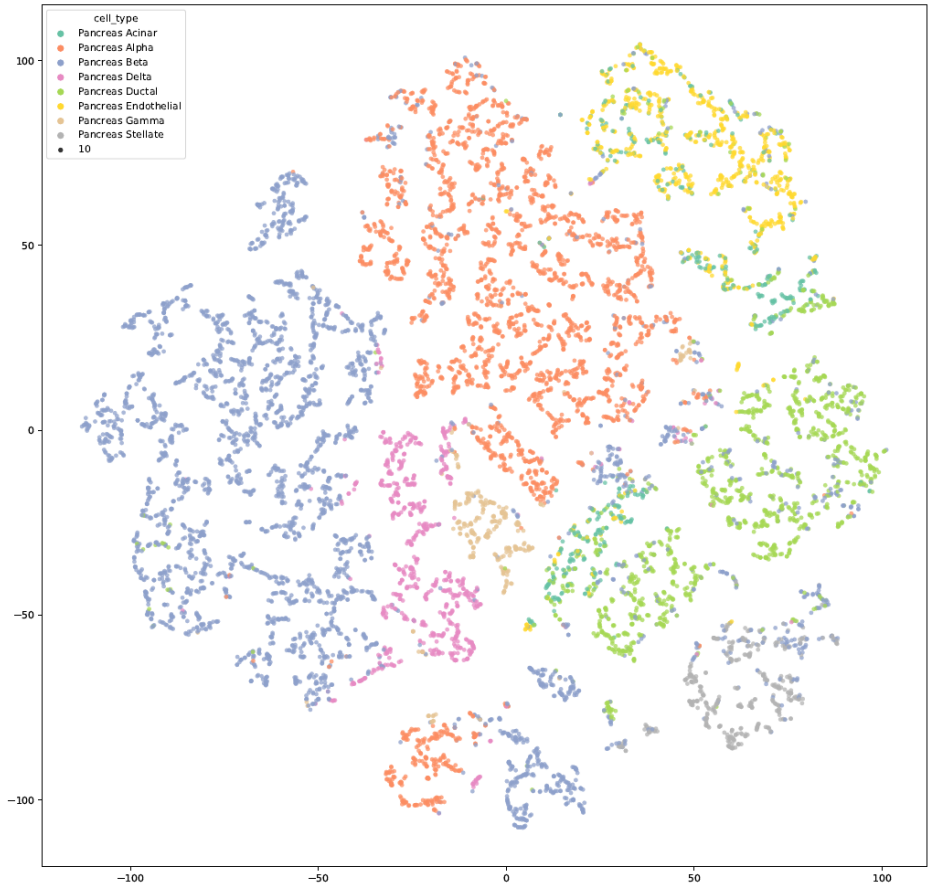
Pancreas, Preliminary Result (7)

Latent Representation H2Z3



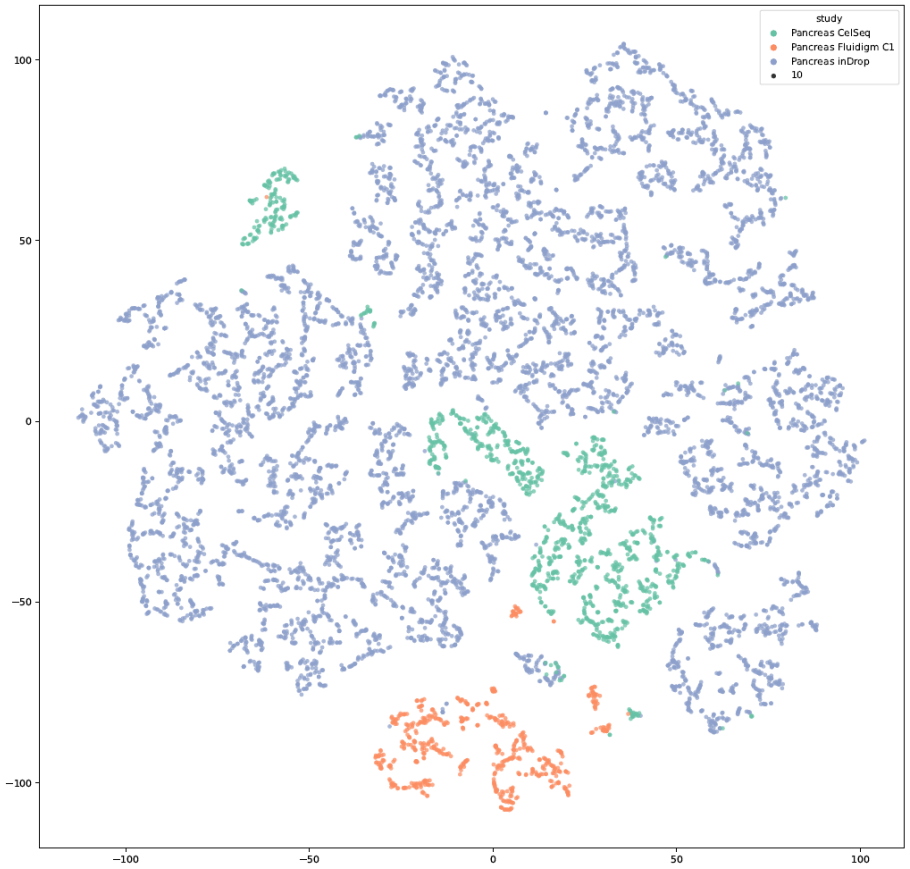
No obvious meaning

ProMFCVAE (3 Components, tSNE, Top Hierarchy)



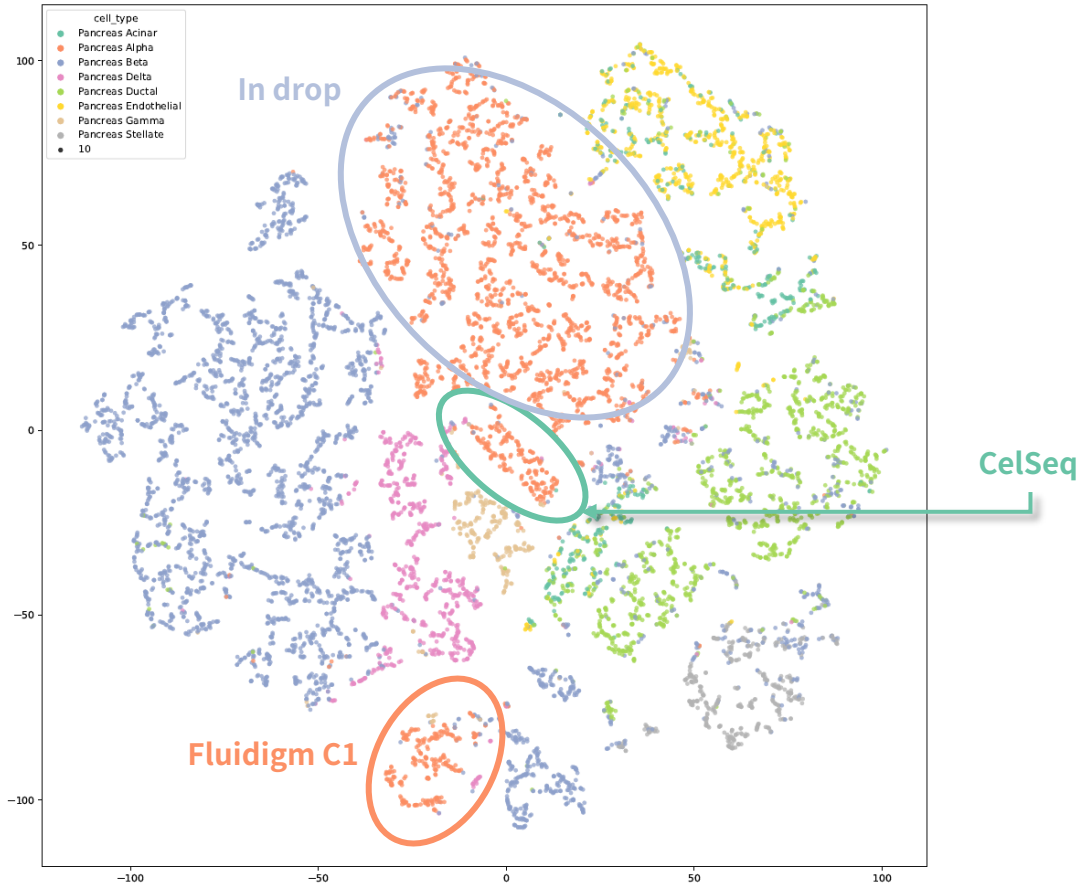
Pancreas, Preliminary Result (8)

scVI



Pancreas Alpha cells have 3 batches

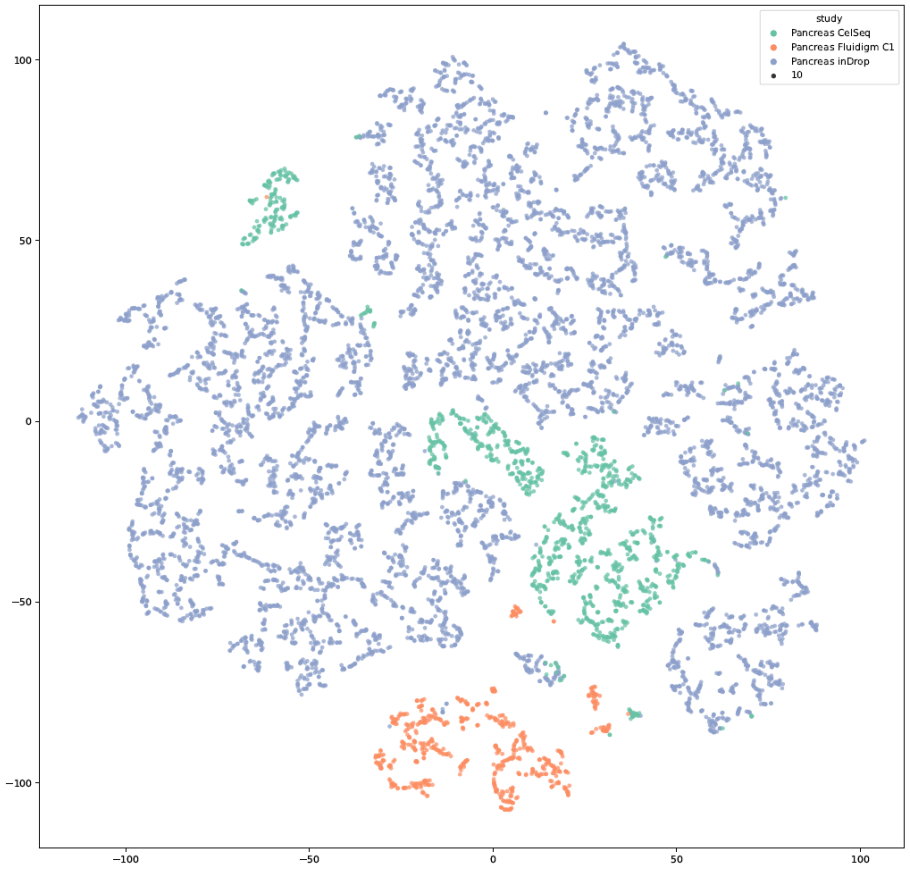
ProMFCVAE (3 Components, tSNE, Top Hierarchy)



The model still encode the batch information in the highest hierarchy
Batch information is expected to be encoded in the lowest hierarchy

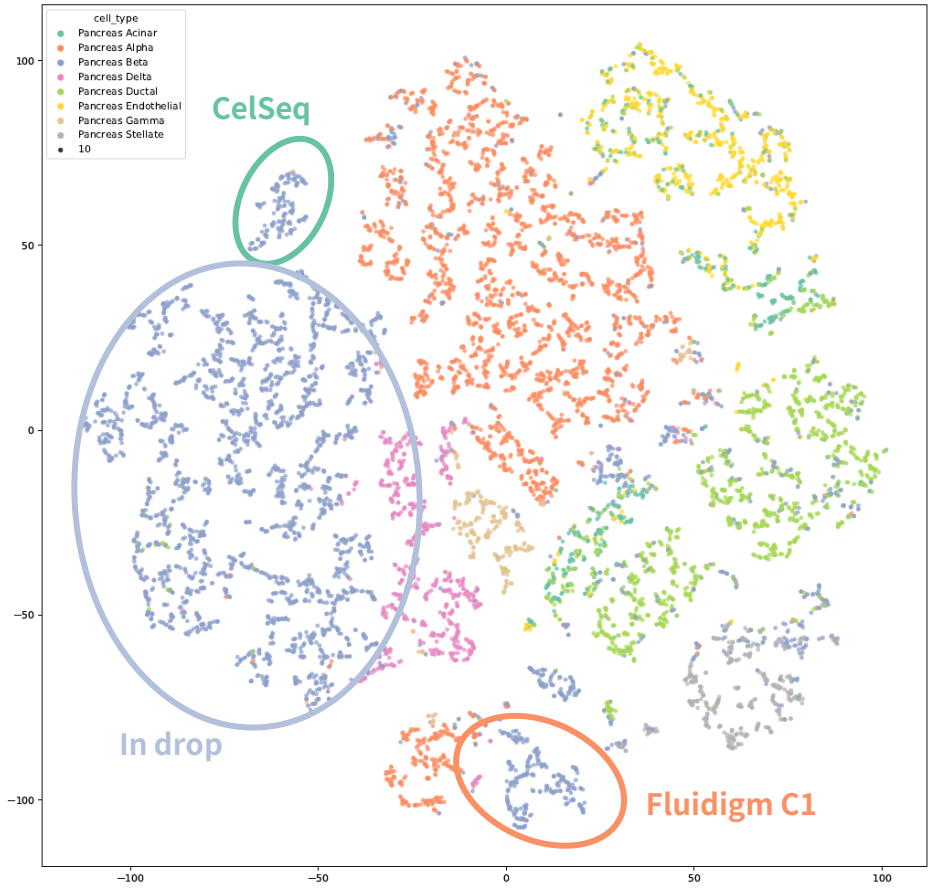
Pancreas, Preliminary Result (9)

scVI



Pancreas Beta cells have 3 batches

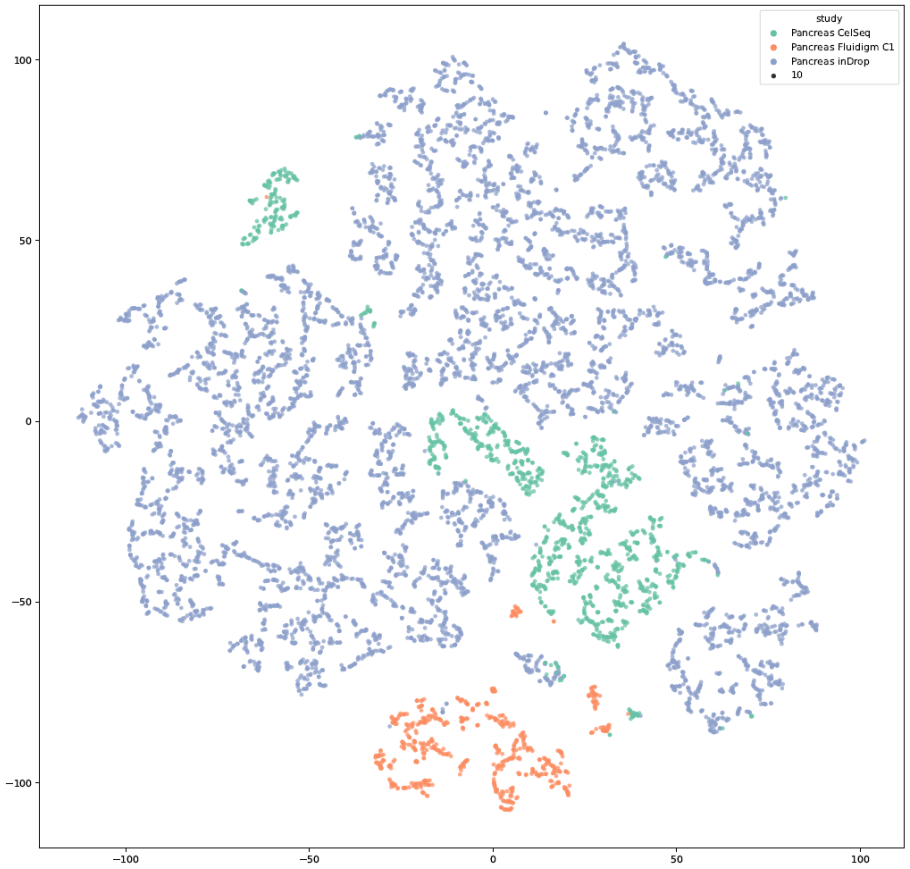
ProMFCVAE (3 Components, tSNE, Top Hierarchy)



The model still encode the batch information in the highest hierarchy
Batch information is expected to be encoded in the lowest hierarchy

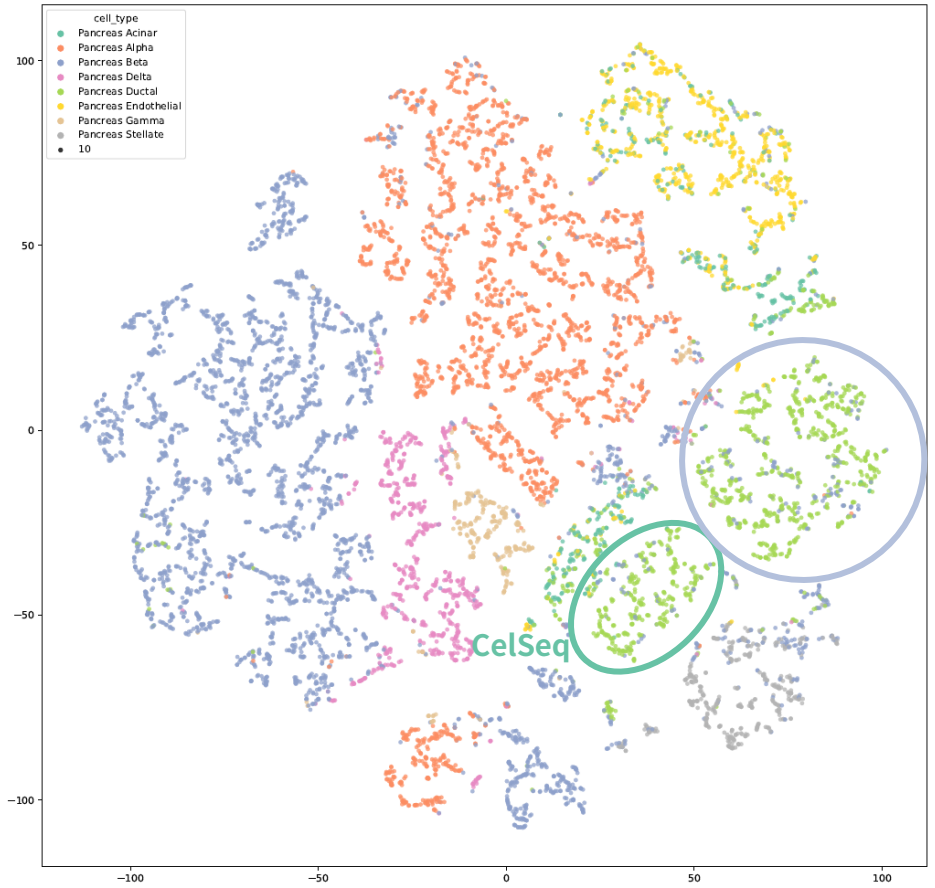
Pancreas, Preliminary Result (10)

scVI



Pancreas Ductal cells have 2 batches

ProMFCVAE (3 Components, tSNE, Top Hierarchy)

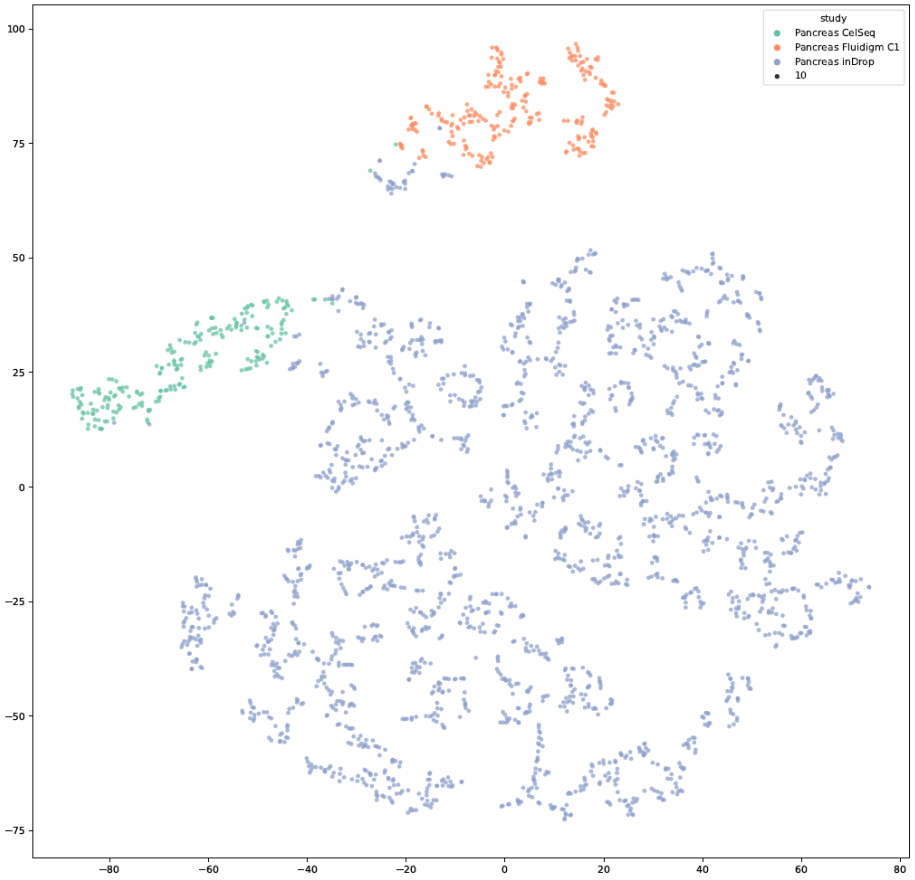


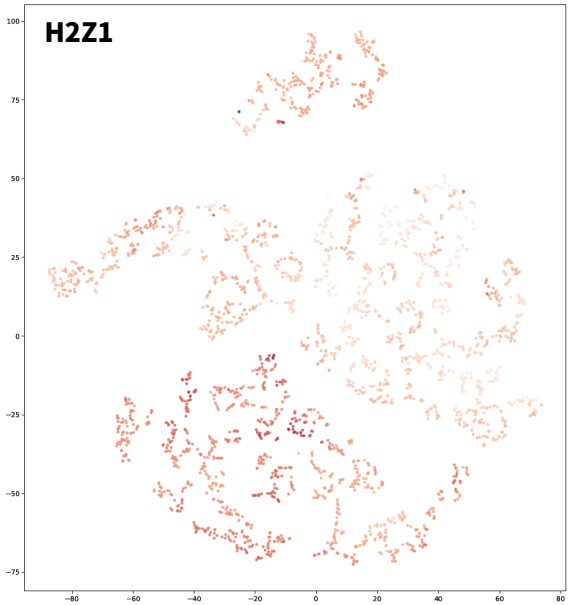
The model still encode the batch information in the highest hierarchy
Batch information is expected to be encoded in the lowest hierarchy

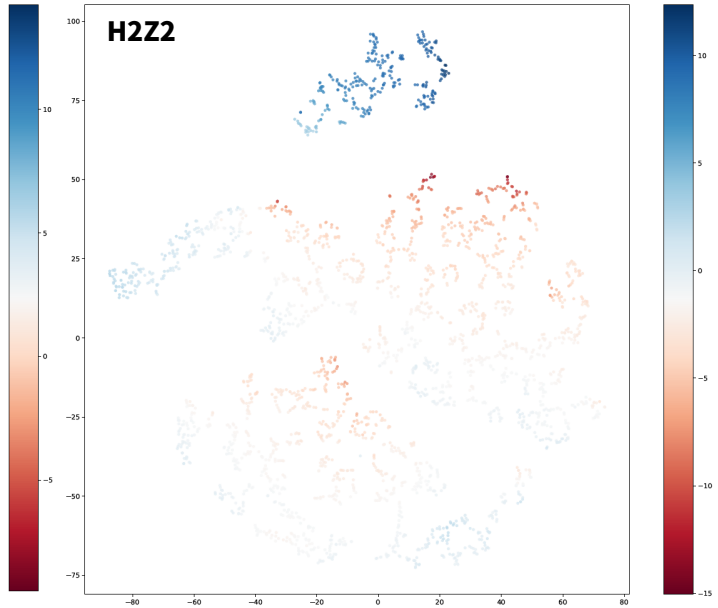
Pancreas, Preliminary Result

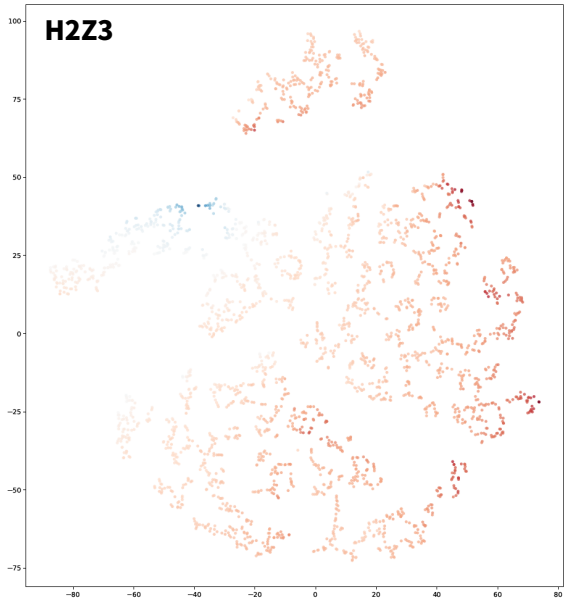
Pancreas Alpha

ProMFCVAE (3 Components, tSNE, Top Hierarchy)









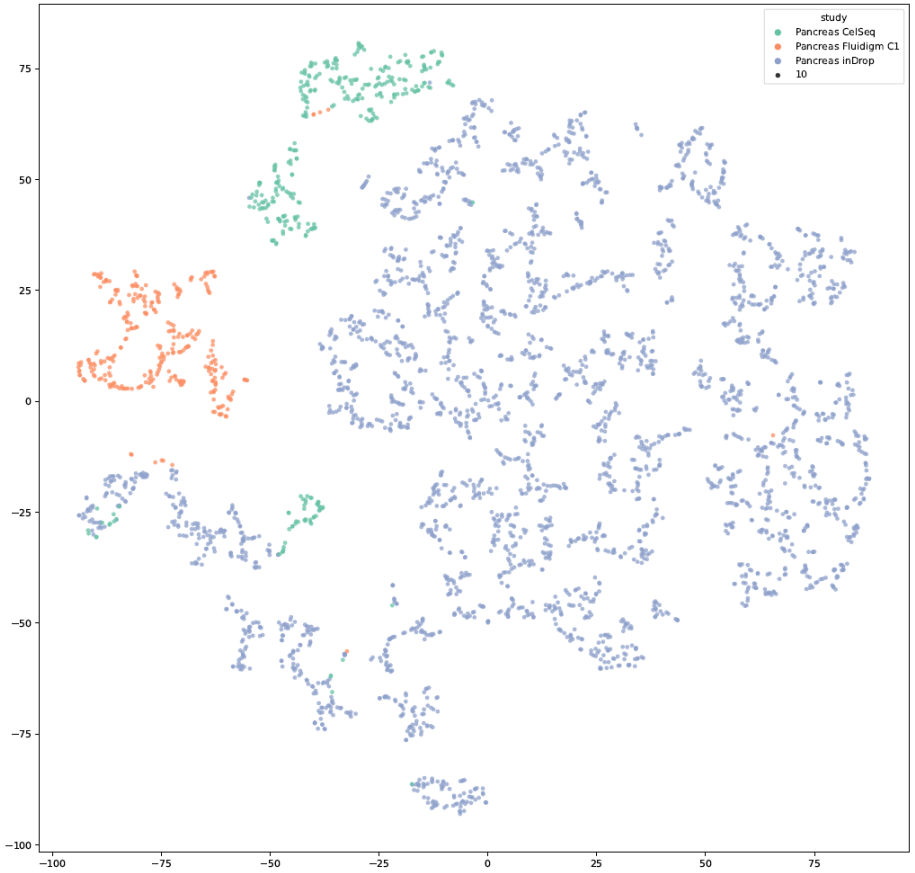
Z2 and Z3 can distinguish Pancreas inDrop

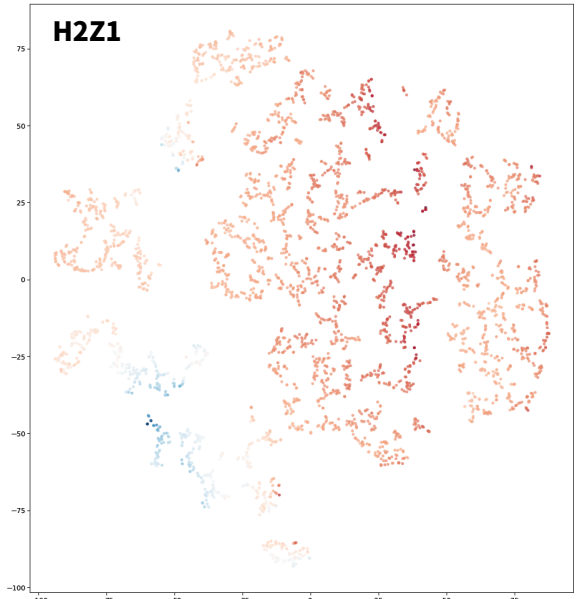
No embedding for CelSeq and Fluidigm C1

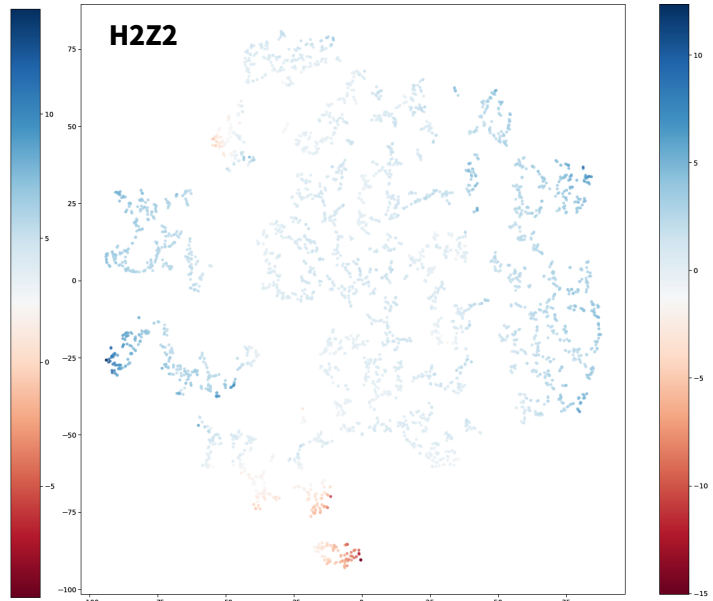
Pancreas, Preliminary Result

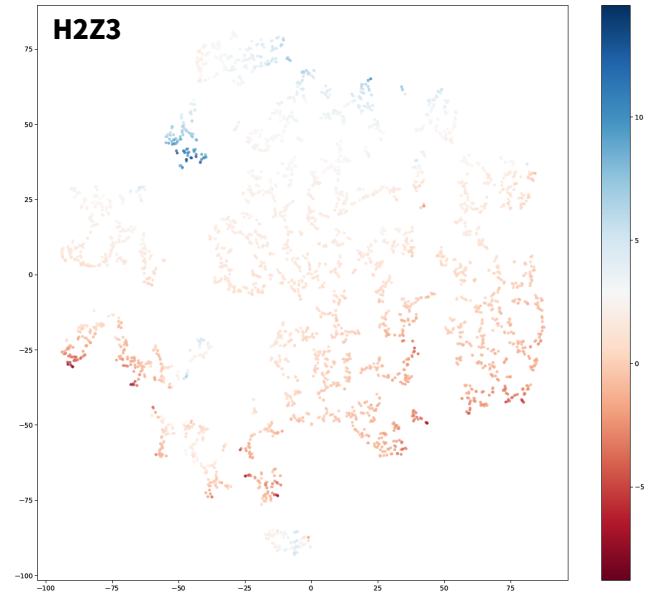
Pancreas Beta

ProMFCVAE (3 Components, tSNE, Top Hierarchy)







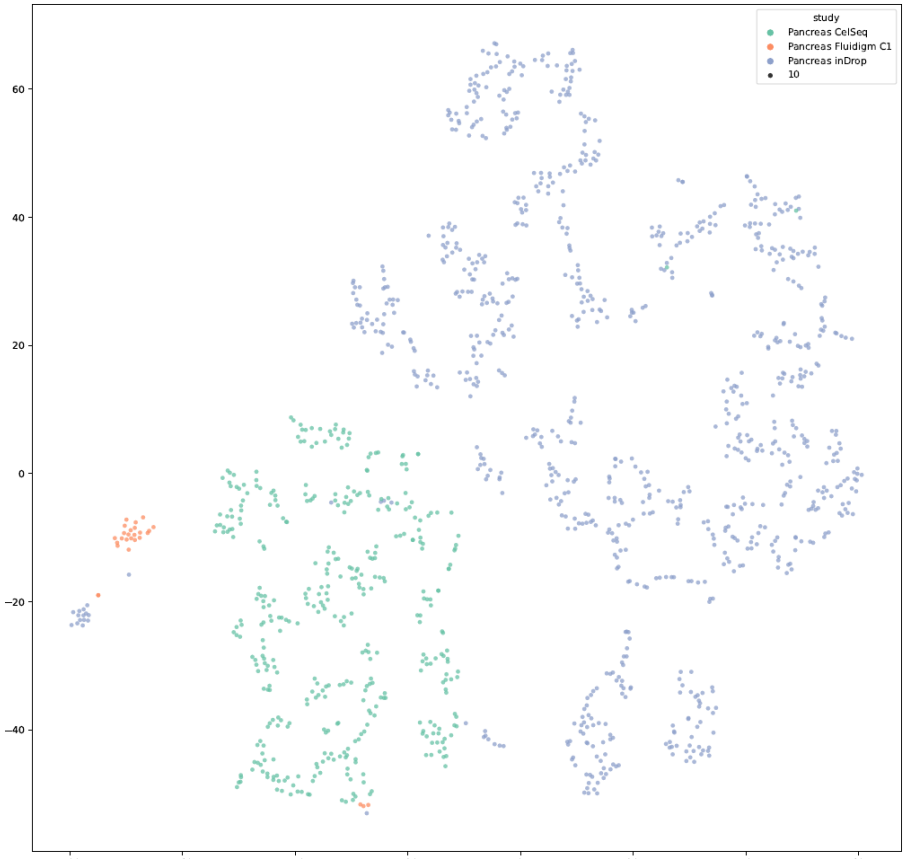


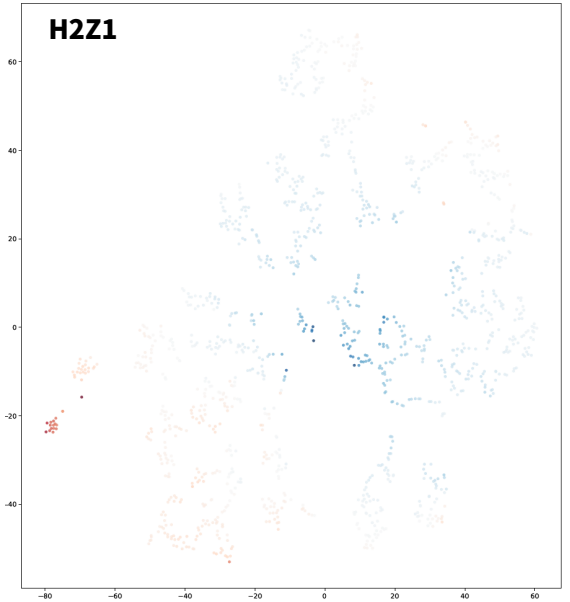
Fluidigm C1 is not embedded

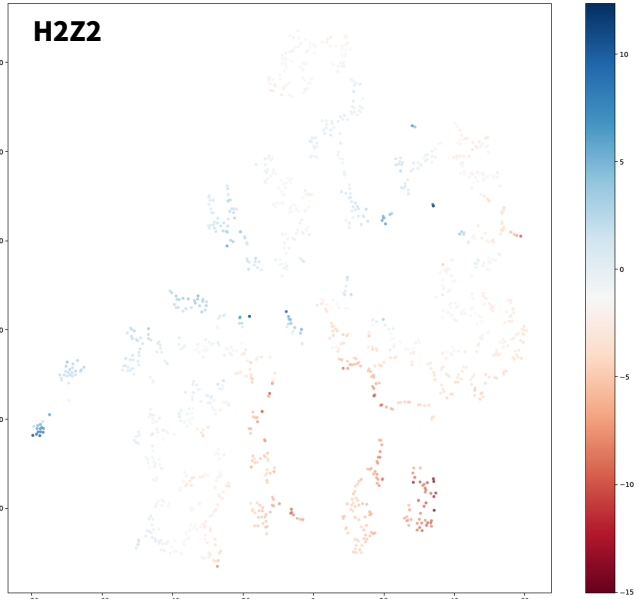
Pancreas, Preliminary Result

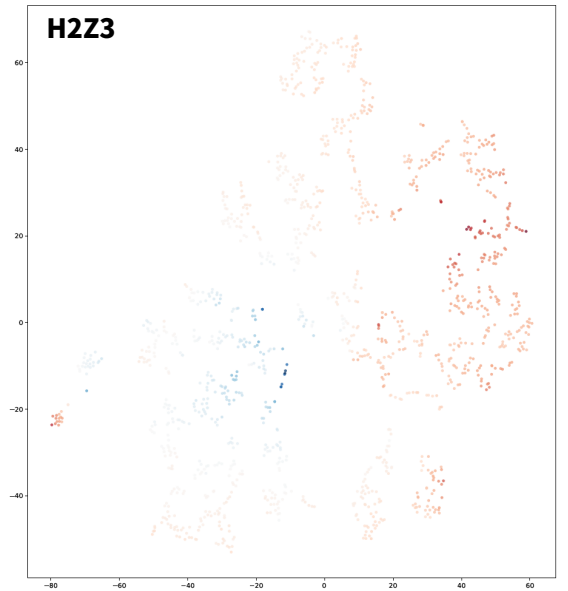
Pancreas Ductal

ProMFCVAE (3 Components, tSNE, Top Hierarchy)









Fluidigm C1 is not embedded

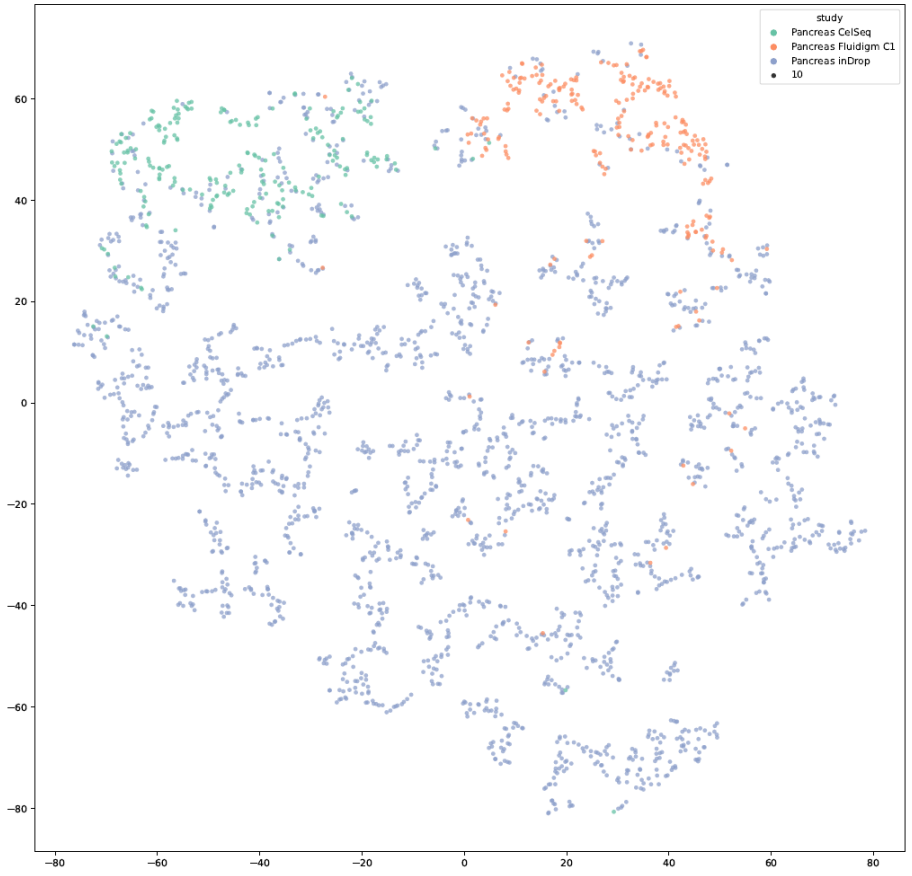
Pancreas, Preliminary Result (11)

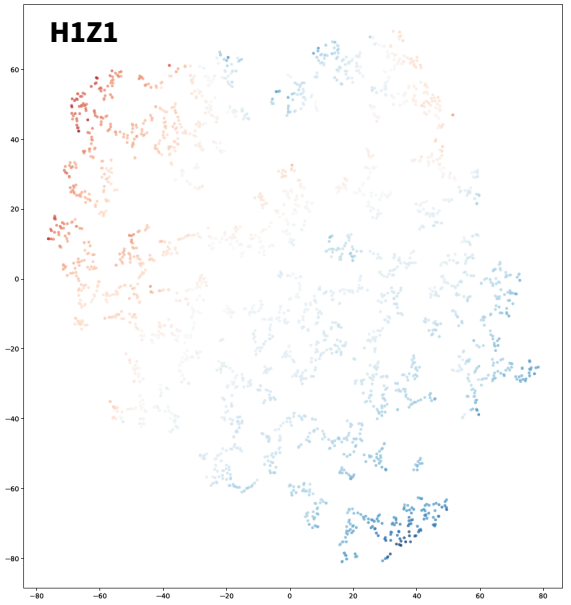
- The batches are not really embedded in the highest hierarchy
  - The latent representation couldn’t be used to explain the batch
  - The latent representation can only be used to explain the cell types
- The reason why It seems like it’s embedded is because there are confounding factors
  - Some cell types are more representative in some study

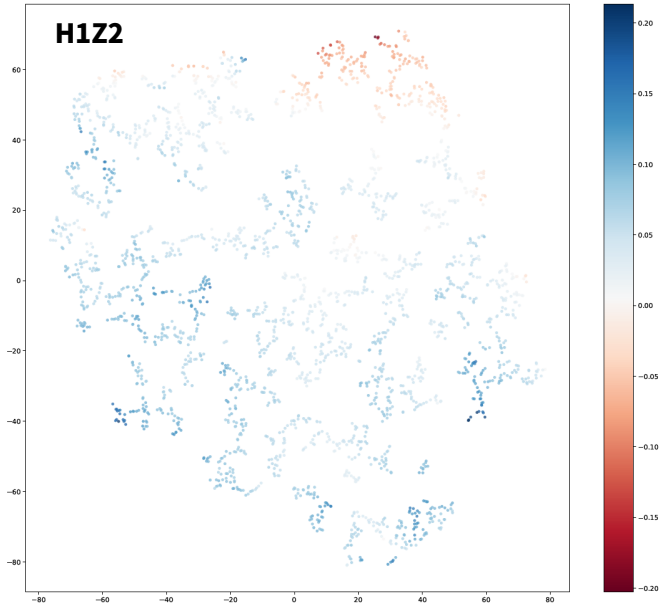
Pancreas, Preliminary Result (4)

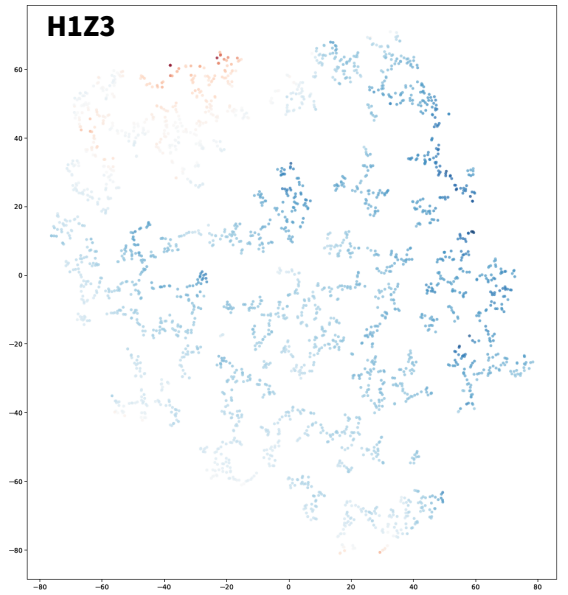
Pancreas Alpha

ProMFCVAE (3 Components, tSNE, Bottom Hierarchy)









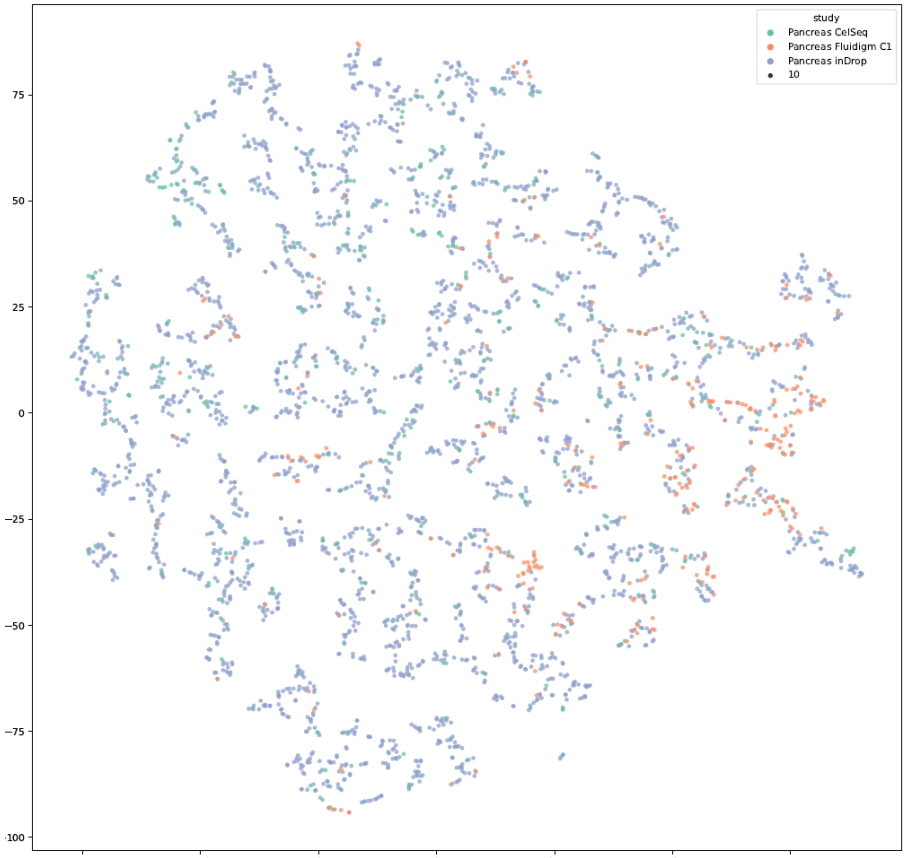
Z1 and Z3 embed Fluidm C1
*** not disentangled**

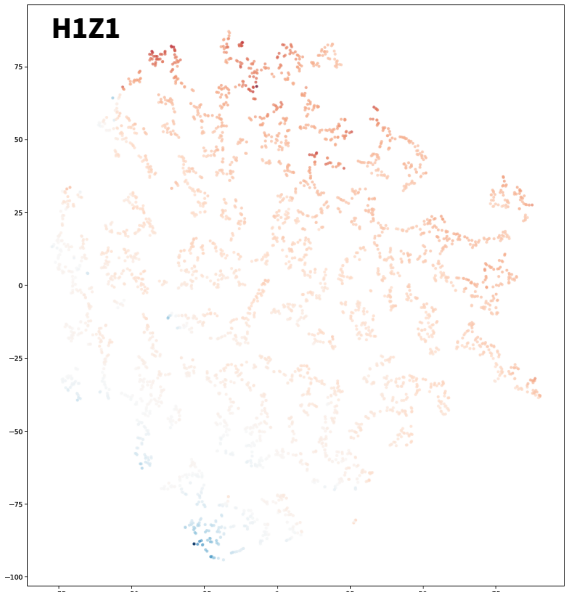
Z2 embed inDrop

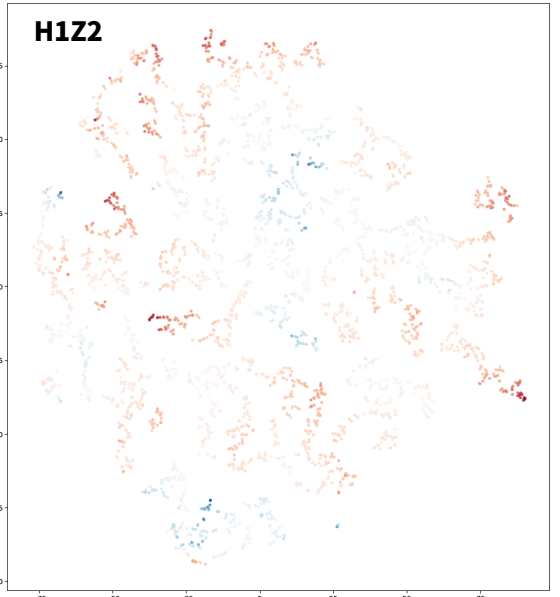
Pancreas, Preliminary Result (4)

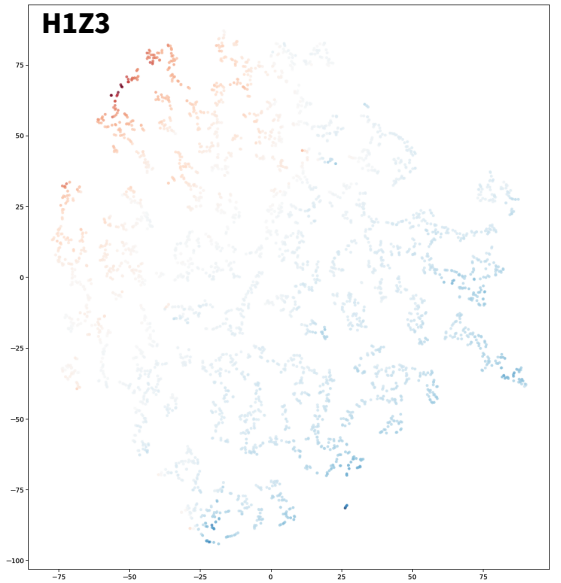
Pancreas Beta

ProMFCVAE (3 Components, tSNE, Bottom Hierarchy)







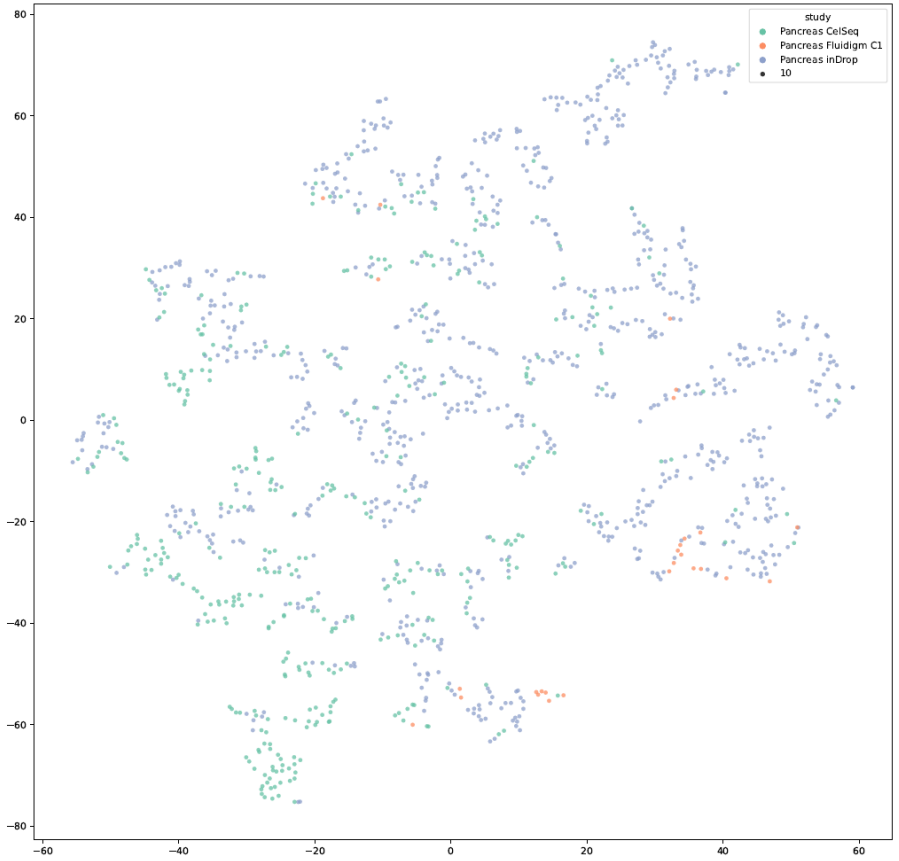


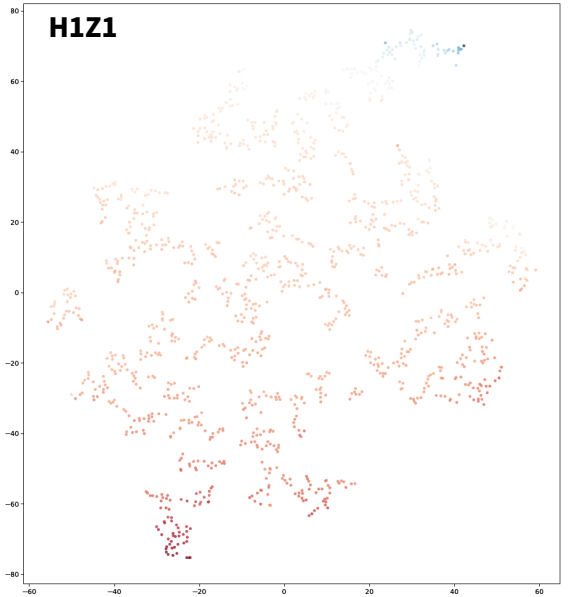
Not embedding anything

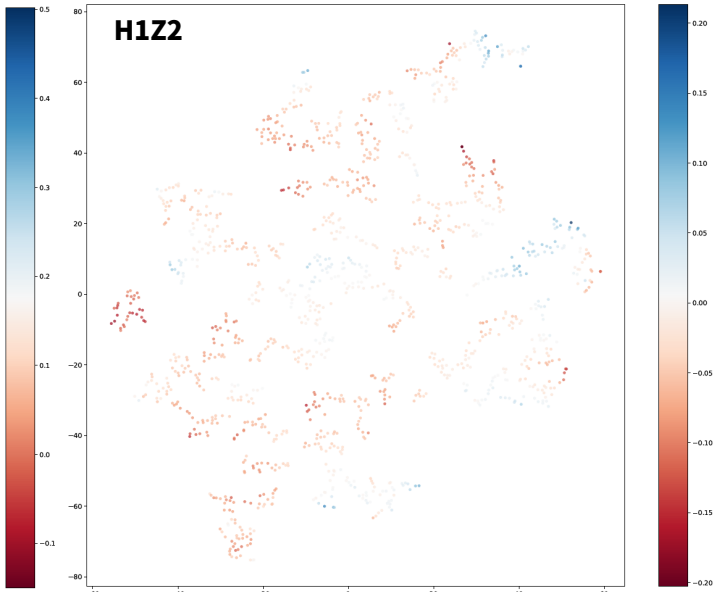
Pancreas, Preliminary Result (4)

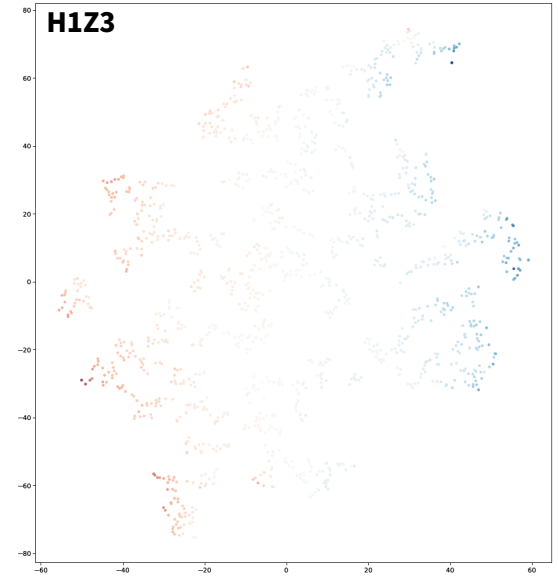
Pancreas Ductal

ProMFCVAE (3 Components, tSNE, Bottom Hierarchy)



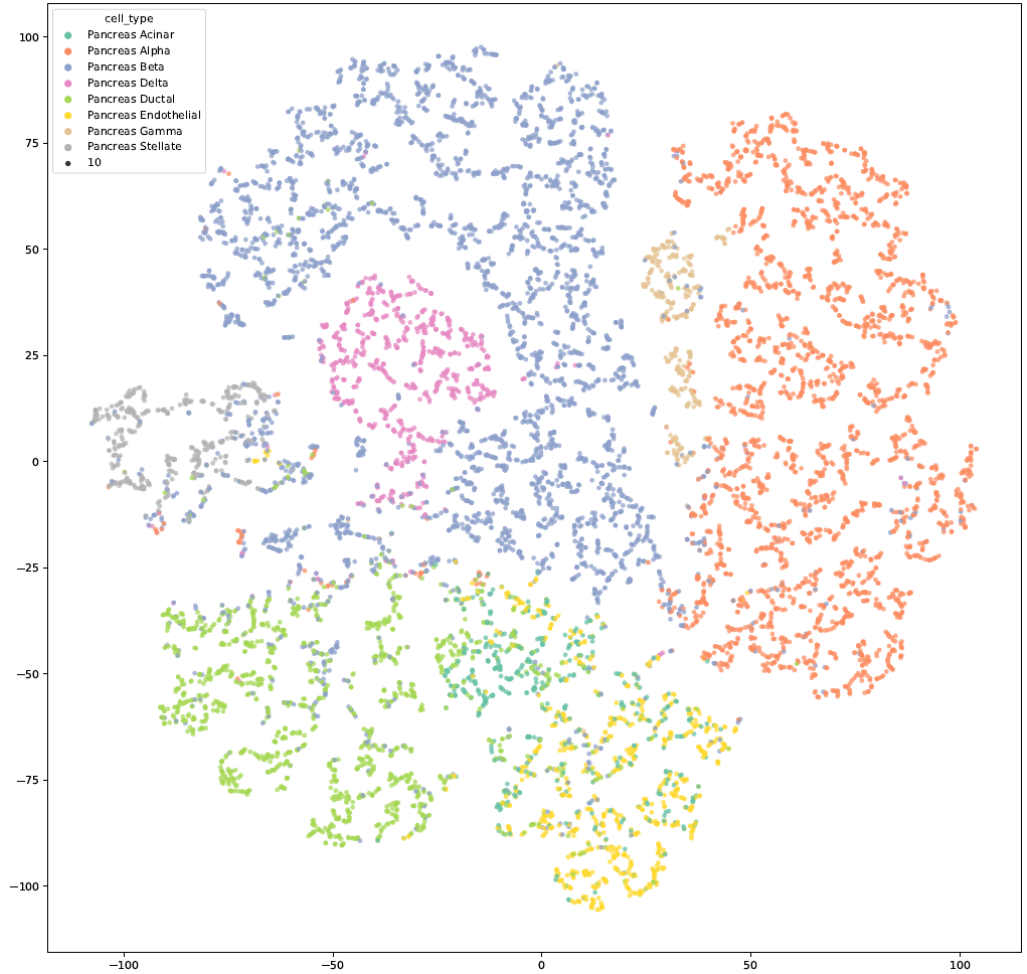


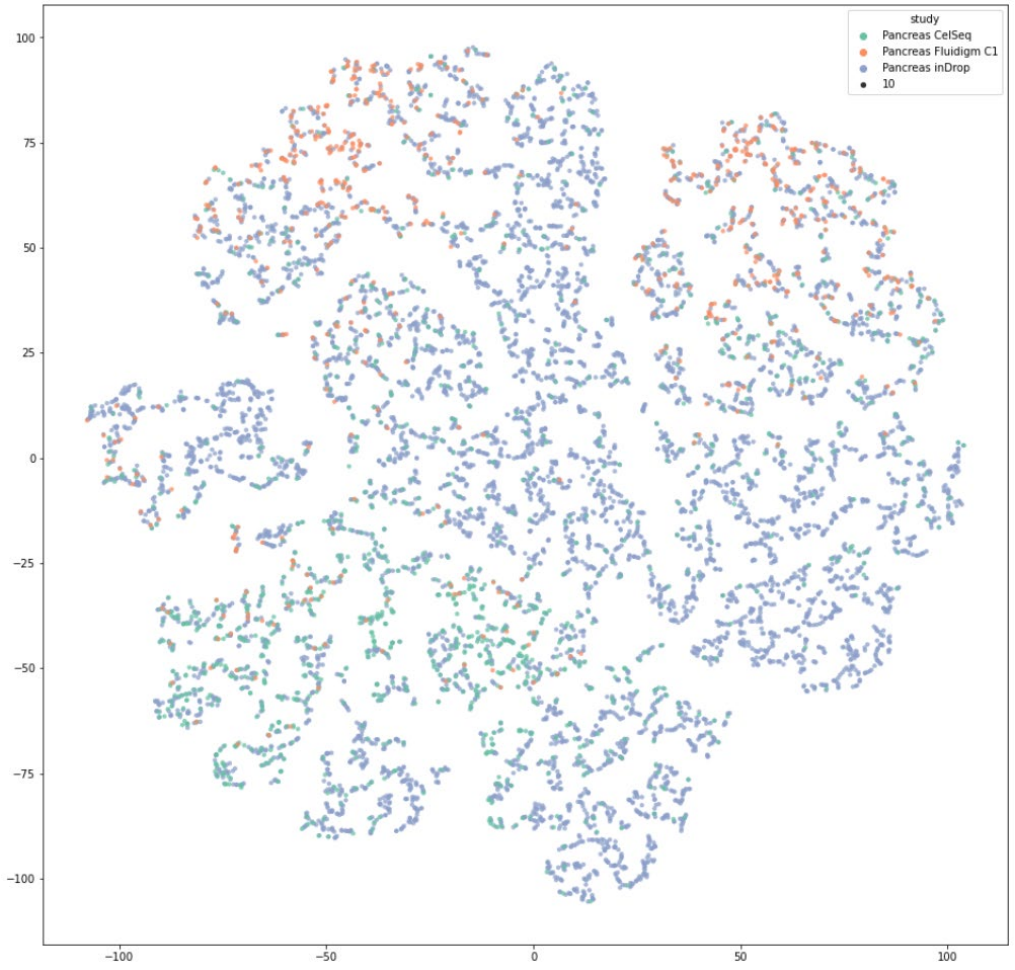




Not embedding anything

Conditioning on Study





Pancreas, Preliminary Result (12)

- Try latent component 2, 3 and 5 (make the model not identifiable), the results are pretty similar.
- The bottom hierarchy only embed the batch information for Pancreas Alpha.
- We could not conclude what information is embedded using the bottom hierarchy for Pancreas Beta and Pancreas Ductal.

- The abstract hierarchy is still too powerful.
- Might need to try fixing the clusters.

Needed Functions

- Done
  - Infer library-size corrected rate for gene expression
  - Infer dropout rate
  - Generate new samples
  - Variable number of latent dimensions and component for each facet
  - Clustering
    - Probability for each cluster
    - Contingency table for clustering
  - tSNE for latent dimension
    - Color by z value
    - Color by groups
  - Classification (train on one dataset, predict the cell label on another dataset)
  - Conditioning on batch effect
- Todo
  - Fixed facet (e.g. facet 1 needs to encode cell types, facet 2 needs to encode batch)
  - Infer batch corrected rate for gene expression
  - Differential expression

Analysis

- Qian et al., 2020. (Portal)
  - 8 lung cancer patients (93,575 cells)
  - 7 colorectal cancer patients (93,575 cells)
  - 5 ovarian cancer patents (45,114 cells)
  - 14 breast cancer patents (44,024 cells)
- Lotfollahi et al., 2021. (Portal)
  - Pancreas
  - Mouse Brain
  - Tabula Senis Muris
  - Mouse Cell Atlas
  - Panorama
  - Covid-19

Discussion and Future Works

Deep Generative Models

Variational Autoencoder

Meaningful Representation

Multi-Facet Clustering Variational Autoencoder (MFC-VAE)
Identifiable Variational Autoencoder (iVAE)
Variational Deep Embedding (VaDE)[2]
Slow Variational Autoencoder (SlowVAE)

Identifiable Variational Autoencoder

Neural Decomposition (ND)
Multi-level Variational Autoencoder (MLVAE)
Group Variational Autoencoder (GVAE)

(Weakly supervised) Identifiable Variational Autoencoder

β-Total Correlation Autoencoder (β-TCVAE)[3]
Factor Variational Autoencoder (FactorVAE)[4]
β-Variational Autoencoder (β-VAE)

Disentangled Variational Autoencoder

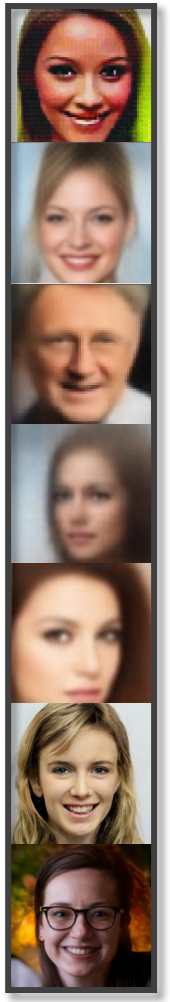
Variational Autoencoder

Vector Quantized Variational Autoencoders (VQ-VAE)[5]
Introspective Variational Autoencoder (IntroVAE)

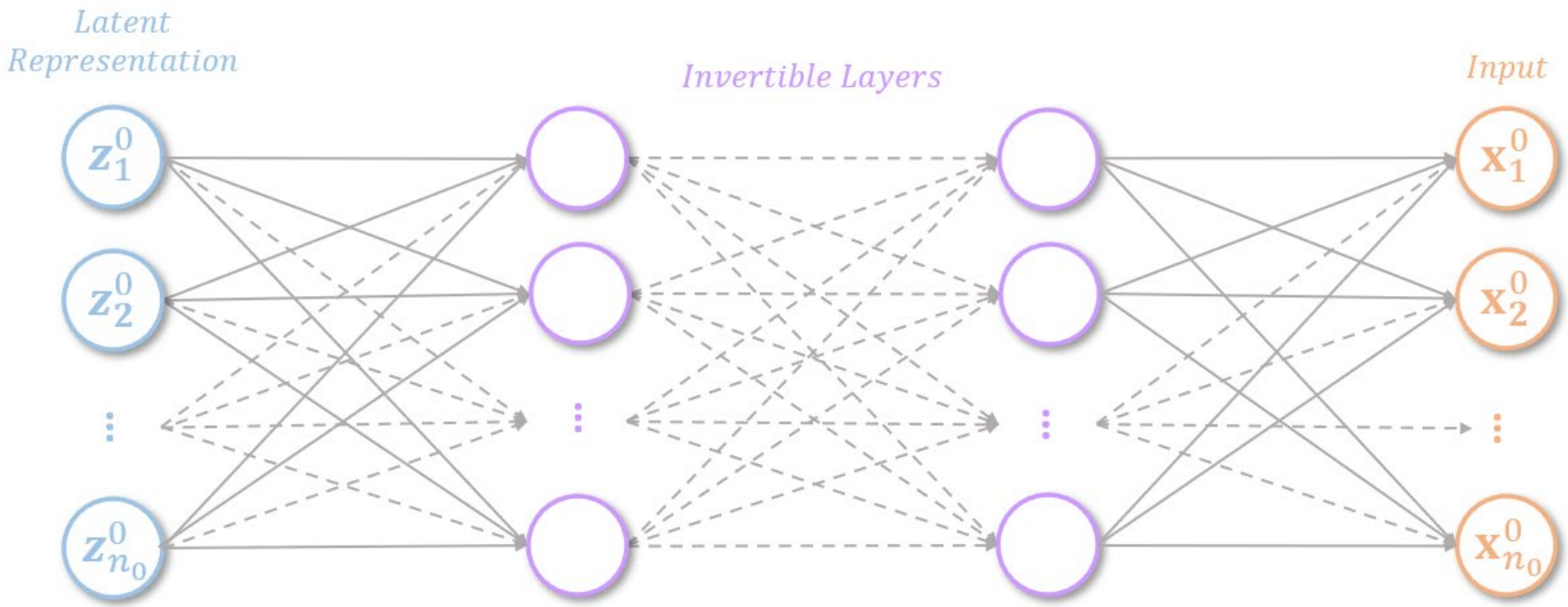
Autoregressive/incorporate GAN

Generative Adversarial Network[6]

Better Reconstruction



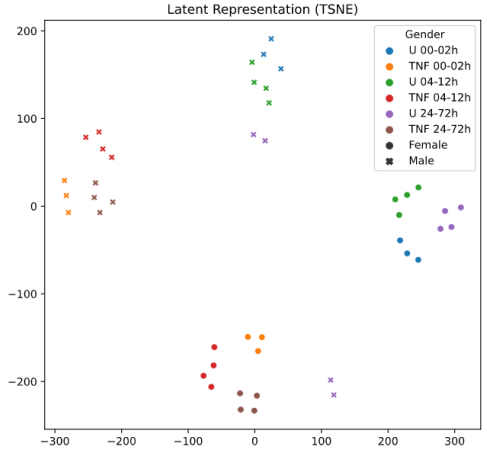
Normalizing Flow



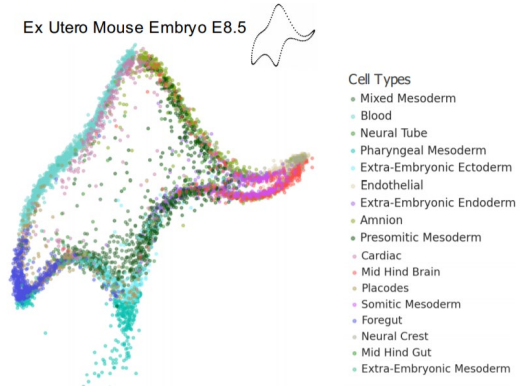
- Directly approximate the posterior $p(x) = p(z)|\det J(z)|^{-1}$
- Cannot perform dimension reduction (without some modification)
- Suitable for time-series data (autoregressive flow)

Discussion and Future Works

- Conclusions and Discussions
  - Be cautious about the selected model as it will affect the conclusion we drawn.
  - What is the inductive bias for application in bioinformatics (e.g. scRNA-Seq).
  - What would be an optimal way to design our computational validation as the cost to perform large-scale experiment on regulation is still demanding.
  - What is the benefit of estimating posterior directly
  - What is the benefit of having homeomorphic transformtion.



- Future Works
  - Recent study proposed a normalizing flow architecture that can flow across dimension.
    - Allow exact likelihood estimation (instead of ELBO) and dimension reduction at the same time.
    - Can we combine the idea of normalizing flow with VaDE (clustering along with dimension reduction)
  - Evaluation of different VAEs on bioinformatics (scRNA-Seq).
    - Similar to image generation on dSprites/Celeb, can we design a metrics to benchmark the performance of our model?



Chari et al., arXiv 2021

Interesting Publications - VAEs

- 2014-ICLR-Auto-Encoding Variational Bayes
- 2015-ICML-MADE-Masked Autoencoder for Distribution Estimation
- 2015-NeurIPS-Learning Structured Output Representation using Deep Conditional Generative Models
- 2016-NeurIPS-Ladder Variational Autoencoders
- 2017-ICLR-β-VAE Learning Basic Visual Concepts with a Constrained Variational Framework
- 2017-IJCAI-Variational Deep Embedding An Unsupervised and Generative Approach to Clustering
- 2017-NeurIPS-Understanding Disentangling in β-VAE
- 2017-PMLR-Learning Hierarchical Features from Generative Models
- 2018-AAAI-Multi-Level Variational Autoencoder Learning Disentangled Representations
- 2018-NeurIPS-IntroVAE Introspective Variational Autoencoders for Photographic Image Synthesis
- 2018-PMLR-Disentangling by Factorising
- 2019-PMLR-Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations
- 2020-AISTATS-Neural Decomposition Functional ANOVA with Variational Autoencoders
- 2020-AISTATS-Variational Autoencoders and Nonlinear ICA-A Unifying Framework
- 2020-ICLR-Progressive Learning and Disentanglement of Hierarchical Representations
- 2020-NeurIPS-NVAE A Deep Hierarchical Variational Autoencoder
- 2021-arXiv-Demystifying Inductive Biases for β-VAE Based Architectures
- 2021-arXiv-I Don’ t Need u Identifiable Non-Linear ICA Without Side Information
- 2021-arXiv-Multi-Facet Clustering Variational Autoencoders

Interesting Publications – Normalizing Flow

- 2014-ICLR-NICE Non-linear Independent Components Estimation
- 2015-ICML-Variational Inference with Normalizing Flows
- 2016-NeurIPS-Improving Variational Inference with Inverse Autoregressive Flow
- 2017-NeurIPS-Masked Autoregressive Flow for Density Estimation
- 2018-arXiv-f-VAEs Improve VAEs with Conditional Flows
- 2018-arXiv-Glow-Generative Flow with Invertible 1x1 Convolutions
- 2018-arXiv-WaveGlow-A Flow-based Generative Network for Speech Synthesis
- 2020-arXiv-Normalizing Flows Across Dimensions
- 2020-ICLR-Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN)
- 2020-NeurIPS-SurVAE Flows Surjections to Bridge the Gap between VAEs and Flows

Thanks