Dimension Reduction

SCVIS – A VAE-based approach

Speaker: Jeff

Introduction

- Paper: Interpretable Dimensionality Reduction of Single Cell Transcriptome Data with Deep Generative Models.

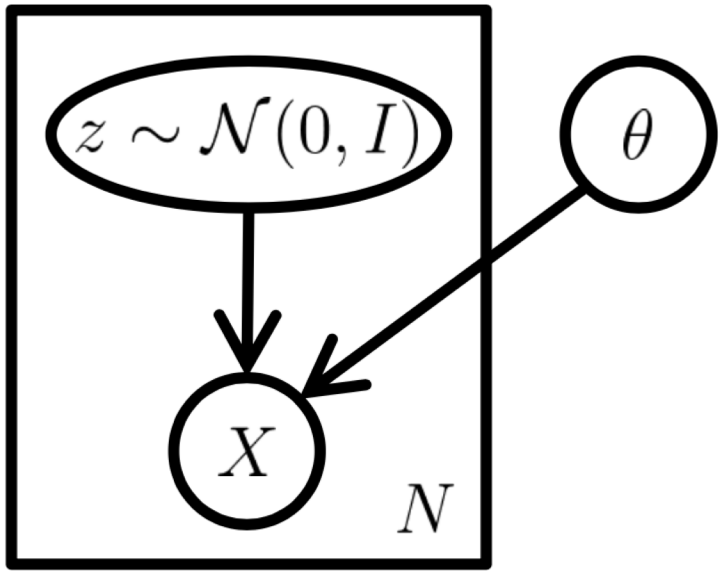
Variational Autoencoder (VAE)

- VAE is a kind of generation model. (Generating real data.)
- $\boldsymbol{X_n}$: Observed data, samples, e.g. MNIST handwritten digits images, gene expression of cells
- $\boldsymbol{Z_n}$: Latent variable, e.g. Digits, Cell type
- Concept:
  - For every datapoint $(\boldsymbol{x_n})$ in the dataset, there is one or many settings of the latent variables $(\boldsymbol{z_n})$ which causes the model $f(\boldsymbol{z_n}, other\ params)$ to generate something very similar to that datapoint $\boldsymbol{x_n}$.
  - Given $\boldsymbol{z_n}$ and $\boldsymbol{\theta}$, there exist a complex function $f(\boldsymbol{z_n};\ \boldsymbol{\theta})$ such that some datapoint $\boldsymbol{x_n}$ can be generated by this complex function.
  - Neural Network: A powerful function approximation technique → Complex function $f(\boldsymbol{Z_n};\ \boldsymbol{\theta})$: Neural Network,

- $\boldsymbol{\theta}$: All the parameters (e.g. weights, activation function, etc.) in the Neural Network (NN)
- Input of the NN: $\boldsymbol{Z_n}, \boldsymbol{\theta}$
- Output of the NN $\big(f(\boldsymbol{Z_n};\ \boldsymbol{\theta})\big)$: $\boldsymbol{\mu_\theta(Z_n)}, \boldsymbol{\sigma_\theta(Z_n)}$

Variational Autoencoder (VAE)

- $P(\boldsymbol{X_n}|\, \boldsymbol{Z_n};\, \boldsymbol{\theta}) = N\big(\boldsymbol{\mu_\theta}(\boldsymbol{z_n}), \boldsymbol{\sigma_\theta}(\boldsymbol{z_n})\big)$. (In general, no limitation)
- Prior: $\mathrm{P}(\boldsymbol{Z_n}) = N(0, I)$
  - Concept: Provided powerful function approximations (e.g. neural network), we can imagine the network using its first few layers to map the normally distributed $\boldsymbol{Z_n}$ to latent values with exactly the right statistics. Then it can use later layers to map those latent layers to the real data.
- Likelihood of $\boldsymbol{X_n}$: $L(\boldsymbol{\theta}, \boldsymbol{X_n}) = P(\boldsymbol{X_n};\, \boldsymbol{\theta}) = \int p(\boldsymbol{x_n}|\boldsymbol{z_n};\, \boldsymbol{\theta}) \cdot p(\boldsymbol{z_n})\, d\boldsymbol{z_n}$



Variational Autoencoder (VAE)

- What does $P(\boldsymbol{X_n};\ \boldsymbol{\theta})$ look like?
  - Finite Gaussian Mixture Model (in GMM) vs. Infinite Gaussian Mixture Model (in VAE)

Gaussian Mixture Model

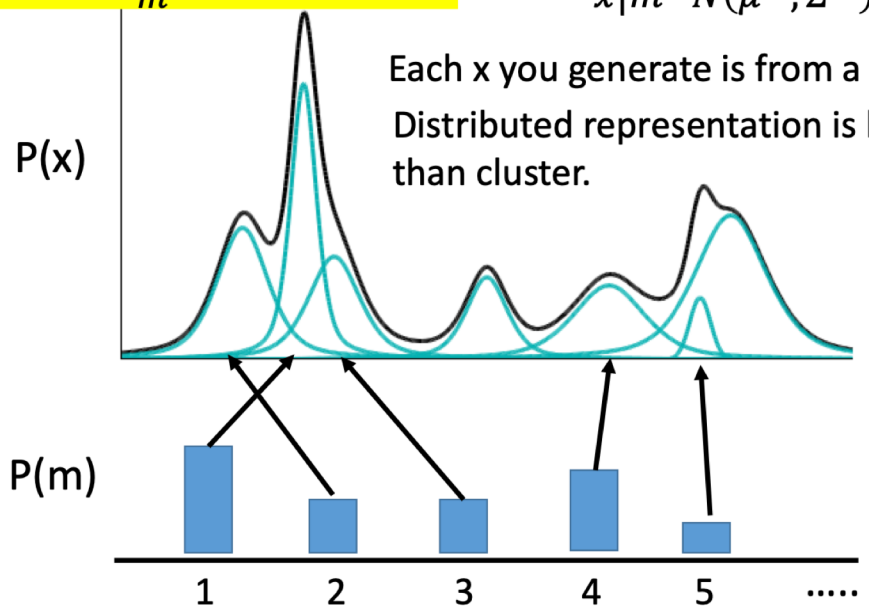
$$P(x) = \sum_m P(m)P(x|m)$$

How to sample?

$m \sim P(m)$ (multinomial)

m is an integer

$x|m \sim N(\mu^m, \Sigma^m)$



Each x you generate is from a mixture

Distributed representation is better than cluster.

What if $P(m)$ is a continuous function?

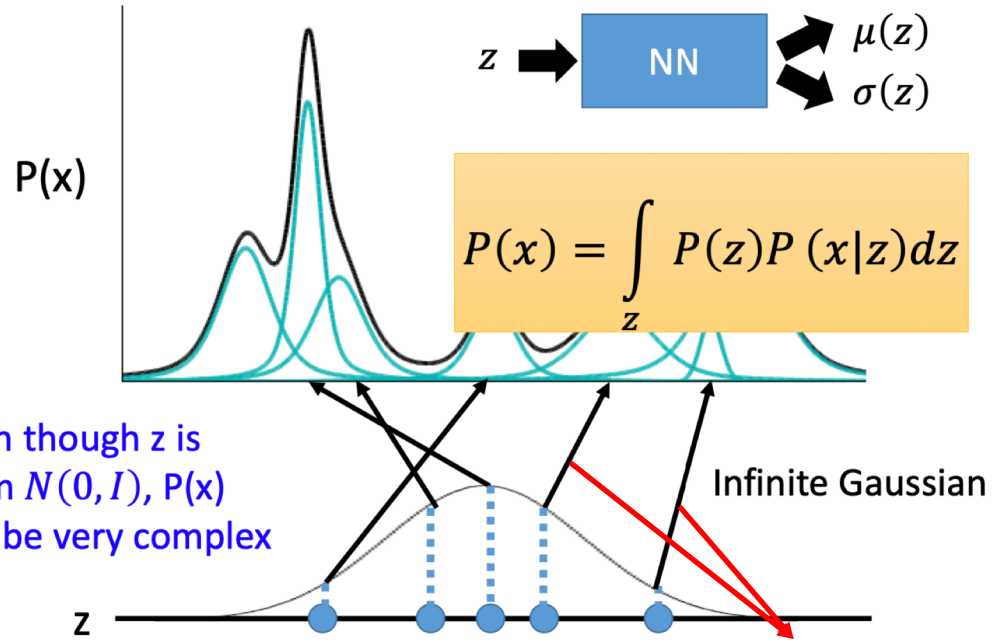
VAE

$z \sim N(0, I)$

z is a vector from normal distribution

$x|z \sim N(\mu(z), \sigma(z))$

Each dimension of z represents an attribute



These mappings ($\boldsymbol{\mu_\theta(z_n)}$) and their shapes of $P(\boldsymbol{X_n}|\ \boldsymbol{Z_n})$ ($\boldsymbol{\sigma_\theta(z_n)}$) are approximated by the neural network.

Variational Autoencoder (VAE)

- Goal: Let the model $P(\boldsymbol{X_n};\ \boldsymbol{\theta})$ be representative of the entire target dataset.
- Concept:
  - Maximize the probability of each $\boldsymbol{X_n}$ in the training set under the entire generative process with respect to $\boldsymbol{\theta}$.
  - Optimize $\boldsymbol{\theta}$ s.t. we can sample $\boldsymbol{Z_n}$ from $\mathrm{P}(\boldsymbol{Z_n})$ and with high probability, $\boldsymbol{\mu_\theta}(\boldsymbol{z_n})$ will be like the $\boldsymbol{X_n}$ in our dataset.
- Interest: Maximize $L(\boldsymbol{\theta}, \boldsymbol{X_n}) = P(\boldsymbol{X_n};\ \boldsymbol{\theta})$ → MLE of $\boldsymbol{\theta}$
- $P(\boldsymbol{X_n};\ \boldsymbol{\theta})$ cannot be handled directly due to the complex model and the integration of $\boldsymbol{Z_n}$.
- Variational Inference (VI) → Find an approximation for the model evidence $\log P(\boldsymbol{X_n};\ \boldsymbol{\theta})$. Additionally, VI can also provide an estimate to the posterior $P(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\theta})$.

Variational Autoencoder (VAE)

- By variational inference:

  $\log L(\boldsymbol{\theta}, \boldsymbol{X_n}) = \log P(\boldsymbol{X_n};\ \boldsymbol{\theta}) = E_{\boldsymbol{z}\sim q}[logP(\boldsymbol{X_n}|\boldsymbol{Z_n};\boldsymbol{\theta})] - KL[q(\boldsymbol{Z_n})||P(\boldsymbol{Z_n})] + KL[q(\boldsymbol{Z_n})||P(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\theta})]$,

  where $q(\boldsymbol{Z_n})$ can be any kind of distribution.

- Since we are interesting in inferring $P(\boldsymbol{X_n};\ \boldsymbol{\theta})$ & $P(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\theta})$ → Let $q(\boldsymbol{Z_n})$ depends on $\boldsymbol{X_n}$

  $\log L(\boldsymbol{\theta}, \boldsymbol{X_n}) = E_{\boldsymbol{z}\sim q}[logP(\boldsymbol{X_n}|\boldsymbol{Z_n};\boldsymbol{\theta})] - KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n})||P(\boldsymbol{Z_n})] + KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n})||P(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\theta})]$

  $L_b(q(\boldsymbol{Z_n}|\boldsymbol{X_n}), \boldsymbol{\theta})$

- Cost function: $L_b(q(\boldsymbol{Z_n}|\boldsymbol{X_n}), \boldsymbol{\theta})$
- Maximize $\log L(\boldsymbol{\theta}, \boldsymbol{X_n})$ ⟺ Maximize $L_b(q(\boldsymbol{Z_n}|\boldsymbol{X_n}), \boldsymbol{\theta})$
- Here, introducing another NN with parameters ϕ and let $q(\boldsymbol{Z_n}|\boldsymbol{X_n}) \Longrightarrow q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})$ and define $\boldsymbol{Z_n}\sim q_\phi \equiv q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})$ be the family of $N\left(\boldsymbol{\mu_\phi}(\boldsymbol{X_n}), diag\left(\boldsymbol{\sigma_\phi}(\boldsymbol{X_n})\right)\right)$
- $L_b(q(\boldsymbol{Z_n}|\boldsymbol{X_n}), \boldsymbol{\theta}) \Longrightarrow L_b(q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi}), \boldsymbol{\theta})$, $KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n})||P(\boldsymbol{Z_n})] \Longrightarrow KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})||P(\boldsymbol{Z_n})]$,
- $KL[N(\mu_0, \Sigma_0)||N(\mu_1, \Sigma_1)] = \frac{1}{2}\left(tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T\Sigma_1^{-1}(\mu_1 - \mu_0) - k + \log\left(\frac{\det(\Sigma_1)}{\det(\Sigma_0)}\right)\right)$
- Thus, $KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})||P(\boldsymbol{Z_n})] = \frac{1}{2}\left(tr\left(diag\left(\boldsymbol{\sigma_\phi}(\boldsymbol{X_n})\right)\right) + \left(\boldsymbol{\mu_\phi}(\boldsymbol{X_n})\right)^T \cdot \left(\boldsymbol{\mu_\phi}(\boldsymbol{X_n})\right) - k + \log\left(\det\left(diag\left(\boldsymbol{\sigma_\phi}(\boldsymbol{X_n})\right)\right)\right)\right)$ → It has an analytical form, so does its gradient.

Variational Autoencoder (VAE)

- By the Monte Carlo Simulation, and W.L.L.N, $E_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}}[logP(\boldsymbol{X_n}|\boldsymbol{Z_n};\boldsymbol{\theta})]$ in the $L_b(q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi}),\boldsymbol{\theta})$ can be approximated by $\frac{1}{L}\sum_{l=1}^{L} log\ P\left(\boldsymbol{X_n}\middle|\boldsymbol{Z_{n,l}};\boldsymbol{\theta}\right)$, where $\boldsymbol{Z_{n,l}} \sim \boldsymbol{q}(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})$, L is the number of samples.
  - If L is large enough, $\frac{1}{L}\sum_{l=1}^{L} log\ P(\boldsymbol{X_n}|\boldsymbol{Z_{n,l}};\boldsymbol{\theta})$ will converge to $E_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}}[logP(\boldsymbol{X_n}|\boldsymbol{Z_n};\boldsymbol{\theta})]$ in probability.

    → Large L is too expensive.

    → In practice, take just one sample of $\boldsymbol{Z_{n,l}}$ from $\boldsymbol{q}(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})$ & treat $P\left(\boldsymbol{X_n}\middle|\boldsymbol{Z_{n,l}};\boldsymbol{\theta}\right)$ for that $\boldsymbol{Z_{n,l}}$ as an approximation of $E_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}}[logP(\boldsymbol{X_n}|\boldsymbol{Z_n};\boldsymbol{\theta})]$ in each batch. (since doing stochastic gradient descent.)
- Problem for Taking Gradient (Reparametrize the Model)
  - $\nabla_{\theta,\phi} KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})||P(\boldsymbol{Z_n})]$ is tractable, but $\nabla_{\theta,\phi} E_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}}[logP(\boldsymbol{X_n}|\boldsymbol{Z_n};\boldsymbol{\theta})]$ cannot be computed.
  - The reparameterization trick: Let $\boldsymbol{\epsilon} \sim N(0,1)$, $\boldsymbol{Z_n} = g(\boldsymbol{\epsilon},\boldsymbol{X_n},\boldsymbol{\phi}) = \boldsymbol{\mu_\phi}(\boldsymbol{X_n}) + \boldsymbol{\epsilon}\cdot diag\left(\boldsymbol{\sigma_\phi}(\boldsymbol{X_n})\right)$

    s.t. $\boldsymbol{Z_{n,l}} \sim \boldsymbol{q}(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi}) \equiv N\left(\boldsymbol{\mu_\phi}(\boldsymbol{X_n}), diag\left(\boldsymbol{\sigma_\phi}(\boldsymbol{X_n})\right)\right)$

    → $\nabla_{\theta,\phi} E_{\boldsymbol{\epsilon}\sim N(0,1)}[logP(\boldsymbol{X_n}|g(\boldsymbol{\epsilon},\boldsymbol{X_n},\boldsymbol{\phi});\boldsymbol{\theta})] = E_{\boldsymbol{\epsilon}\sim N(0,1)}\nabla_{\theta,\phi}[logP(\boldsymbol{X_n}|g(\boldsymbol{\epsilon},\boldsymbol{X_n},\boldsymbol{\phi});\boldsymbol{\theta})]$

    → Monte Carlo Simulation: $\frac{1}{L}\sum_{l=1}^{L} log\ P(\boldsymbol{X_n}|g(\boldsymbol{\epsilon},\boldsymbol{X_n},\boldsymbol{\phi});\boldsymbol{\theta})$

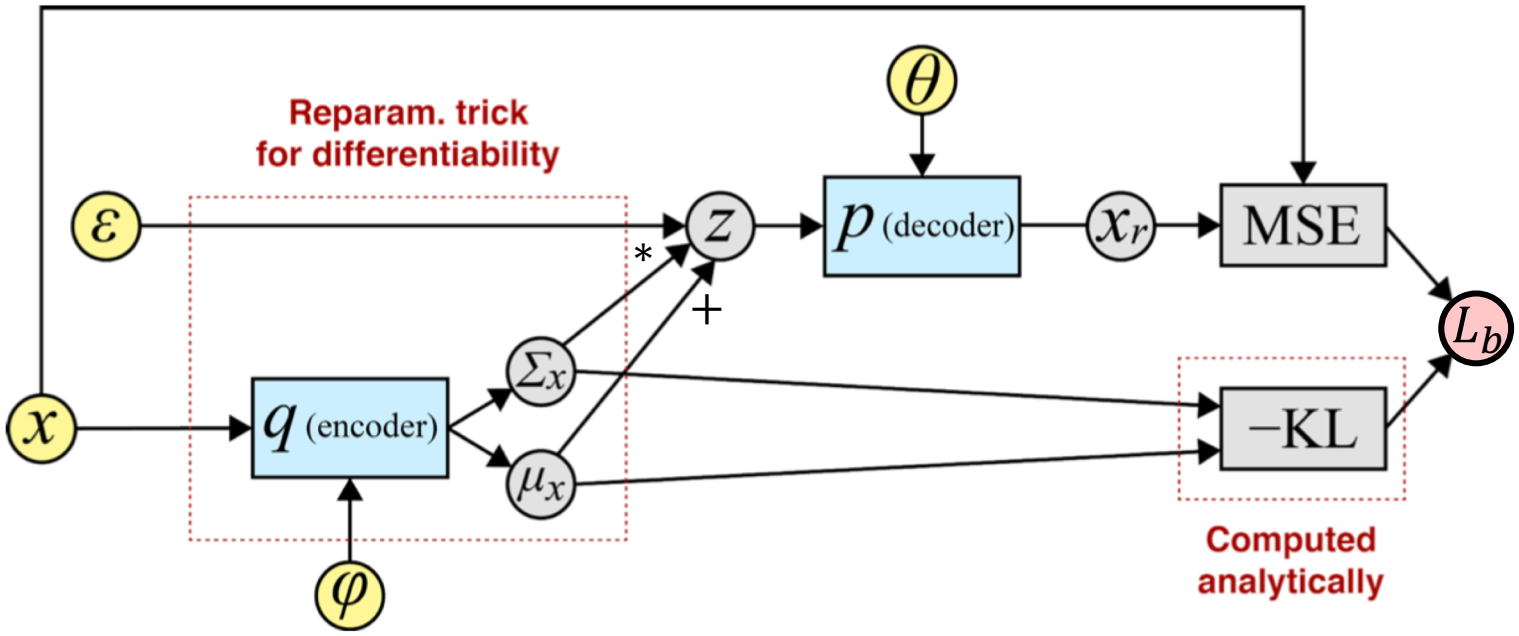
    → Use same trick as $E_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}}[logP(\boldsymbol{X_n}|\boldsymbol{Z_n};\boldsymbol{\theta})]$

Variational Autoencoder (VAE)

- Relationship with the Autoencoder
  - Cost function: $L_b(q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi}),\boldsymbol{\theta}) = E_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}}[logP(\boldsymbol{X_n}|\boldsymbol{Z_n};\boldsymbol{\theta})] - KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})||P(\boldsymbol{Z_n})]$

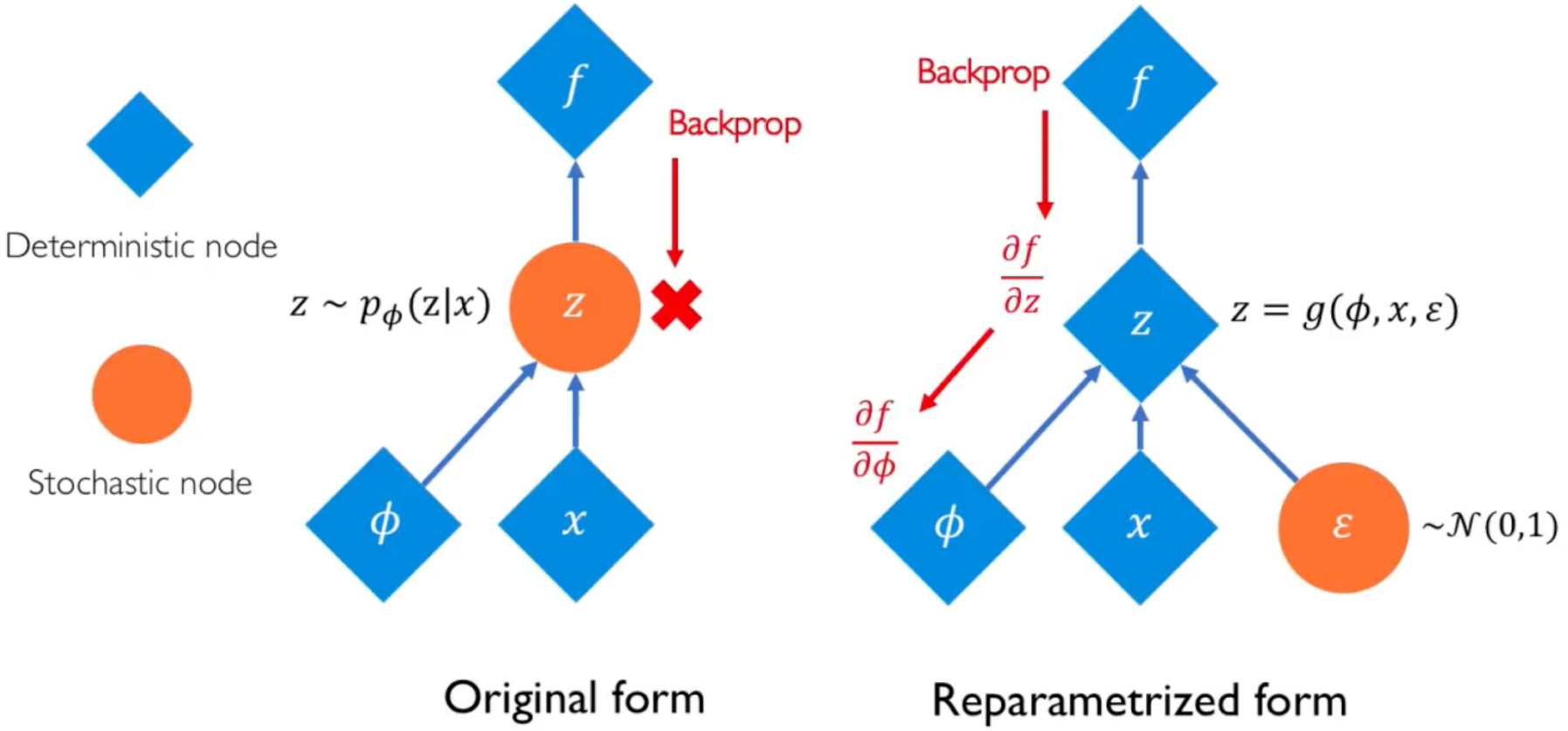
    $= E_{\boldsymbol{\epsilon}\sim N(0,1)}[logP(\boldsymbol{X_n}|g(\boldsymbol{\epsilon},\boldsymbol{X_n},\boldsymbol{\phi});\boldsymbol{\theta})] - KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})||P(\boldsymbol{Z_n})]$
    - $q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})$: Encoder, $P(\boldsymbol{X_n}|g(\boldsymbol{\epsilon},\boldsymbol{X_n},\boldsymbol{\phi});\boldsymbol{\theta})$: Decoder, $\boldsymbol{\mu_\phi}(\boldsymbol{X_n}) = \mu_x$, $\boldsymbol{\sigma_\phi}(\boldsymbol{X_n}) = \Sigma_x$
    - $E_{\boldsymbol{\epsilon}\sim N(0,1)}[logP(\boldsymbol{X_n}|g(\boldsymbol{\epsilon},\boldsymbol{X_n},\boldsymbol{\phi});\boldsymbol{\theta})]$: MSE, (Reconstruction Error), it based on the encoder and the decoder.
    - $KL[q(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\phi})||P(\boldsymbol{Z_n})]$ only based on the encoder.

      (A penalty for Σ_x (if Σ_x close to 0 → autoencoder), we want it to close 1 (same as prior).)

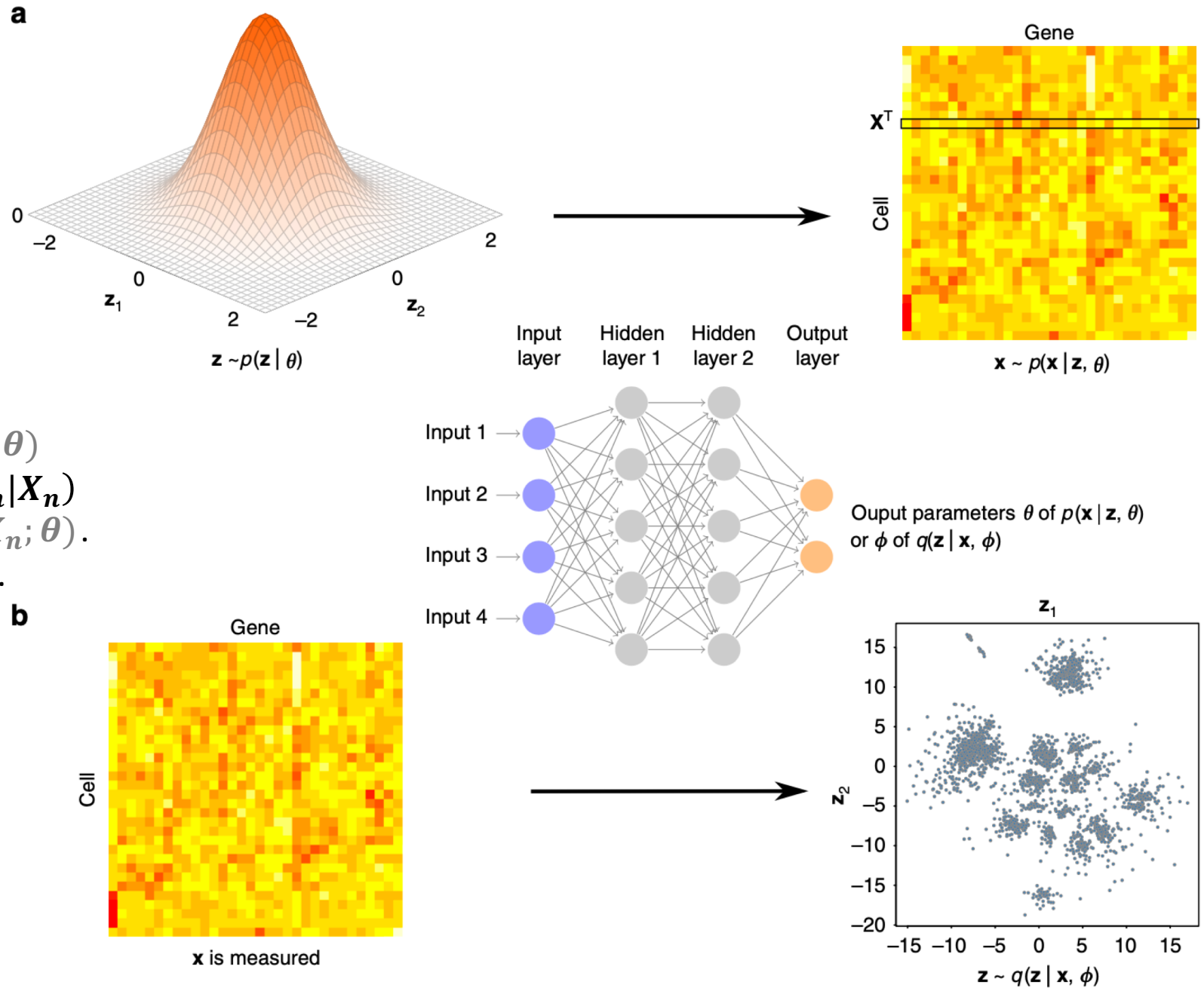


Variational Autoencoder (VAE)

- Graphical Illustration for the Reparametrized Trick



SCVIS



- Interest: Find $P(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\theta})$
  → Find an optimal $q(\boldsymbol{Z_n}|\boldsymbol{X_n})$
  to approximate $P(\boldsymbol{Z_n}|\boldsymbol{X_n};\boldsymbol{\theta})$.
  → Variational Inference.

SCVIS

- Let $P(\boldsymbol{X_n}|\,\boldsymbol{Z_n};\,\boldsymbol{\theta}) = N\big(\boldsymbol{\mu_\theta}(\boldsymbol{z_n}), \boldsymbol{\sigma_\theta}(\boldsymbol{z_n})\big) \rightarrow student - t\ dist. with\ degree\ with\ freedom\ \boldsymbol{v}.$

$$p(\mathbf{x}_n|\mathbf{z}_n, \boldsymbol{\theta}) = \mathcal{T}(\mathbf{x}_n|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_n), \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{z}_n), \boldsymbol{\nu})$$

Prevent overfitting.

- $L_b(q(\boldsymbol{Z_n}|\boldsymbol{X_n}), \boldsymbol{\theta}) = \text{ELBO}_n = -\mathbb{KL}(q(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\phi})||p(\mathbf{z}_n|\boldsymbol{\theta})) + \frac{1}{L}\sum_{l=1}^{L} \log p(\mathbf{x}_n|\mathbf{z}_{n,l}, \boldsymbol{\theta})$

- Adding non-symmetrized t-SNE regularizers on the latent variables:
  - For High-D:

$$p_{j|i} = \frac{\exp\left(-\mathbf{x}_i - \mathbf{x}_j^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\mathbf{x}_i - \mathbf{x}_k^2/2\sigma_i^2\right)}$$

  - For Low-D:

$$q_{j|i} = \frac{\left(1+\left\|\mathbf{z}_i-\mathbf{z}_j\right\|^2/\nu\right)^{-\frac{\nu+1}{2}}}{\sum_{k,k\neq i}\left(1+\|\mathbf{z}_i-\mathbf{z}_k\|^2/\nu\right)^{-\frac{\nu+1}{2}}}$$

  - Encourage forming gaps between clusters.

SCVIS

- Regularizer:

$$\sum_i \mathbb{KL}\left(p_{\cdot|i} || q_{\cdot|i}\right) = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

- Final objective function:

$$\arg\min_{\theta,\phi} \left(-\sum_{n=1}^{N} \mathrm{ELBO}_n + \alpha \sum_{n=1}^{N} \mathbb{KL}\left(p_{\cdot|n} || q_{\cdot|n}\right) \right)$$

  - α is set to the dimensionality of the input high-dimensional data.
    (ELBO scales with the dimensionality of the input data)

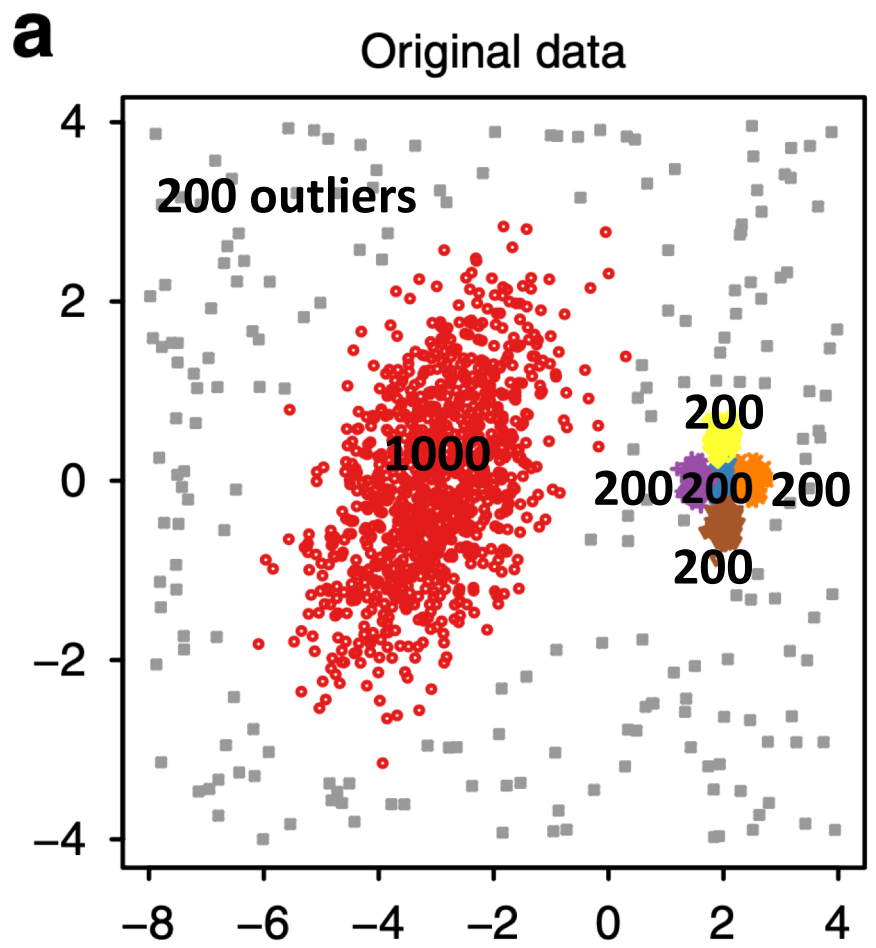
SCVIS

- Other parameters settings
  - Variational approximation NN hidden layers: 3 (128, 64, 32 hidden units for each layer.)
  - Model NN hidden layers: 5 (32, 32, 32, 64, 128 hidden units for each layer.)
  - Activation function: exponential linear unit (speed up the convergence of optimization).
  - L2 regularizer of 0.001 on the weights of the neural networks (prevent overfitting.)
  - Mini-batch size: 512 (cells) (for controlling the time complexity of t-SNE computation.)
  - Adam stochastic gradient descent with learning rate: 0.01.
  - Run optimization for 500 epochs for each dataset with at least 3000 iterations.
  - For large dataset, running a maximum of 30000 iteration or two epochs.

Simulation

Simulation

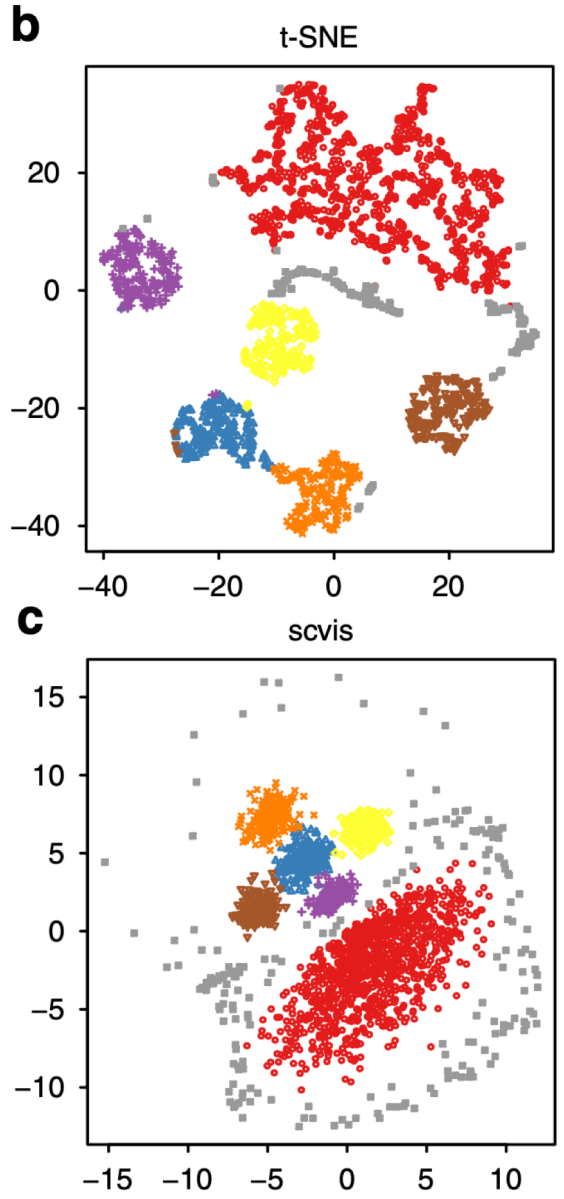
- Simulated data



- Six clusters with outlier data (gray), total: 2200 samples.
- A transformation was made:
  $2D\colon (x, y) \rightarrow 9D\colon (x + y, x - y, xy, x^2, y^2, x^2y, xy^2, x^3, y^3)$
- Each of nine features was divided its $\max(|\cdot|)$,
  $|\cdot|$: absolute value.
- Compare with t-SNE, Gaussian process latent variate model (GPLVM), parametric t-SNE (pt-SNE), and PCA.

Simulation

- Performance
  - t-SNE is challenging to infer the overall layout of the six clusters. (Also, the global structure is not reliable.)
  - t-SNE put the outliers data (gray) into several compact clusters.
  - SCVIS preserved the overall structure of the original data. (The relative positions of the clusters were also preserved.)
  - SCVIS let the outliers data (gray) surround the genuine clusters as the originally untransformed data.

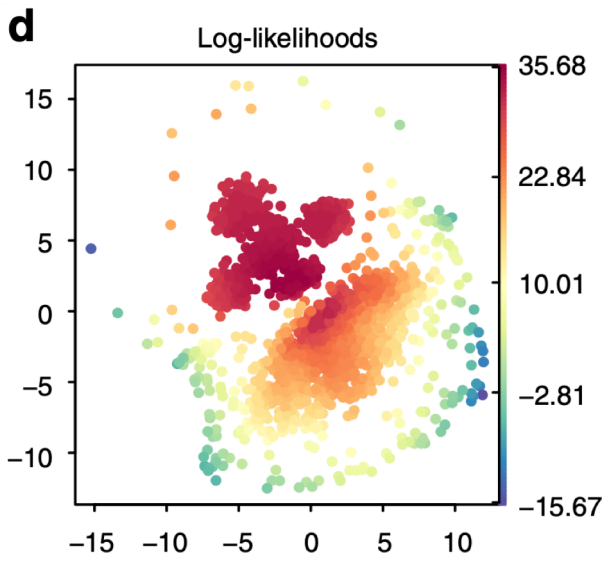


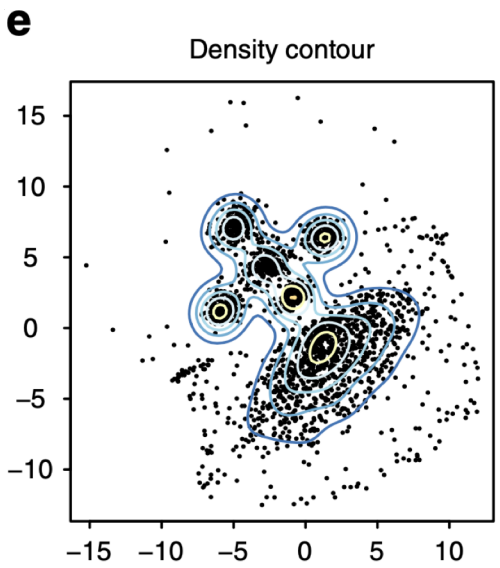
Simulation

- Performance

  - Due to the use of generating model, SCVIS provide a way to quantify the uncertainty of the low-dimensional mapping by its log-likelihood.

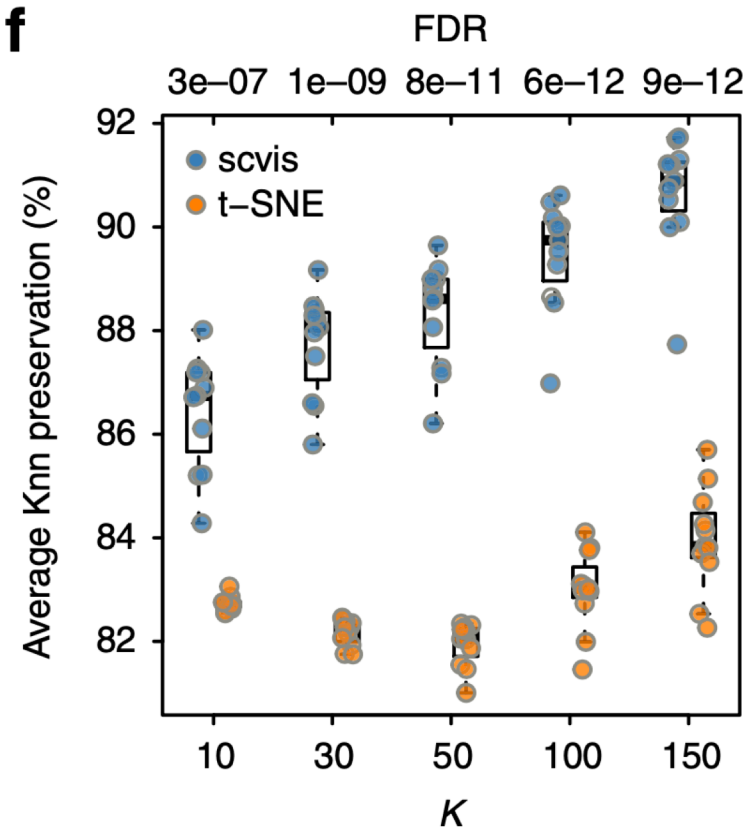
  - SCVIS put most of its modeling power to model the five compact clusters, while the outliers have lower log-likelihood.





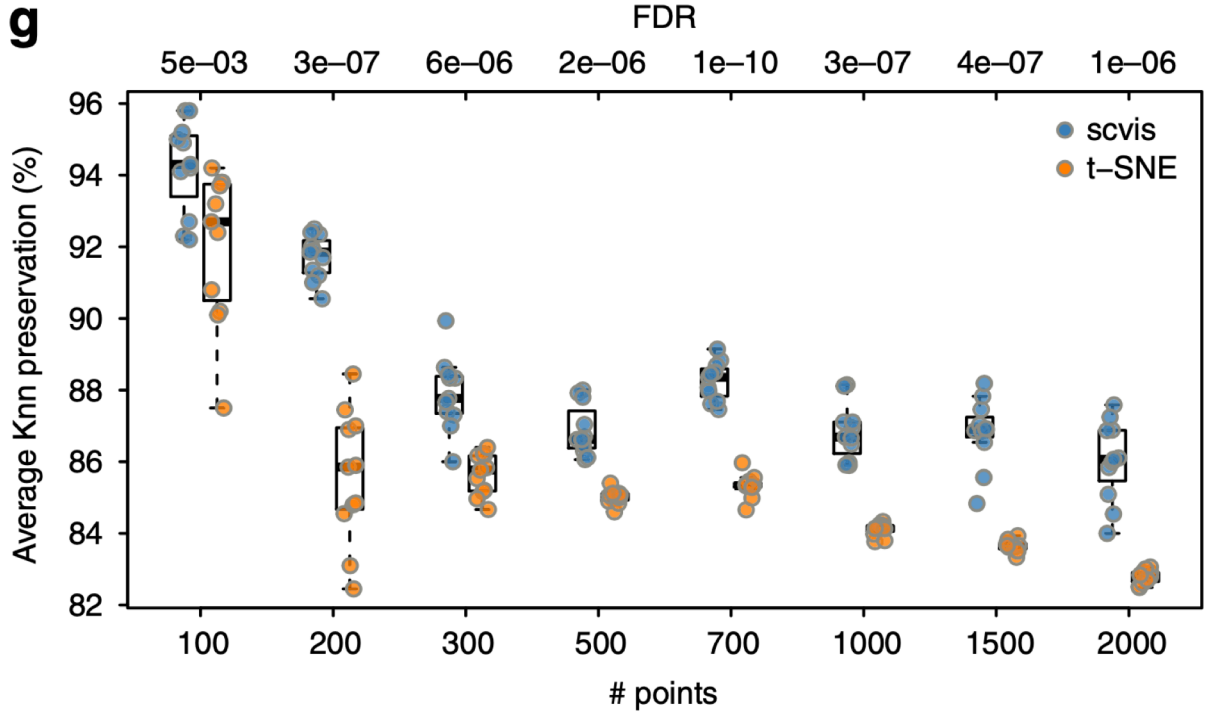
Simulation

- Stability evaluation (t-SNE vs. SCVIS)
  - Low-dimensional embedding results from 10 runs are similar → Stable.
  - A quantitative evaluation method: the average KNN preservations for each run.
    - SCVIS preserved KNN more effectively than t-SNE.
    - The t-SNE result consistent with the property of preserving local structure.



Simulation

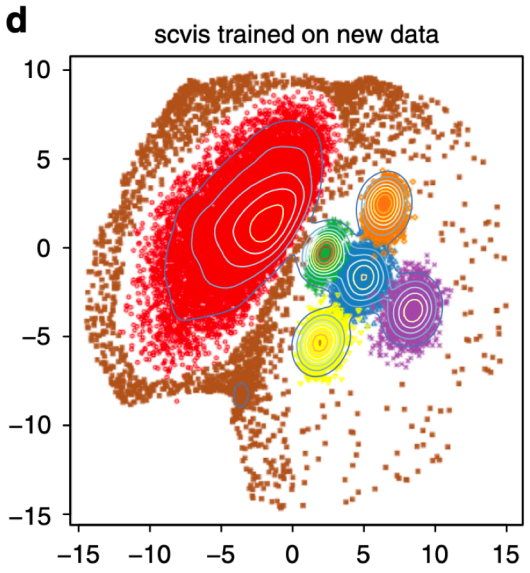
- Small datasets evaluation (t-SNE vs. SCVIS)
  - Data is sampled from the original simulation dataset.
  - Sample size: 100, 200, 300, 500, 700, 1000, 1500, 2000
  - Evaluation: KNN with K = 10



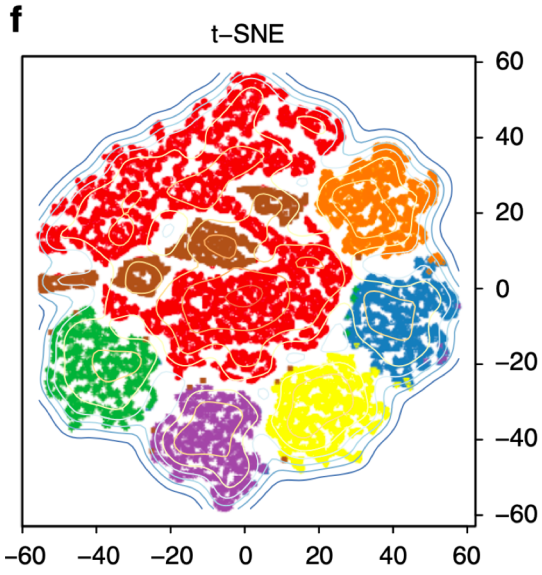
- Compare with t-SNE, SCVIS performs very well on all the subsampled data. (preserved much of the structure of the data)

Simulation

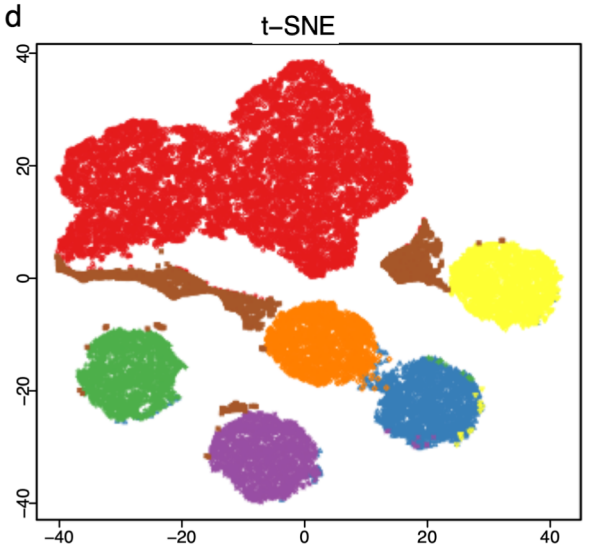
- Large datasets evaluation (t-SNE vs. SCVIS)
  - Tenfold the number of points in each cluster generated by a different random seed (total: 22000 points).
  - In t-SNE, sample size increase → the optimal perplexity increase.
  - Sensitivity to the hyperparameter setting: the perplexity parameter.



Robust to the perplexity parameter.
(Sample size fixed (512) at each training step.)



Without perplexity adjustment



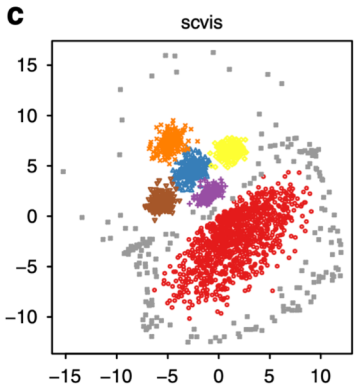
With perplexity adjustment

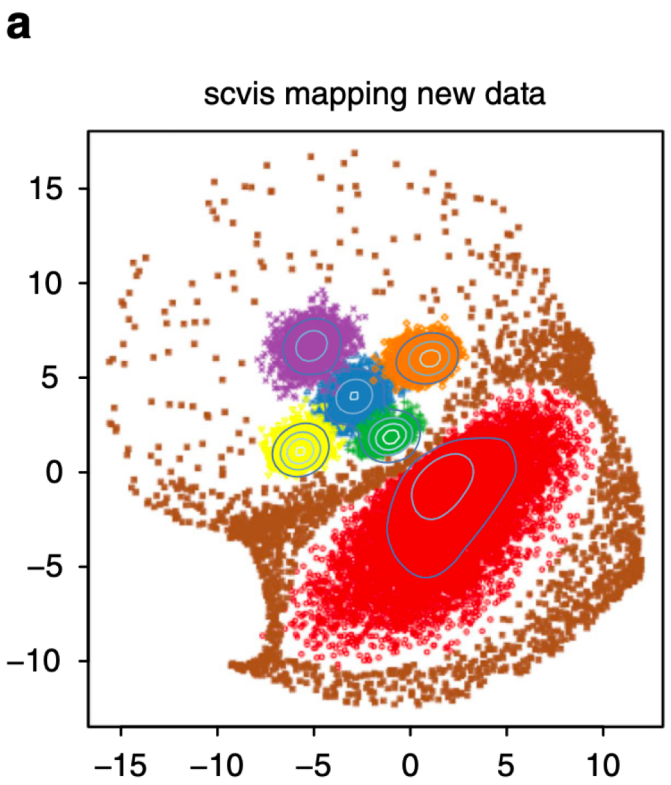
Hard to set this parameter in practice.

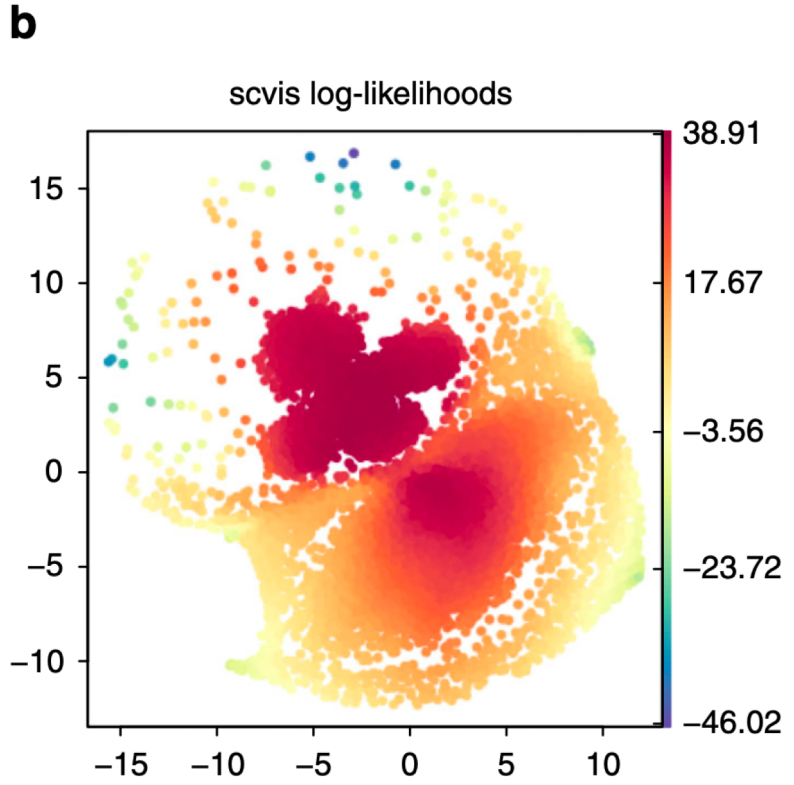
Simulation

- Model performance on the new data embedding (SCVIS)
  - Use the original simulation model (based on the 2200 data points).
  - The SCVIS embedding result is similar to the original simulation data.

Original data

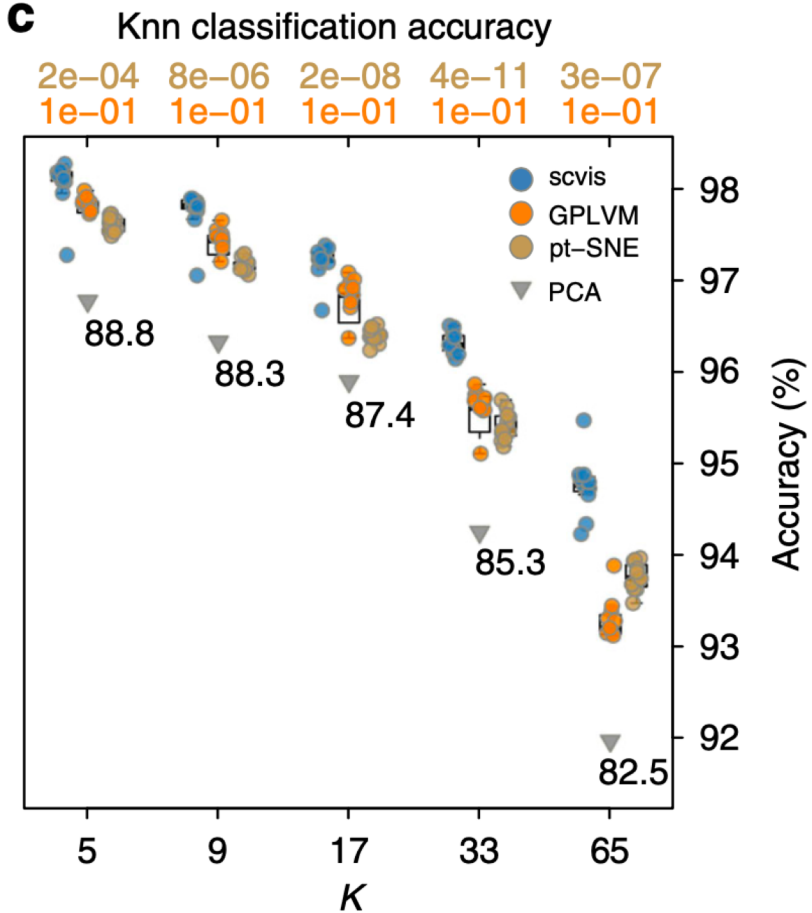






Simulation

- Model performance on the new data embedding (pt-SNE, GPLVM, PCA, and SCVIS)
  - Train SCVIS, GPLVM, pt-SNE and KNN classifiers on the original 2200 simulated data.
  - Applied these trained models to the tenfold dataset (22000 simulated data.)



- SCVIS performs significantly better for different Ks.
- For a larger K, SCVIS assigns the outliers to the six genuine clusters (performance decrease).
- PCA is worse than other methods.

Real Data Analysis
(Single-cell datasets)

Single-cell Datasets

- A. (scRNA-seq) Intratumor heterogeneity & tumor microenvironment in oligodendroglioma.
  - Expression of 23686 genes in 4347 cells from six IDH1 or IDH2 mutant human oligodendroglioma patients.
  - Expression value: $\log_2(TPM/10+1)$, TPM: Transcripts Per Million.
  - 303 normal (without detectable copy number alterations) cells.

- B. (scRNA-seq) Intratumor heterogeneity & tumor microenvironment in metastatic melanoma.
  - Expression of 23686 genes in 4645 cells from 19 metastatic melanoma patients.
  - Includes malignant cells, immune cells, storm cells, endothelial cells data.

Single-cell Datasets

- C. (scRNA-seq) The mouse bipolar cell populations of the retina.
  - 27499 mouse retinal bipolar neural cells from a transgenic mouse.
  - 26 putative cell types were identified by clustering the first 37 principal components of all 27499 cells.
    - 15 clusters (96% of 27499 cells)
      - 14 clusters: Bipolar cells
      - 1 cluster: Muellerglia cells
    - 11 clusters (1060 cells)

- D. (scRNA-seq) All cell types in the mouse retina.
  - 44808 cells from the retinas of 14-day-old mice.
  - 39 clusters were identified using DBSCAN based on the two-dim t-SNE embedding.

- For all the scRNA-seq datasets, use PCA to reduce the data into 100-dimensional space, and use the projected data as inputs to scvis.

Single-cell Datasets

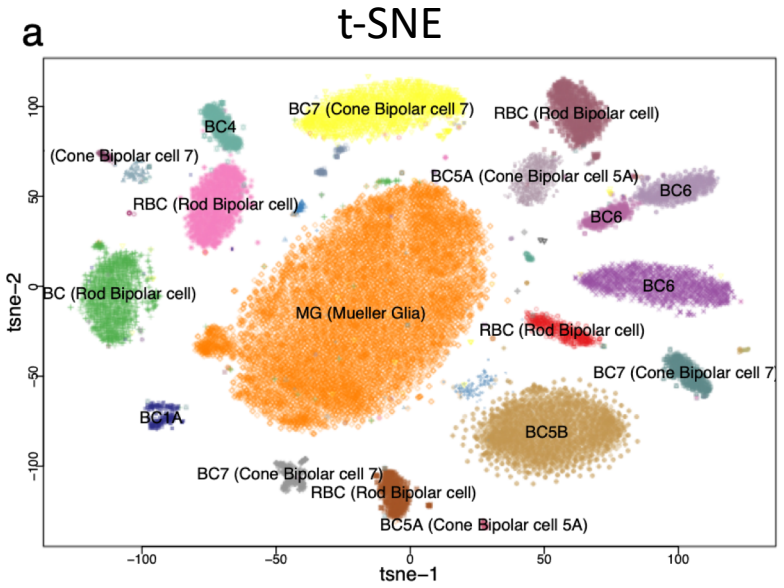
- Mass cytometry (CyTOF) datasets consists of bone marrow mononuclear cells from two healthy men H1 & H2.
  - E. Manually assigned 72463 cells to 14 cell types based on 32 measured surface protein markers on H1.
  - F. Manually assigned 31721 cells to 14 cell types based on 32 measured surface protein markers on H2.

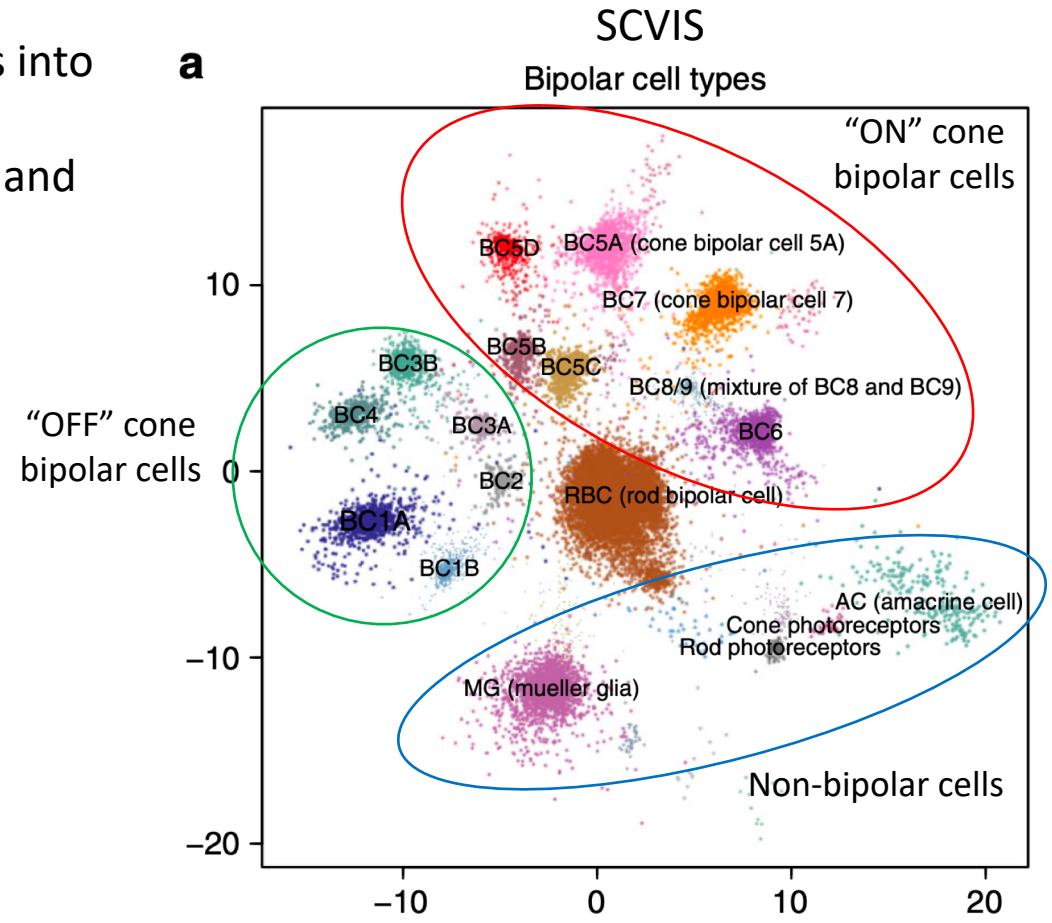
- For CyTOF dataset, use the original data as inputs to scvis.

- 10X Genomics neural cells datasets from two mice.
  - 1306127 cells from cortex, hippocampus, and subventricular zones of two E18 C57BL/6 mice.

Real Data Analysis

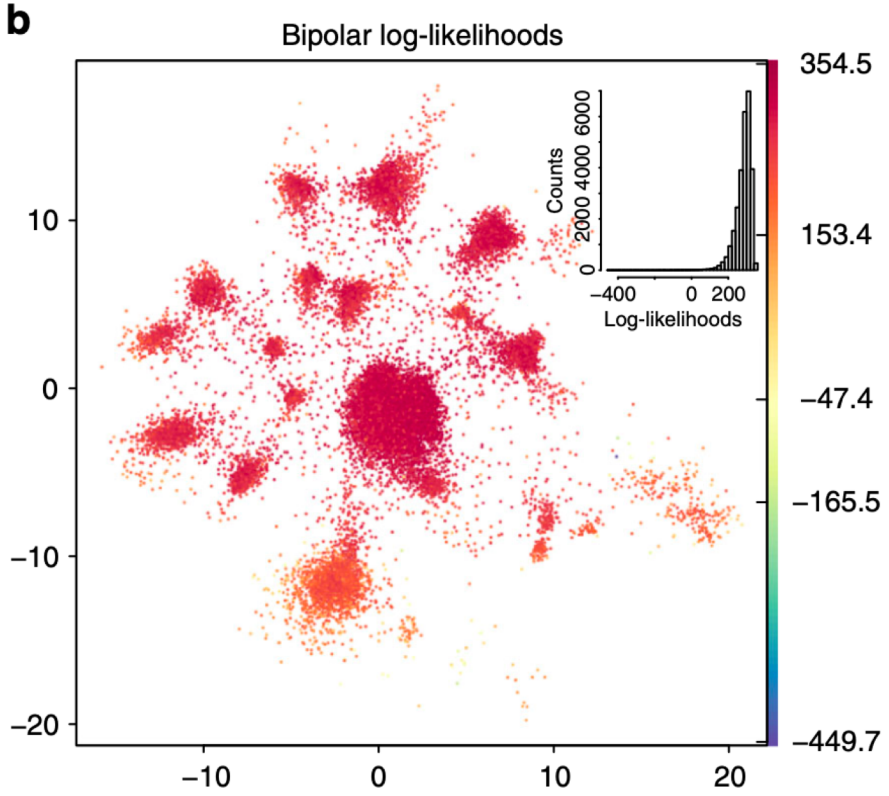
- Performance on a parametric mapping for a single-cell dataset (C. and D.)
  - Trained SCVIS model on the C. dataset and map the D. dataset.
  - Performance on the training data:
    - SCVIS mapped the cell doublets and contaminants into the low-density region in the low-D region.
    - Unlike SCVIS, t-SNE put the “outlier” cell doublets and contaminates into distinct compact cluster.
    - t-SNE cannot preserve global information.

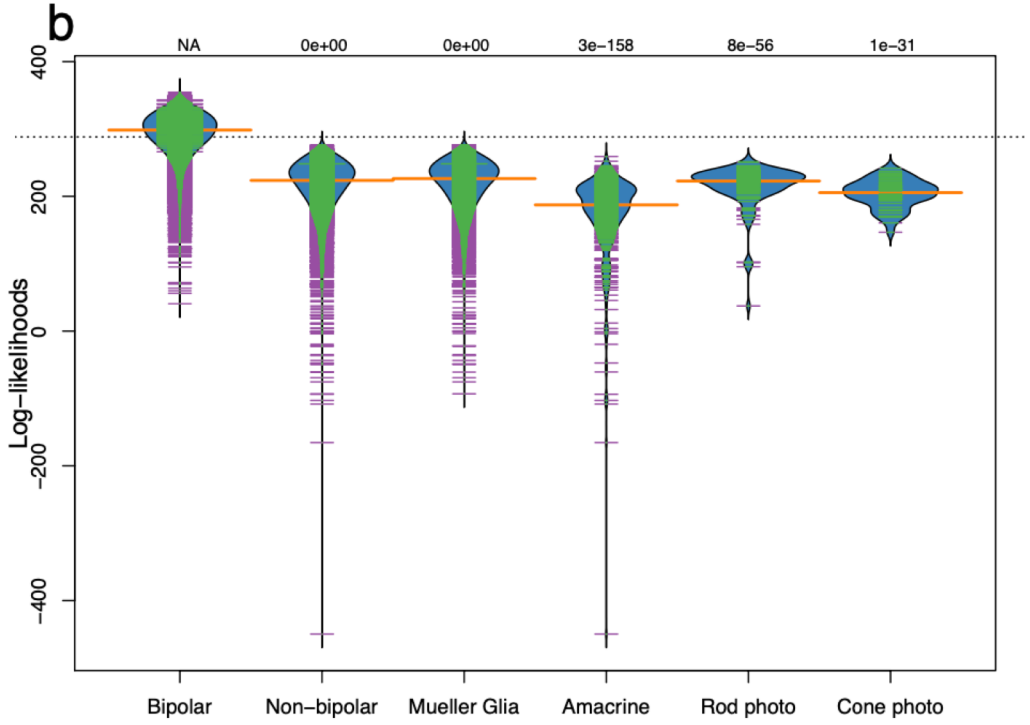




Real Data Analysis

- Performance on a parametric mapping for a single-cell dataset (C. and D.)
  - Performance on the training data:
    - The bipolar cells tends to have higher log-likelihoods.
      → the model used most of its power to model the bipolar cells, but other cell types were not modeled as well.

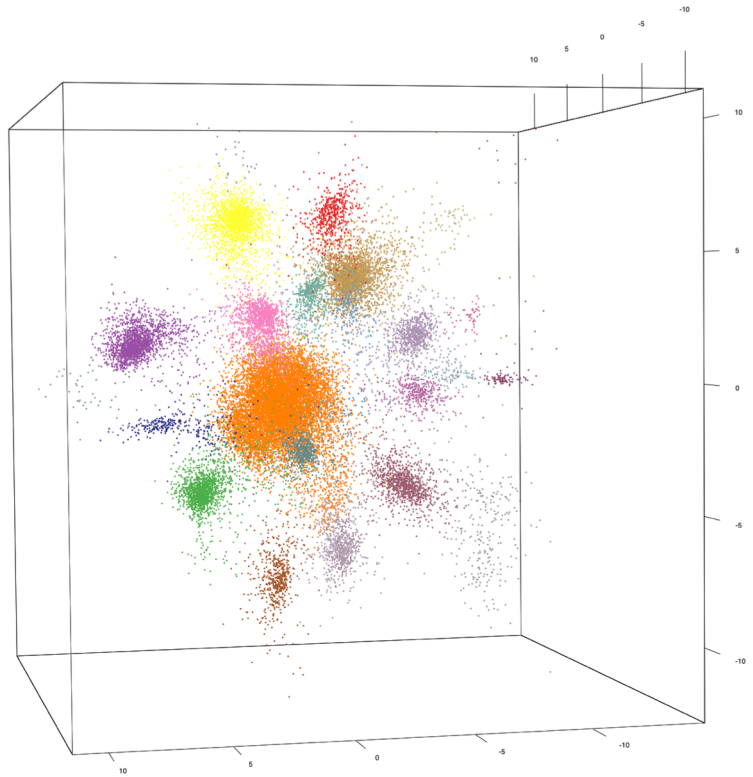




- The amacrine cells had the lowest median log-likelihood.

Real Data Analysis

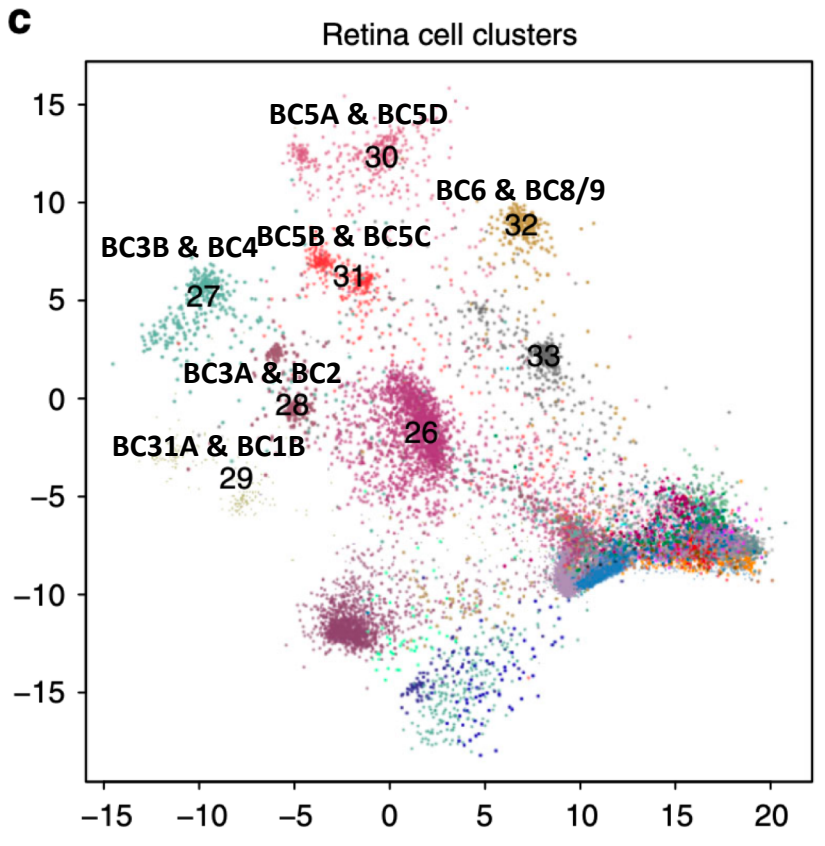
- Performance on a parametric mapping for a single-cell dataset (C. and D.)
  - Performance on the training data:
    - Project the C. data to a three-dimensional space.
    - Obtained a better average log-likelihood per data point and smaller KL divergence.



Supplementary Fig. 7: Projecting the bipolar data to a three-dimensional space. We obtained better average log-likelihood per data point, i.e., 255.1 versus 253.3 (from the last 100 iterations)

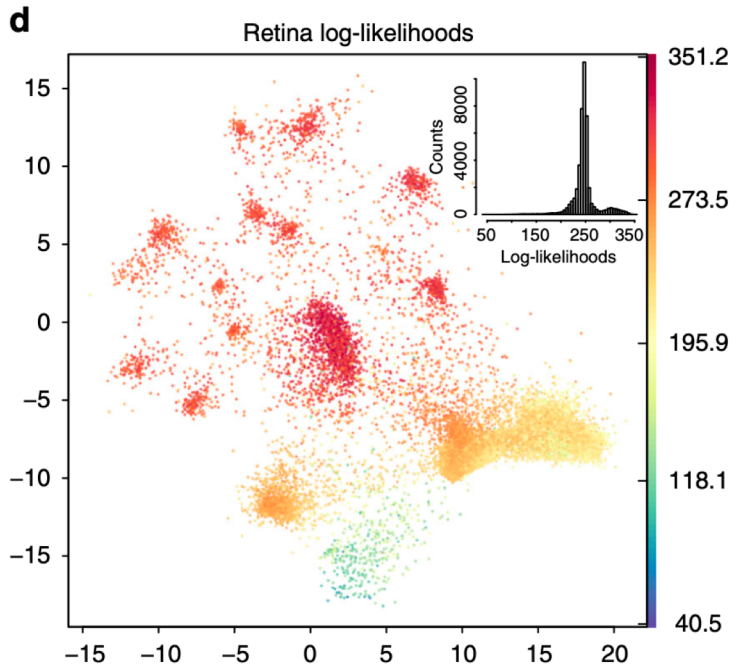
Real Data Analysis

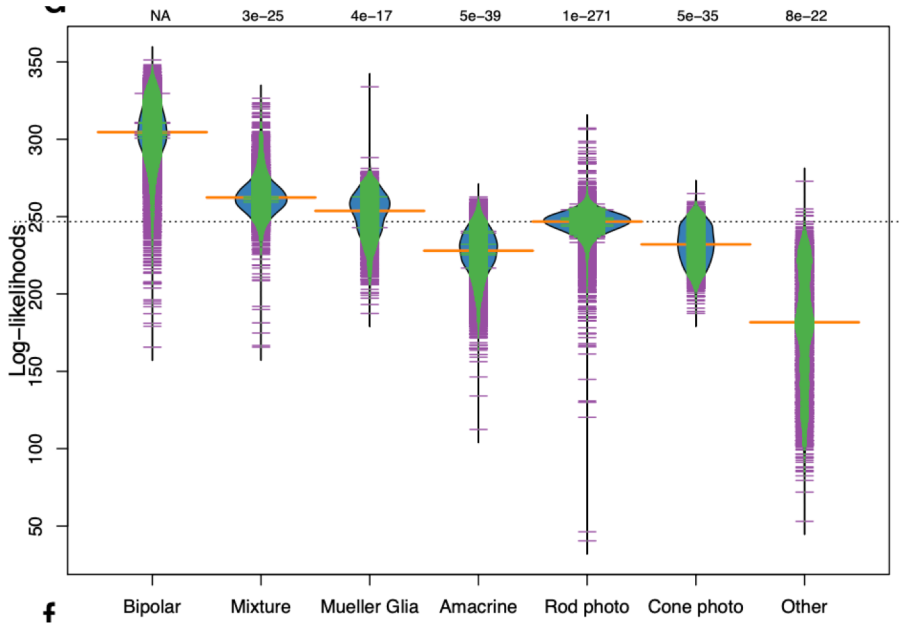
- Performance on a parametric mapping for a single-cell dataset (C. and D.)
  - Use PCA to project the D. dataset to the subspace of the first 100 principal component of the C. dataset.
  - Use the model trained on the C. dataset to map the D. dataset into two-dimensional space.
  - Performance on the D. data:
    - Tends to map data with the layout learned in the training data (C.)
    - Recently 14 subtypes of bipolar cells were present in this dataset.
    - The mixture of bipolar cells data should be further investigated.



Real Data Analysis

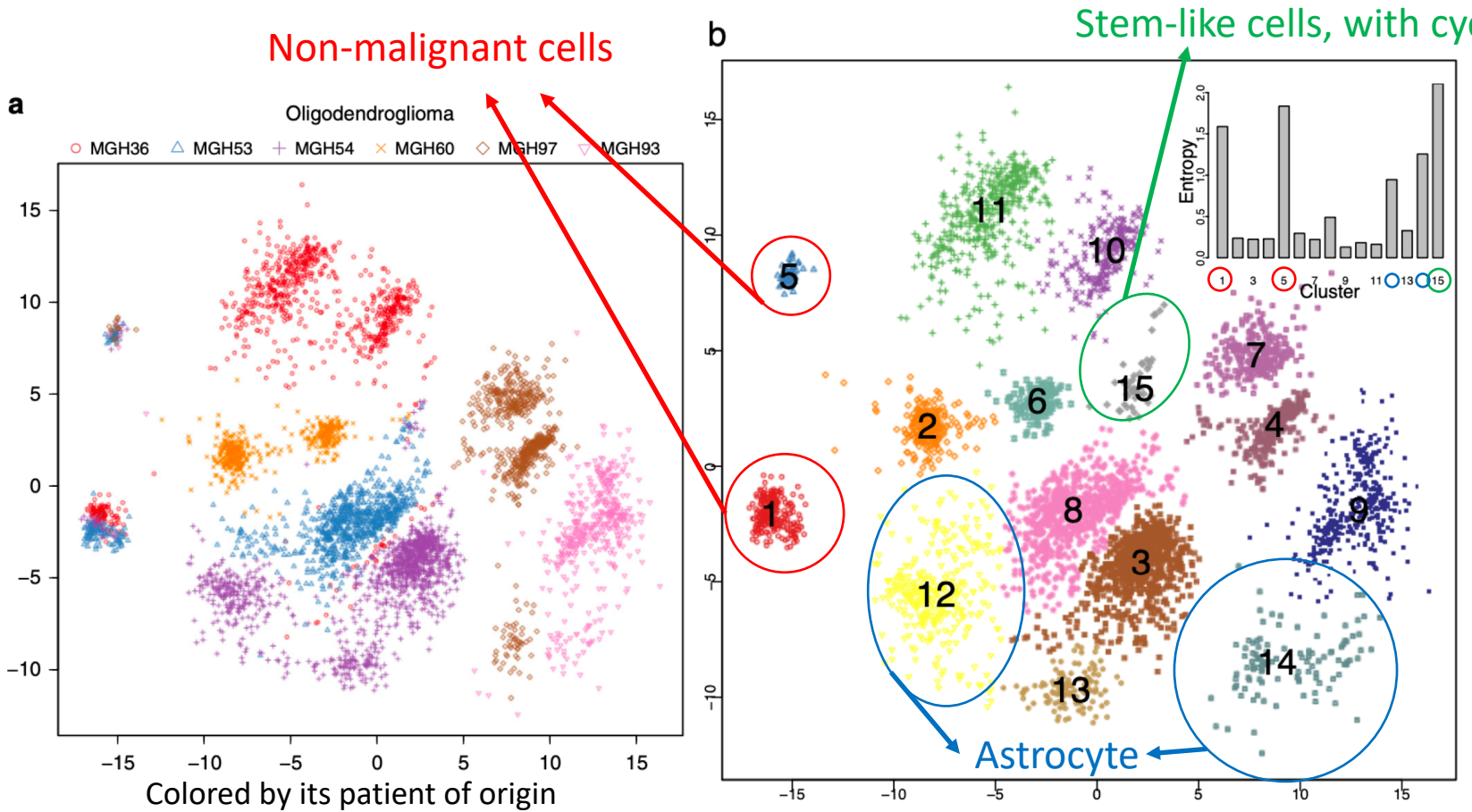
- Performance on a parametric mapping for a single-cell dataset (C. and D.)
  - Performance on the D. data:
    - The mixture of bipolar cells data had lower log-likelihoods.
    - As in the C. data, the bipolar cells in the D. data tend to have high log-likelihood.
    - The bipolar cells in the mixture cluster were substantially different from other bipolar cells. (by KNN classifiers result.)





Real Data Analysis

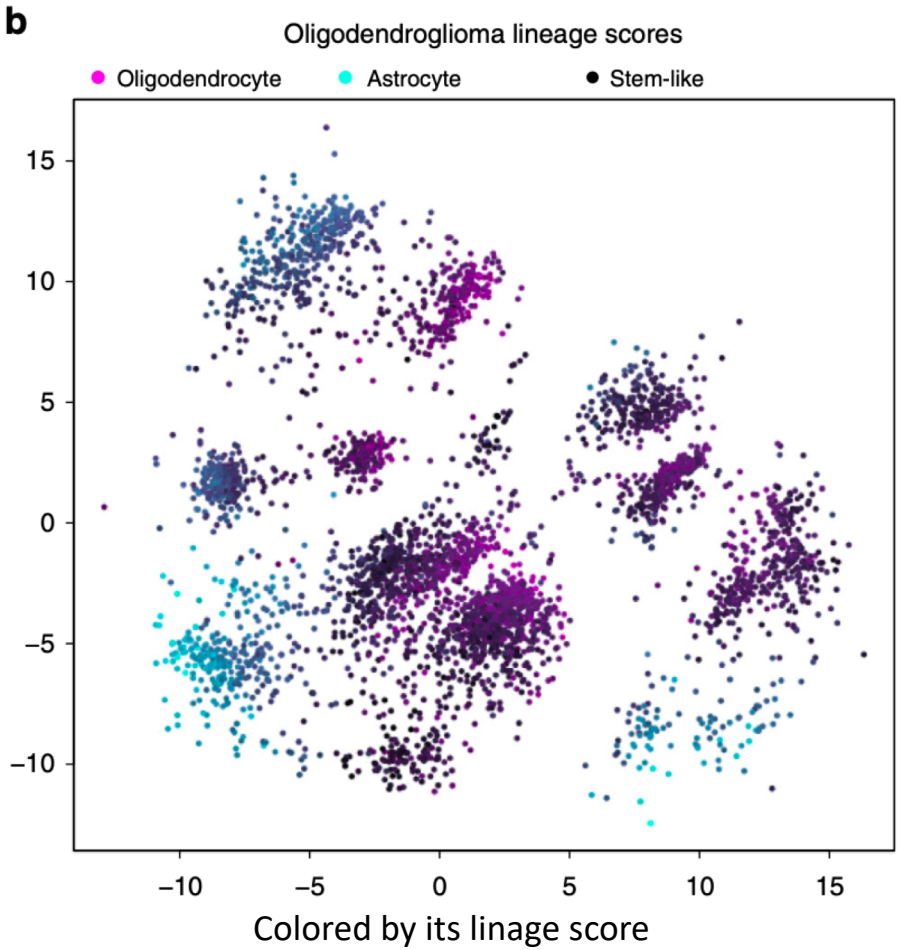
- Performance on the tumor microenvironments and intratumor heterogeneity (A. and B.)
  - A dataset consists of mostly malignant cells (includes oligodendrocyte, astrocyte, and stem-like cells).
  - 15 clusters are found by using densitycut based on the SCVIS embedding.



- Entropy is based on the cells of origins (a).
- High entropy: #1, #5, #12, #14, #15
- Malignant cells formed distinct clusters even if they were from the same patient.

Real Data Analysis

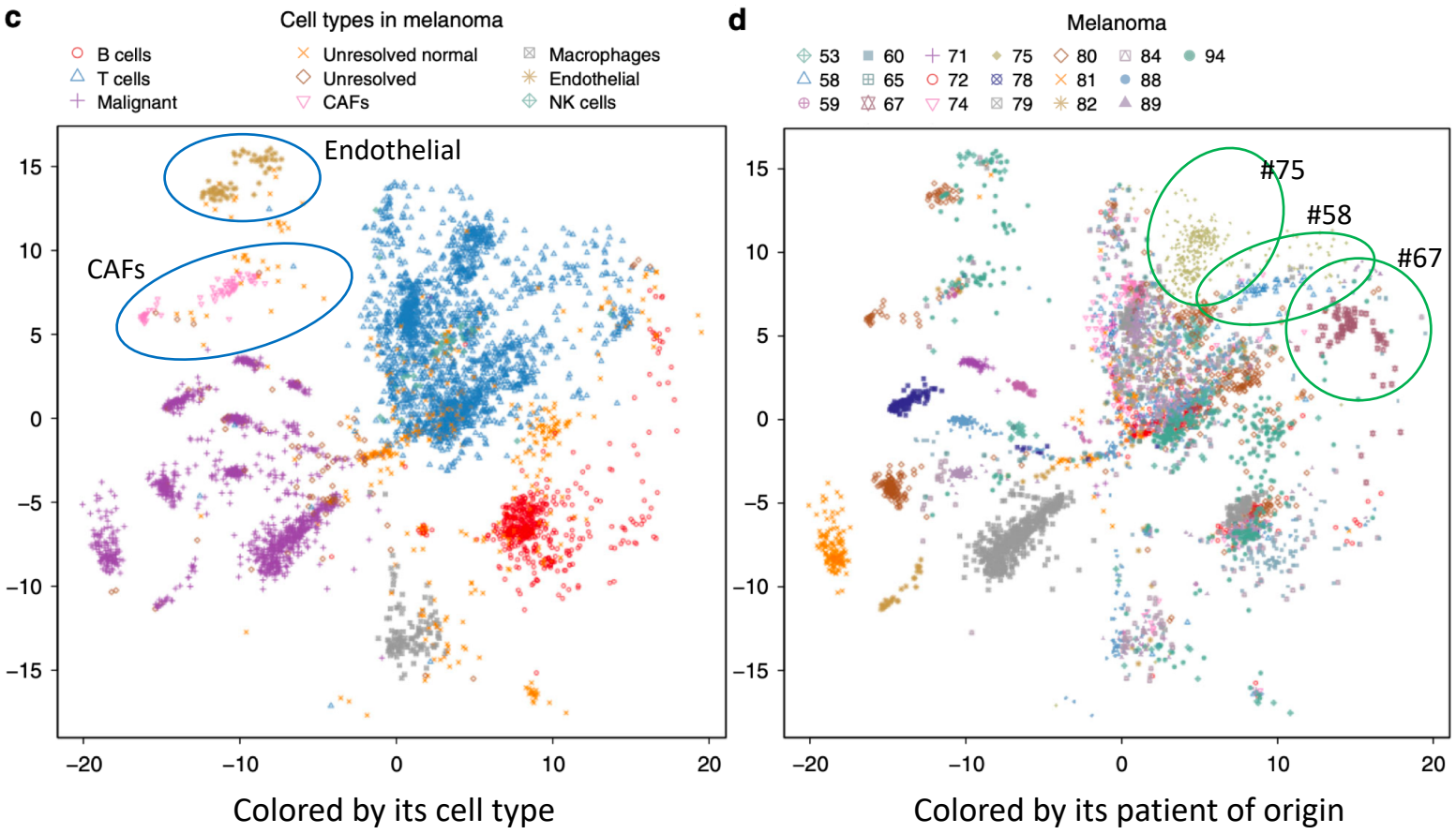
- Performance on the tumor microenvironments and intratumor heterogeneity (A. and B.)
  - A dataset embedding result colored by lineage score.



- Cells in some clusters highly expressed the astrocyte gene markers or the oligodendrocyte.
- The stem-like cells tends to be rare and could link outliers connecting oligodendrocyte and astrocyte cells.
- Some clusters consisted of mixture of cells
  → other factors (e.g. genetic mutation and epigenetic measurements) would be required to interpret the clustering structures furtherly.

Real Data Analysis

- Performance on the tumor microenvironments and intratumor heterogeneity (A. and B.)
  - B dataset consists malignant and non-malignant cells.



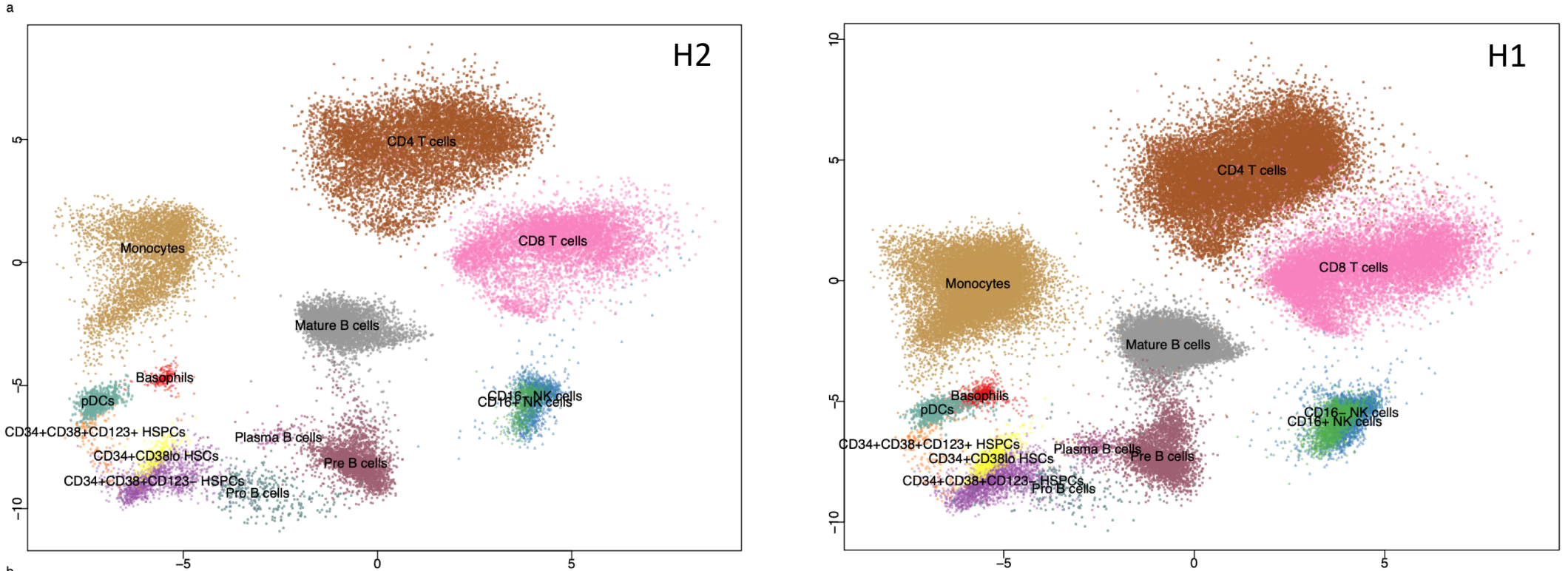
- Non-malignant immune cells (B cells, T cells) tend to be grouped together by cell types instead of patients of origin of cells.

- Some non-malignant cells (CAFs) were map to the region adjacent to the malignant cells (Endothelial).
  → CAFs and endothelial cells were truly more similar to malignant cells (by a hierarchical clustering analysis).

- Some immune cells of some patients showed patient-specific bias (#75, #58, #67)
  → Batch effect between patient 75 and other patient.

Real Data Analysis

- Performance on other types of single-cell data (E. and F.)
  - Learn the parametric mapping from the F. data and apply on the E. data (project into two-dimensional space.)
  - All the 14 cell types were separated and CD4 T cells & CD8 T cells are adjacent to each other.
  - The high quality of the mapping carried over on the H1 data.



Summary

Summary

- SCVIS is developed for modeling and reducing dimensionality of single-cell gene expression data.

- SCVIS provides:
  - A low-D embeddings of high-D data while preserving global structure of the high-D measurements.

  - The log-likelihood as a measure of the quality (uncertainty) of the embedding.
    - Detecting outliers.
    - An indicator of further analysis for adjacent/overlapping region.

- Compare with other methods, SCVIS can generate a non-linear and interpretable result at the same time.

Appendix

Variational Autoencoder (VAE)

- Some important property
  - Generate a linear combination of current datapoints in the database.
  - Provide a buffer area (sigma) in estimating current image. (Autoencoder does not provide this buffer.)

Variational Autoencoder (VAE)

- The Derivation from the Variational Inference

Variational Autoencoder (VAE)

- The Reparametrized Trick in Math