

2023 ICML Workshop on Computational Biology (Contributed Talk)

Single-cell RNA-seq data imputation using Feature Propagation

Sukwon Yun*, Junseok Lee*, Chanyoung Park

Korea Advanced Institute of Science and Technology (KAIST)



TABLE OF CONTENTS

▪ Background

- Single-cell RNA-seq
- As a Graph: Feature Propagation

▪ Motivation

- Single-cell RNA-seq data imputation using Feature Propagation

▪ scFP: single-cell Feature Propagation

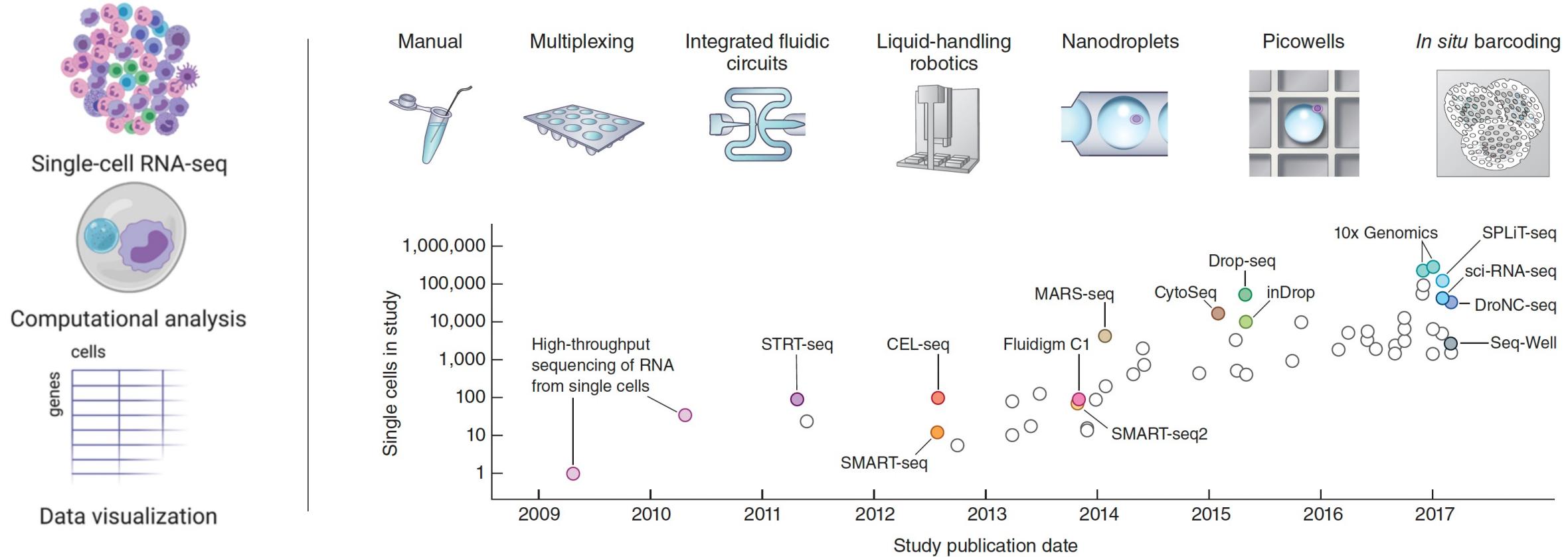
▪ Experiments

▪ Conclusion



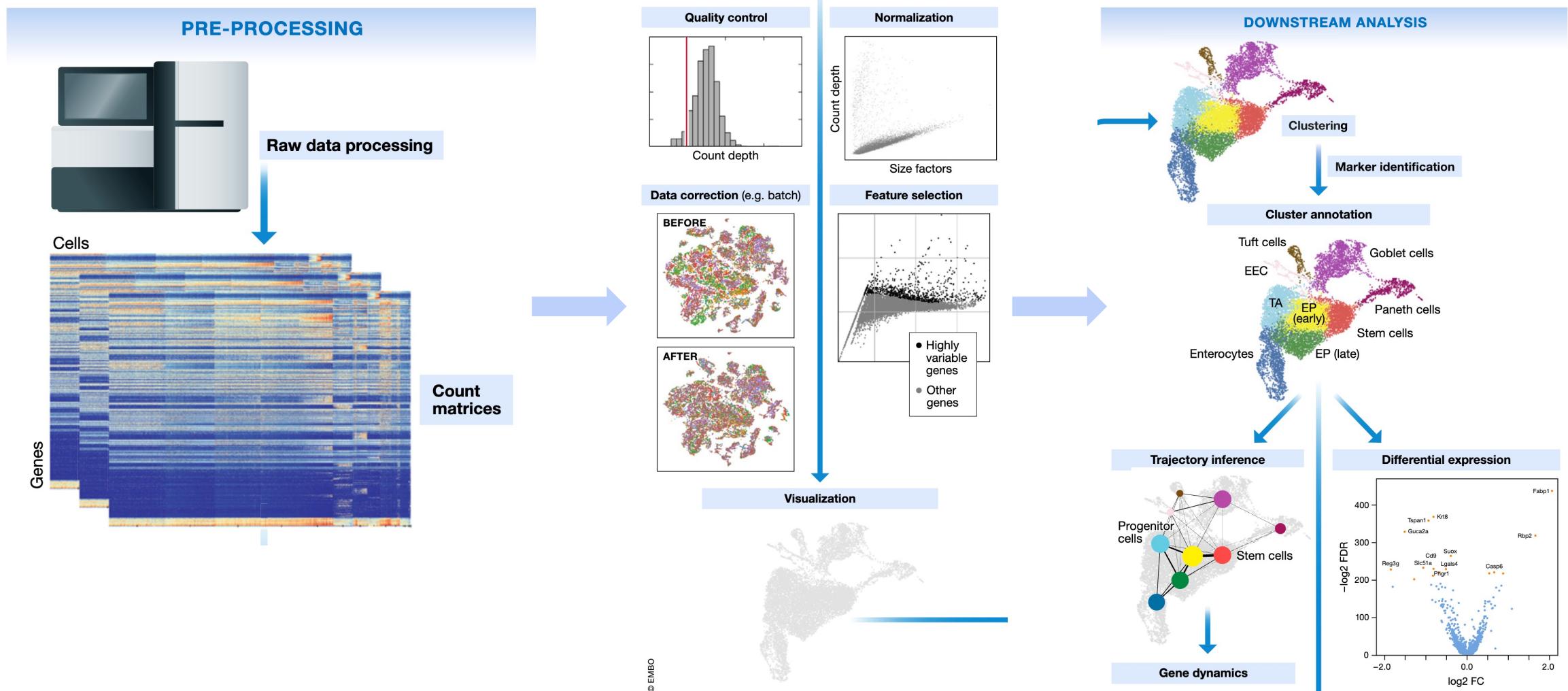
BACKGROUND: SINGLE-CELL RNA-SEQ

▪ Advances in single-cell RNA-seq



BACKGROUND: SINGLE-CELL RNA-SEQ

▪ Workflow of single-cell RNA-seq (scRNA-seq) analysis

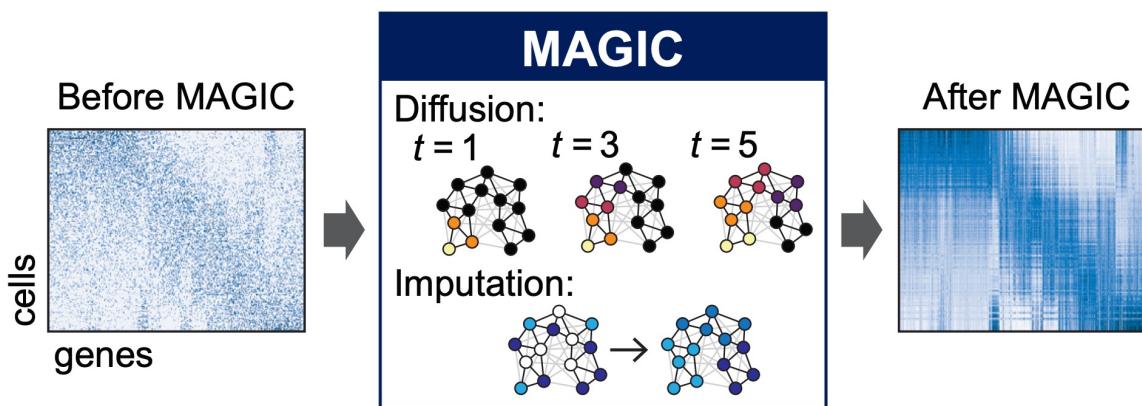


© EMBO

BACKGROUND: AS A GRAPH

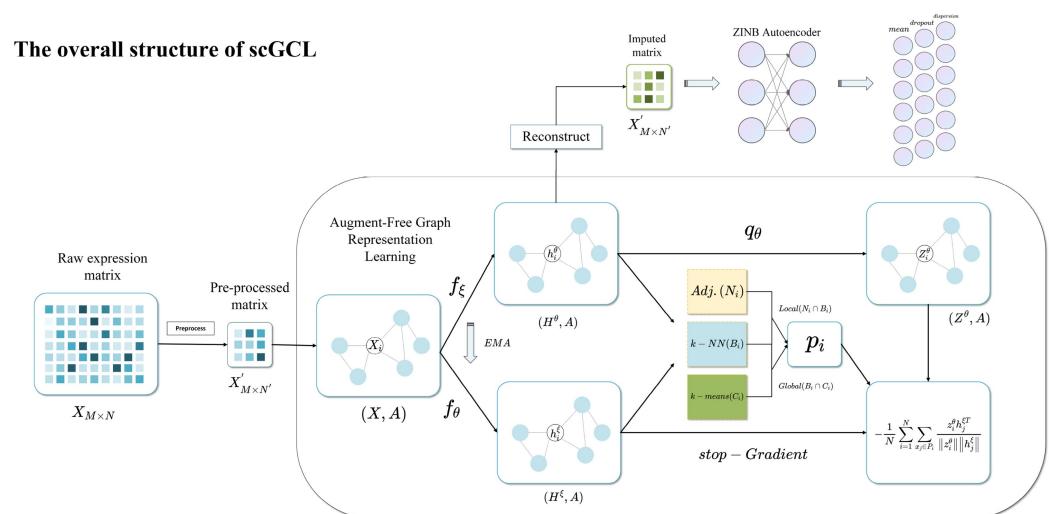
■ Cell-Gene Matrix as a Graph Structure

- Graphs facilitate clustering algorithms such as the min-cut algorithm (spectral clustering)
- Graphs enable a better understanding of paths of progression or trajectories of differentiation
- Graphs capture relationships among cells and facilitate message-passing schemes for information propagation



MAGIC (van Dijk et al., 2018)

The overall structure of scGCL

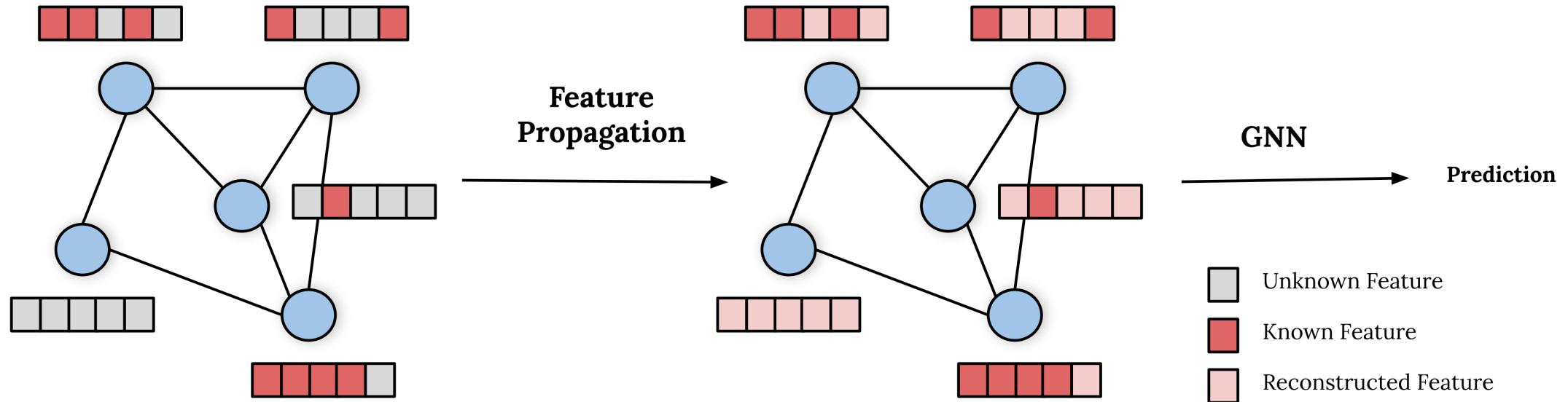


scGCL (Xiong et al., 2023)

BACKGROUND: AS A GRAPH

■ Feature Propagation (Rossi et al., 2022)

- **Motivation:** In many real-world applications, features are partially available
- **Idea:** General approach for handling missing features in graph machine learning based on minimizing Dirichlet energy



BACKGROUND: AS A GRAPH

■ Feature Propagation (Rossi et al., 2022)

- **Motivation:** In many real-world applications, features are partially available
- **Idea:** General approach for handling missing features in graph machine learning based on minimizing Dirichlet energy

Dirichlet Energy

Analytic Approach

$$\ell(\mathbf{x}, G) = \frac{1}{2} \mathbf{x}^\top \Delta \mathbf{x} = \frac{1}{2} \sum_{ij} \tilde{a}_{ij} (x_i - x_j)^2$$

Gradient flow

$$\dot{\mathbf{x}}(t) = -\nabla \ell(\mathbf{x}(t)) = -\Delta \mathbf{x}(t)$$

Heat Diffusion Equation

$$\dot{\mathbf{x}}(t) = -\Delta \mathbf{x}(t) \quad (\text{IC}) \quad \mathbf{x}(0) = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_u(0) \end{bmatrix} \quad (\text{BC}) \quad \mathbf{x}_k(t) = \mathbf{x}_k$$

$$\begin{bmatrix} \dot{\mathbf{x}}_k(t) \\ \dot{\mathbf{x}}_u(t) \end{bmatrix} = - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \Delta_{uk} & \Delta_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_u(t) \end{bmatrix} = - \begin{bmatrix} \mathbf{0} \\ \Delta_{uk} \mathbf{x}_k + \Delta_{uu} \mathbf{x}_u(t) \end{bmatrix}$$

$$\nabla_{\mathbf{x}_u} \ell = \mathbf{0} \longrightarrow \boxed{\mathbf{x}_u = -\Delta_{uu}^{-1} \Delta_{ku}^\top \mathbf{x}_k} \quad \mathcal{O}(|\mathcal{V}_u|^3)$$

BACKGROUND: AS A GRAPH

■ Feature Propagation (Rossi et al., 2022)

- **Motivation:** In many real-world applications, features are partially available
- **Idea:** General approach for handling missing features in graph machine learning based on minimizing Dirichlet energy

Dirichlet Energy

Analytic Approach

$$\ell(\mathbf{x}, G) = \frac{1}{2} \mathbf{x}^\top \Delta \mathbf{x} = \frac{1}{2} \sum_{ij} \tilde{a}_{ij} (x_i - x_j)^2$$

Gradient flow

$$\dot{\mathbf{x}}(t) = -\nabla \ell(\mathbf{x}(t)) = -\Delta \mathbf{x}(t)$$

Heat Diffusion Equation

$$\dot{\mathbf{x}}(t) = -\Delta \mathbf{x}(t) \quad (\text{IC}) \quad \mathbf{x}(0) = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_u(0) \end{bmatrix} \quad (\text{BC}) \quad \mathbf{x}_k(t) = \mathbf{x}_k$$

$$\begin{bmatrix} \dot{\mathbf{x}}_k(t) \\ \dot{\mathbf{x}}_u(t) \end{bmatrix} = - \begin{bmatrix} 0 & 0 \\ \Delta_{uk} & \Delta_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_u(t) \end{bmatrix} = - \begin{bmatrix} 0 \\ \Delta_{uk} \mathbf{x}_k + \Delta_{uu} \mathbf{x}_u(t) \end{bmatrix}$$

$$\nabla_{\mathbf{x}_u} \ell = \mathbf{0} \longrightarrow \mathbf{x}_u = -\Delta_{uu}^{-1} \Delta_{ku}^\top \mathbf{x}_k$$

Euler Scheme

Iterative Approach

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - h \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \Delta_{uk} & \Delta_{uu} \end{bmatrix} \mathbf{x}^{(k)}$$

$$= \left(\mathbf{I} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ h\Delta_{uk} & h\Delta_{uu} \end{bmatrix} \right) \mathbf{x}^{(k)} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -h\Delta_{uk} & \mathbf{I} - h\Delta_{uu} \end{bmatrix} \mathbf{x}^{(k)}$$

when $h = 1$,

$$\tilde{\mathbf{A}} = \mathbf{I} - \Delta = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \Delta_{kk} & \Delta_{ku} \\ \Delta_{uk} & \Delta_{uu} \end{bmatrix} = \begin{bmatrix} \mathbf{I} - \Delta_{kk} & -\Delta_{ku} \\ -\Delta_{uk} & \mathbf{I} - \Delta_{uu} \end{bmatrix}$$

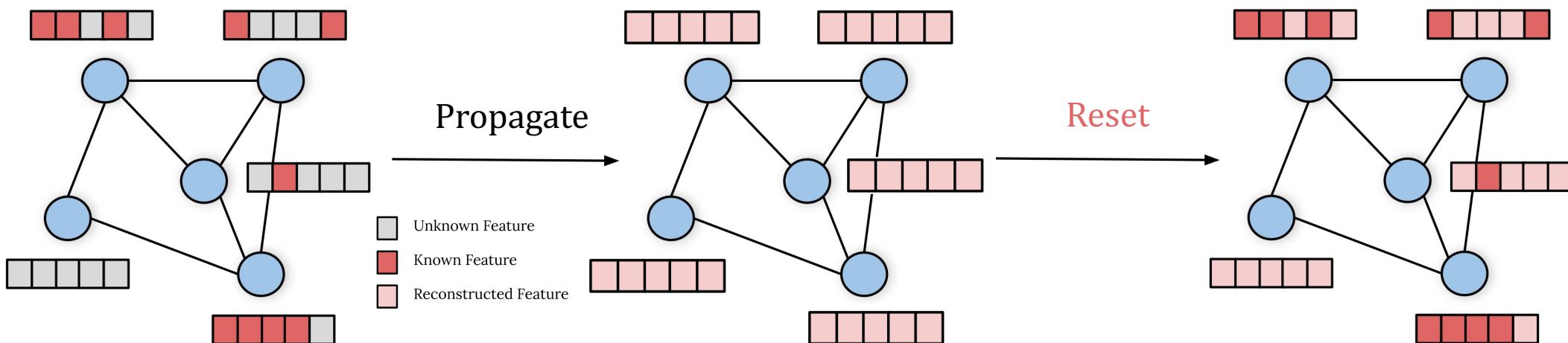
$$\mathbf{x}^{(k+1)} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \tilde{\mathbf{A}}_{uk} & \tilde{\mathbf{A}}_{uu} \end{bmatrix} \mathbf{x}^{(k)}$$

BACKGROUND: AS A GRAPH

- Feature Propagation (Rossi et al., 2022)

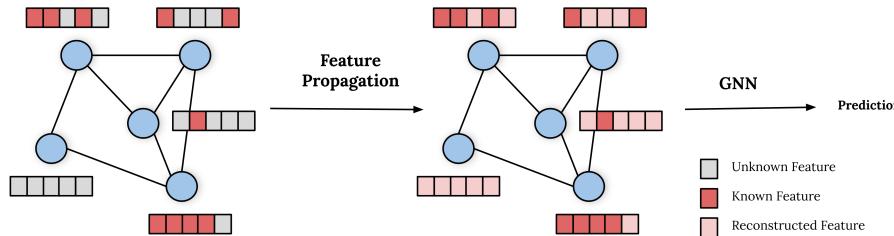
- **Motivation:** In many real-world applications, features are partially available
- **Idea:** General approach for handling missing features in graph machine learning based on minimizing Dirichlet energy

$$\mathbf{x}^{(k+1)} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \tilde{\mathbf{A}}_{uk} & \tilde{\mathbf{A}}_{uu} \end{bmatrix} \mathbf{x}^{(k)} \quad \left| \begin{array}{l} 1) \text{ Propagate: } \mathbf{x}^{(k+1)} = \tilde{\mathbf{A}}\mathbf{x}^{(k)} = \begin{bmatrix} \tilde{\mathbf{A}}_{kk} & \tilde{\mathbf{A}}_{ku} \\ \tilde{\mathbf{A}}_{uk} & \tilde{\mathbf{A}}_{uu} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_k^{(k)} \\ \mathbf{x}_u^{(k)} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}}_{kk}\mathbf{x}_k^{(k)} + \tilde{\mathbf{A}}_{ku}\mathbf{x}_u^{(k)} \\ \tilde{\mathbf{A}}_{uk}\mathbf{x}_k^{(k)} + \tilde{\mathbf{A}}_{uu}\mathbf{x}_u^{(k)} \end{bmatrix} \\ 2) \text{ Reset: } \mathbf{x}_k^{(k+1)} = \mathbf{x}_k^{(0)} \rightarrow \mathbf{x}^{(k+1)} = \begin{bmatrix} \mathbf{x}_k^{(0)} \\ \tilde{\mathbf{A}}_{uk}\mathbf{x}_k^{(k)} + \tilde{\mathbf{A}}_{uu}\mathbf{x}_u^{(k)} \end{bmatrix} \end{array} \right.$$



MOTIVATION: scRNA-SEQ DATA IMPUTATION USING FEATURE PROPAGATION

- Research Direction: Feature Propagation on scRNA-seq data



Euler Scheme

Iterative Approach

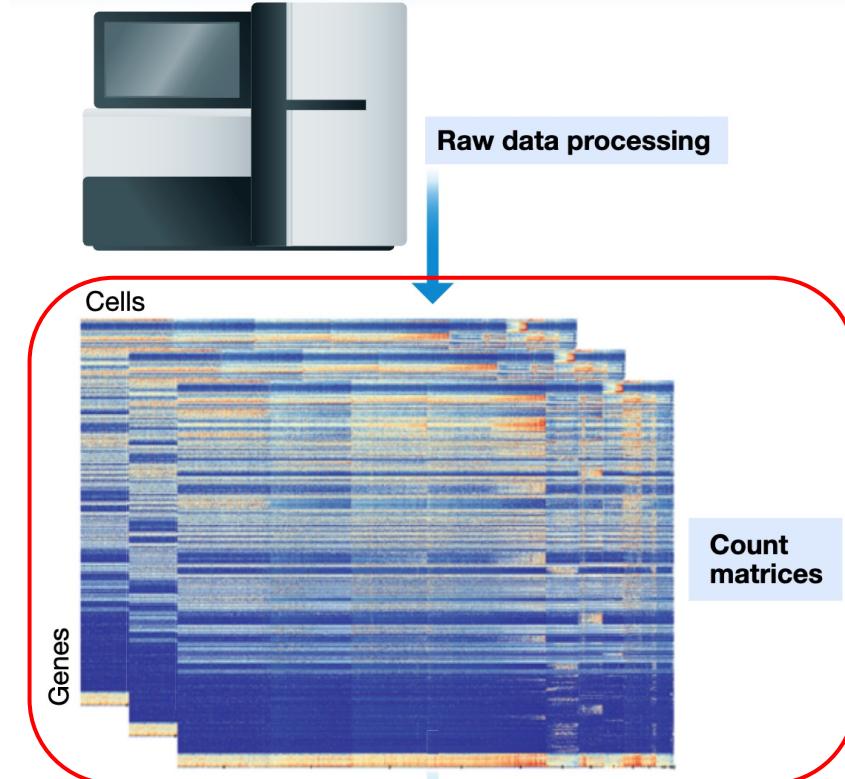
$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - h \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \Delta_{uk} & \Delta_{uu} \end{bmatrix} \mathbf{x}^{(k)} \\ &= \left(\mathbf{I} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ h\Delta_{uk} & h\Delta_{uu} \end{bmatrix} \right) \mathbf{x}^{(k)} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -h\Delta_{uk} & \mathbf{I} - h\Delta_{uu} \end{bmatrix} \mathbf{x}^{(k)} \end{aligned}$$

when $h = 1$,

$$\tilde{\mathbf{A}} = \mathbf{I} - \Delta = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \Delta_{kk} & \Delta_{ku} \\ \Delta_{uk} & \Delta_{uu} \end{bmatrix} = \begin{bmatrix} \mathbf{I} - \Delta_{kk} & -\Delta_{ku} \\ -\Delta_{uk} & \mathbf{I} - \Delta_{uu} \end{bmatrix}$$

$$\mathbf{x}^{(k+1)} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \tilde{\mathbf{A}}_{uk} & \tilde{\mathbf{A}}_{uu} \end{bmatrix} \mathbf{x}^{(k)}$$

* Challenges

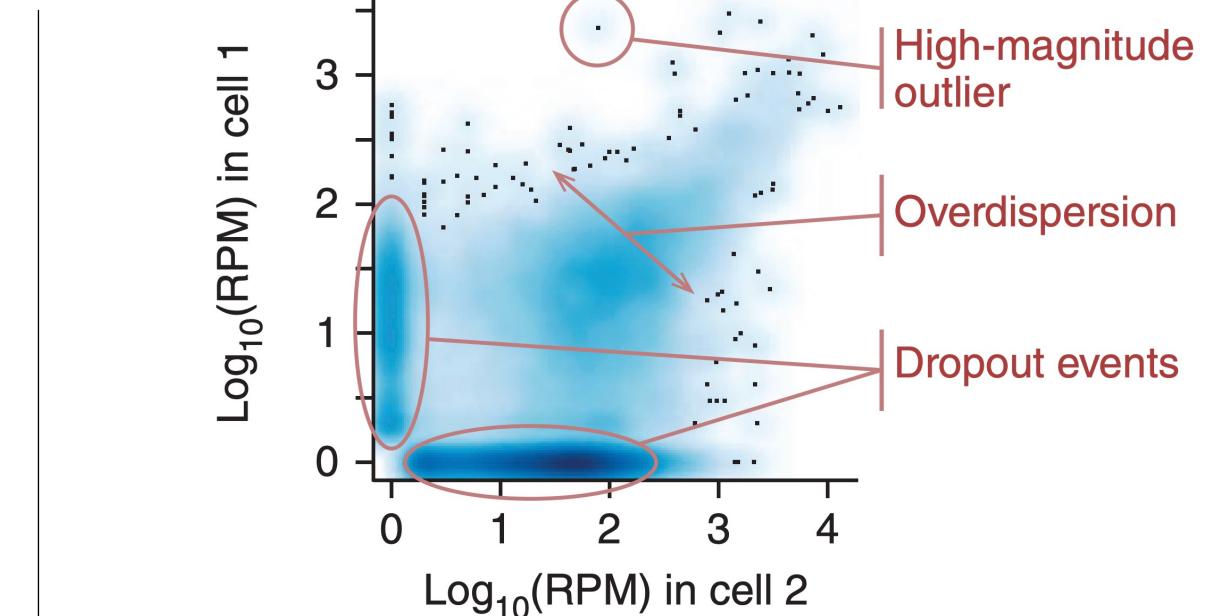
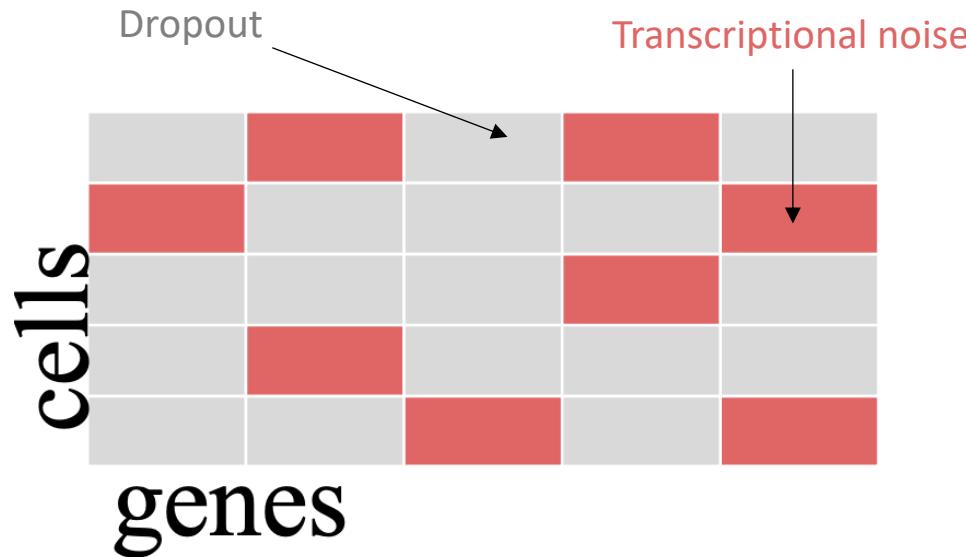


1. The information regarding which features are missing or noisy is not provided
2. Biologically relevant graph structure is not provided

MOTIVATION: scRNA-SEQ DATA IMPUTATION USING FEATURE PROPAGATION

- Challenge 1) Missing and noise in cell-gene matrix

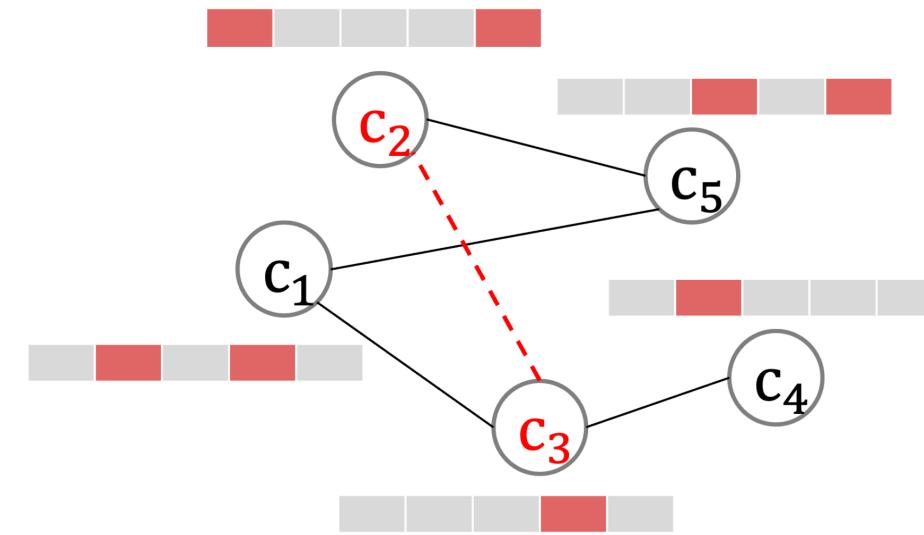
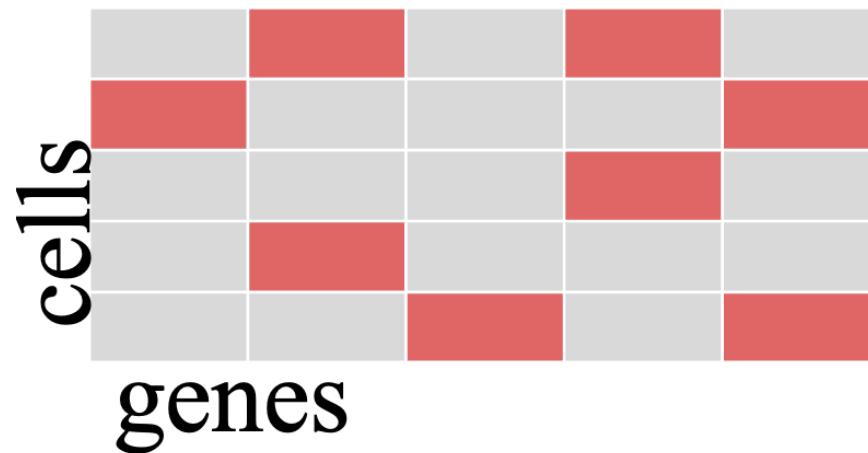
- Zero-values (Missing): Often regarded as a dropout (e.g., false-zeros)
- Non-zero values (Noise): Might capture biologically irrelevant signals (e.g., batch effects, transcriptional noise)



Careful handling of both zero-values and non-zero values is crucial

MOTIVATION: scRNA-SEQ DATA IMPUTATION USING FEATURE PROPAGATION

- Challenge 2) Biologically relevant graph structure is not provided
 - k NN Graph based on initial sparse matrix may not be optimal

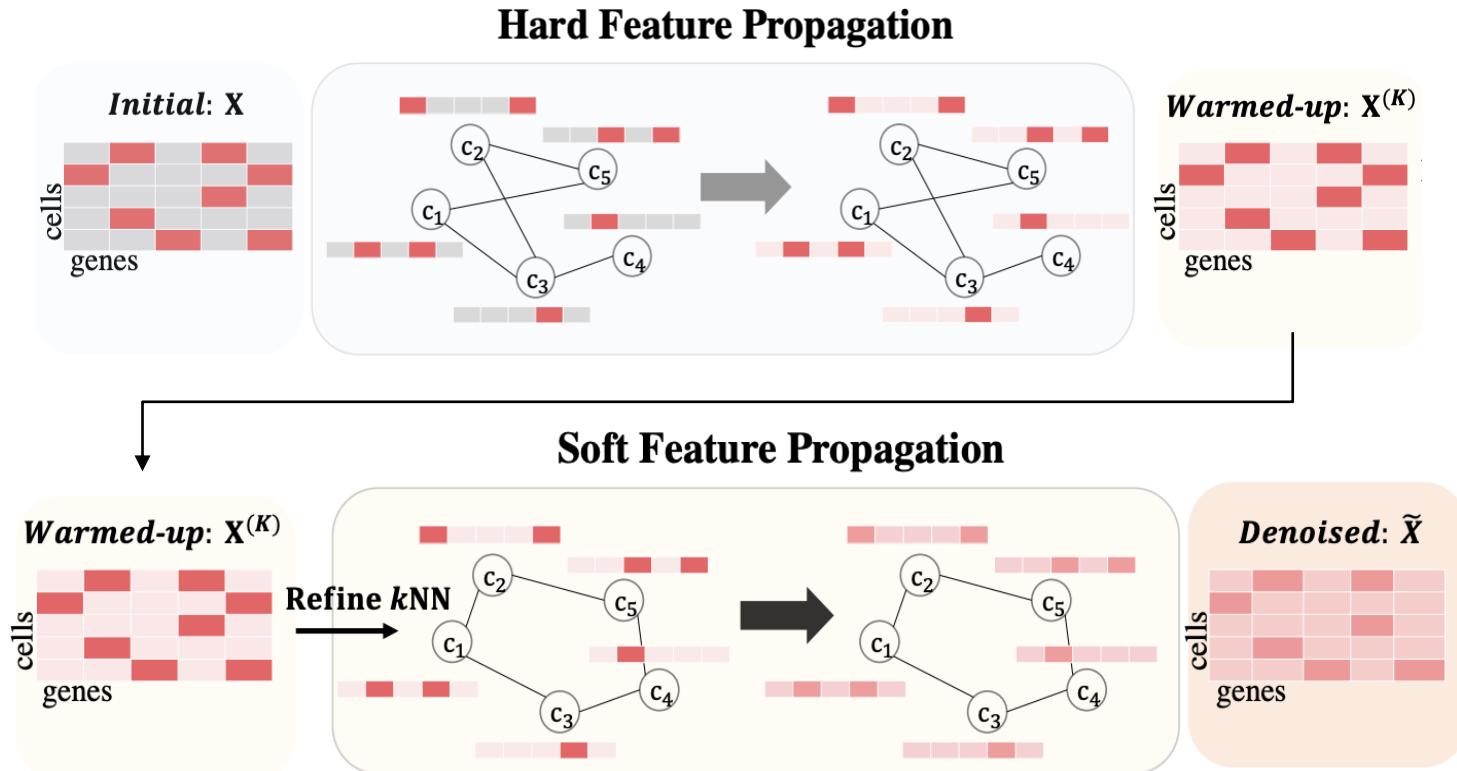


< k NN Graph on sparse cell-gene matrix >

When generating a graph, it is essential to carefully consider the biologically relevant relationships among cells

scFP: single-cell FEATURE PROPAGATION

- Overall Framework of scFP

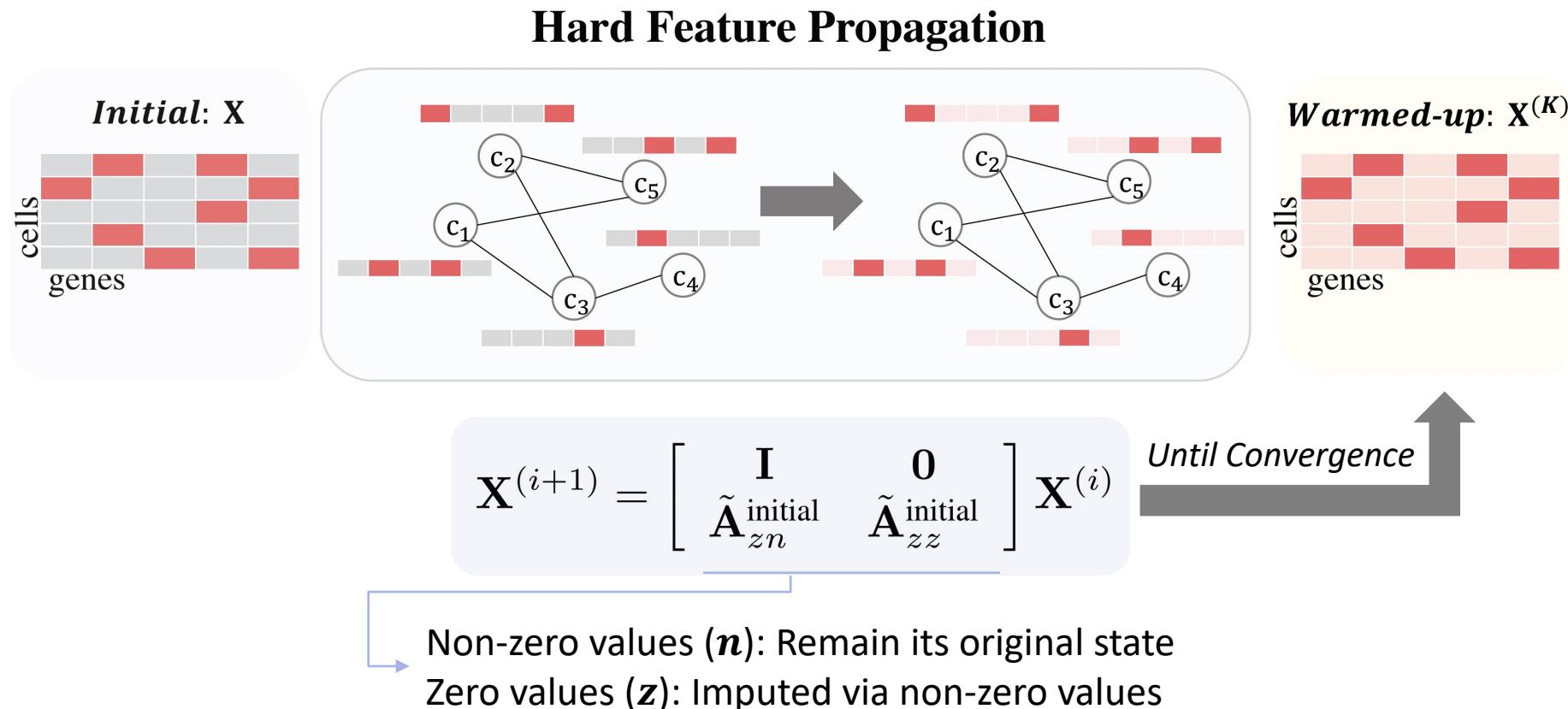


- 1) Hard Feature Propagation
- 2) Refine kNN Graph
- 3) Soft Feature Propagation

scFP: single-cell FEATURE PROPAGATION

▪ 1) Hard Feature Propagation

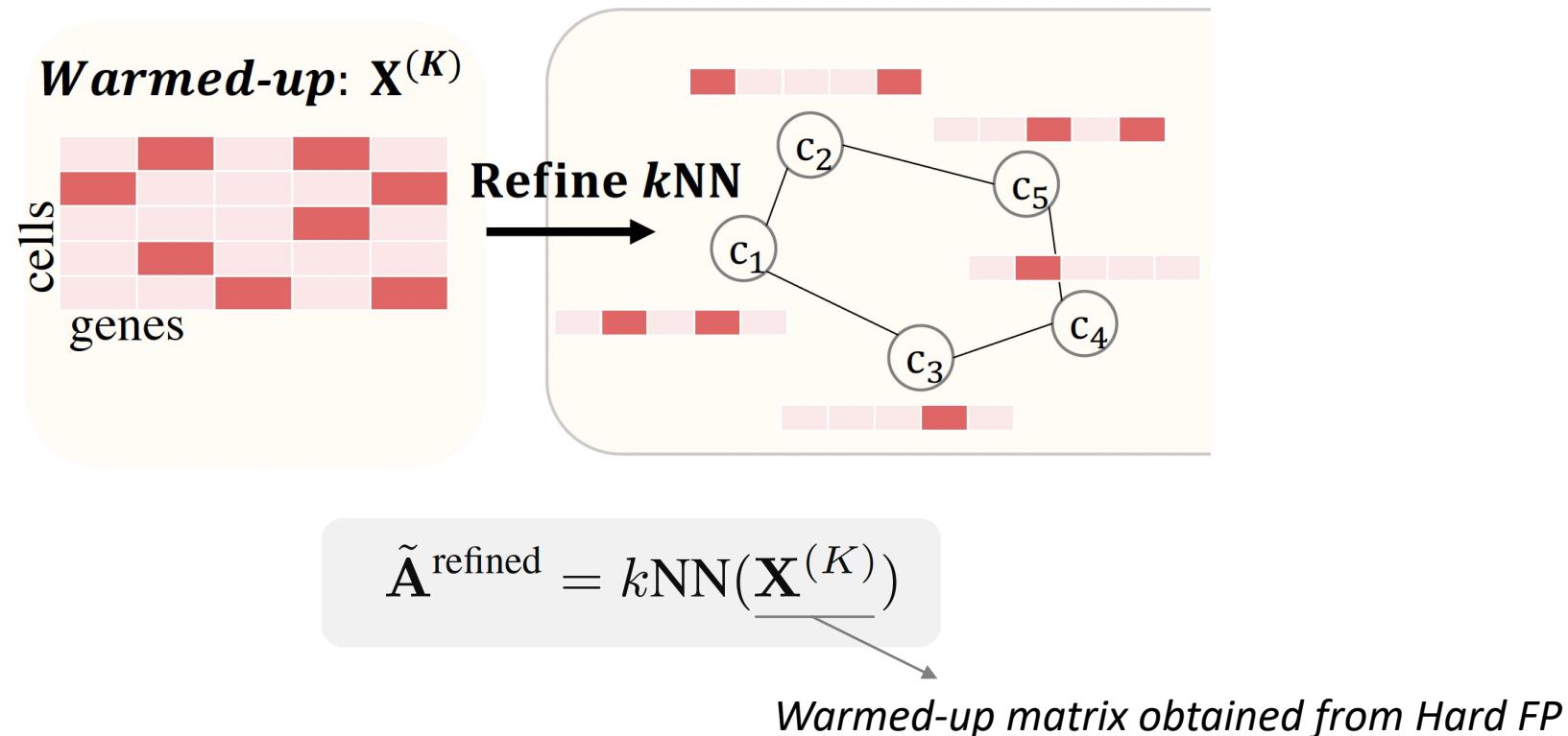
- Impute zero values (dropout) via observed gene expression and obtain warmed-up cell-gene matrix
- *Assumption: Imputing zeros (dropout) is more significant than denoising non-zeros at the initial stage*



scFP: single-cell FEATURE PROPAGATION

▪ 2) Refine k NN Graph

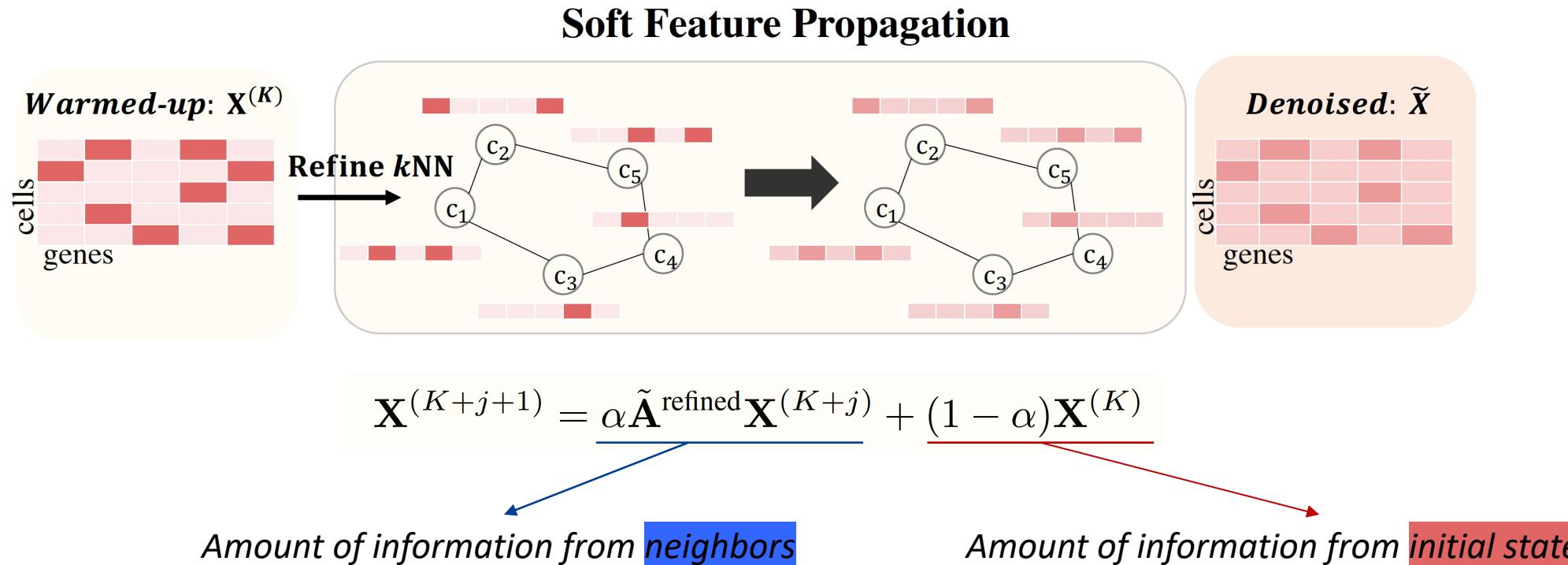
- Build k NN Graph via warmed-up cell-gene matrix
- Compared to initial k NN Graph, it would potentially reveal hidden or implicit graph structures



scFP: single-cell FEATURE PROPAGATION

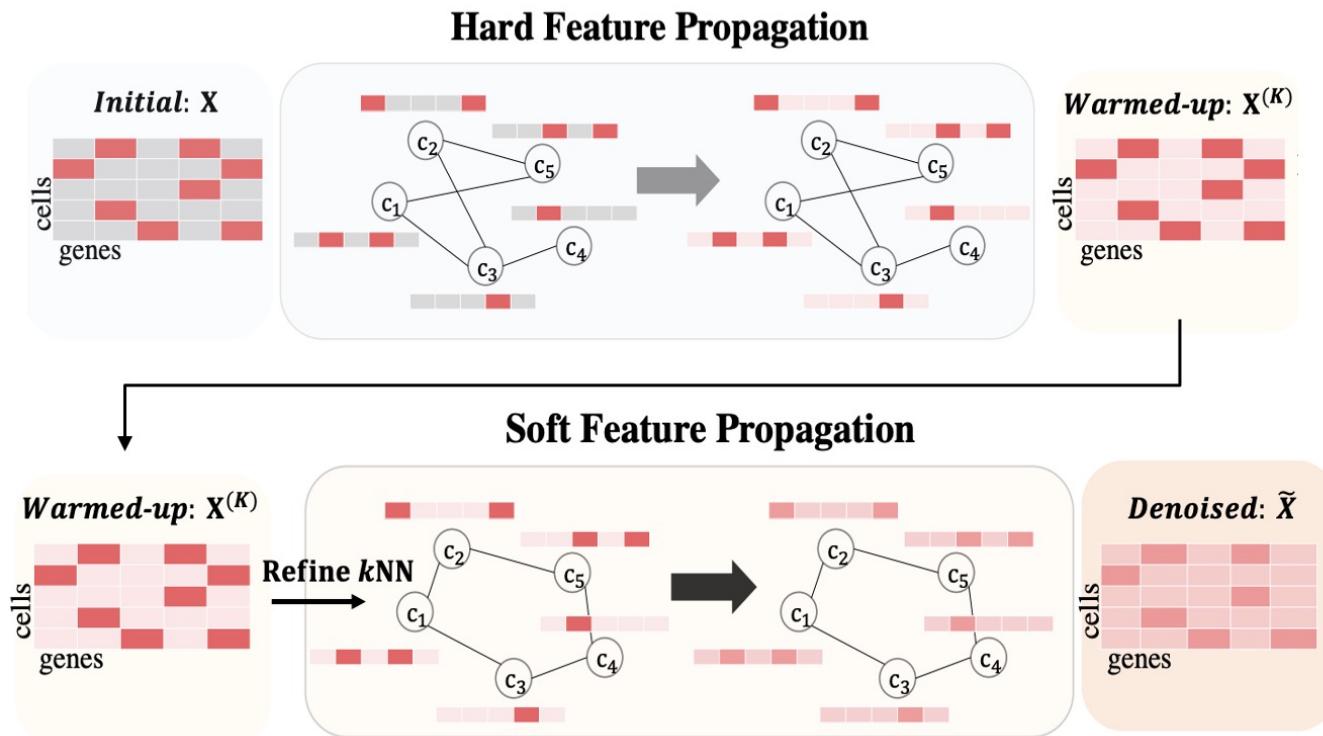
▪ 3) Soft Feature Propagation

- Denoise observed gene expression (irrelevant signals)
- Focus on updating non-zero values → used constant α as 0.99 during experiments



scFP: single-cell FEATURE PROPAGATION

- In a nutshell,



Algorithm 1 single-cell Feature Propagation (scFP)

Input: Cell-Gene Matrix X , Initial $k\text{NN}$ $\tilde{\mathbf{A}}^{\text{initial}}$

Output: Denoised Cell-Gene Matrix $\tilde{\mathbf{X}}$

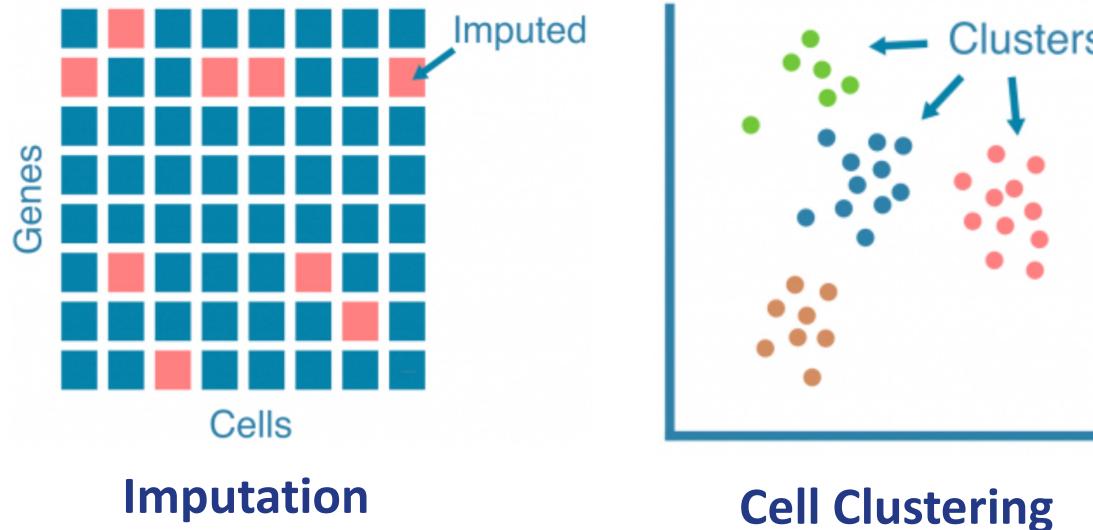
```
1:  $\mathbf{Y} \leftarrow \mathbf{X}$ 
2: while  $\mathbf{X}$  has not converged do
3:    $\mathbf{X} \leftarrow \tilde{\mathbf{A}}^{\text{initial}} \mathbf{X}$ 
4:    $\mathbf{X}_{k,d} \leftarrow \mathbf{Y}_{k,d} \forall k \in \mathcal{V}_{k,d}, \forall d \leq M$            Hard Clamping
5: end while
6:  $\tilde{\mathbf{A}}^{\text{refined}} = k\text{NN}(\mathbf{X}^{(K)})$                                 Refine kNN
7: while  $\mathbf{X}^{(K)}$  has not converged do
8:    $\mathbf{X}^{(K)} \leftarrow \alpha \tilde{\mathbf{A}}^{\text{refined}} \mathbf{X}^{(K)} + (1 - \alpha) \mathbf{X}^{(K)}$  Soft Clamping
9: end while
```

Impute zeros (Dropouts) → Refine $k\text{NN}$ Graph → Denoise non-zeros (Irrelevant signals)

EXPERIMENTS

▪ Data Statistics & Evaluation Metrics

Data	# of Cells	# of Genes	# of Subgroups
Baron Mouse	1,886	14,861	13
Mouse ES cells	2,717	24,047	4
Mouse bladder cells	2,746	19,771	16
Zeisel	3,005	19,972	7
Baron Human	8,569	20,125	14
Shekhar mouse retina cells	27,499	13,166	19



1) Imputation Task

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N y_i - \hat{y}_i}$$

N: # of cells
y_i: ground-truth gene expression i-th cell
ŷ_i: predicted gene expression i-th cell

2) Cell Clustering Task

- Adjusted Rand Index (ARI)

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \quad RI = \frac{a + b}{NC_2}$$

a: # of pairs successfully belong to the same cluster
b: # of pairs correctly labeled as different cluster

- Normalized Mutual Information (NMI)

$$NMI = \frac{2 \times I(S; C)}{[H(S) + H(C)]}$$

S: ground-truth cell type
C: cluster assignment by model
I(·, ·): mutual information
H(·): entropy

- Clustering Accuracy (CA)

$$CA = \max_m \frac{\sum_{i=1}^N \mathbb{1}_{[s_i = m(c_i)]}}{N}$$

N: # of cells
m(·): matching function
s_i: ground-truth cell type of i-th cell
c_i: cluster assignment of i-th cell

EXPERIMENTS

- Performance on imputation and cell clustering task

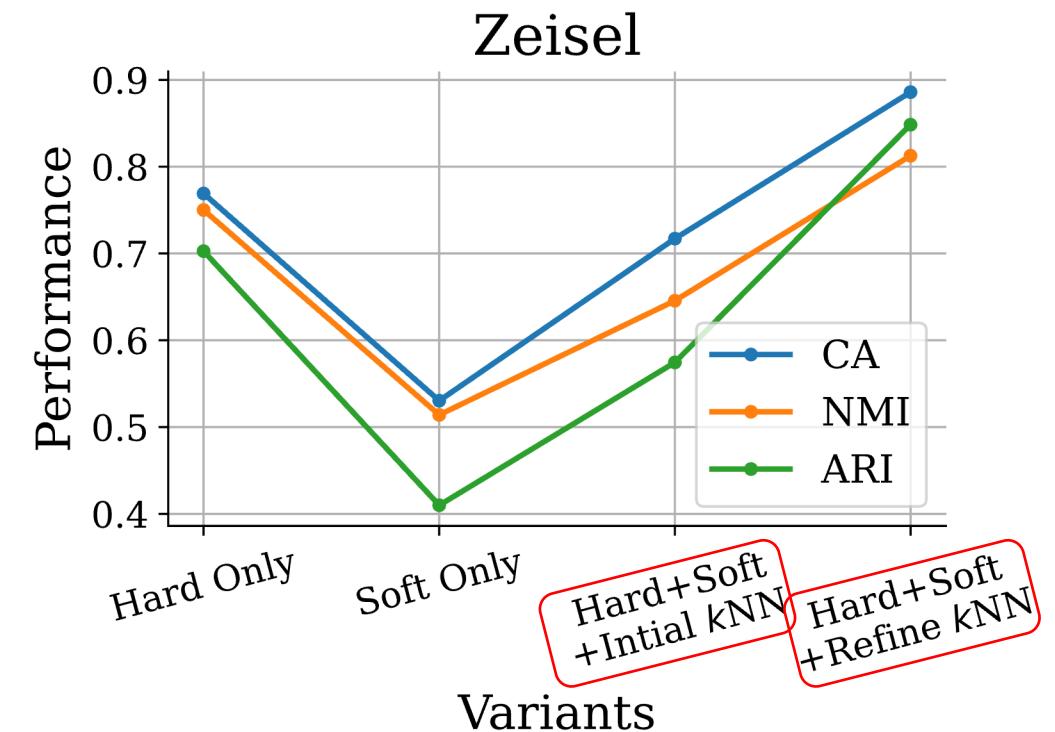
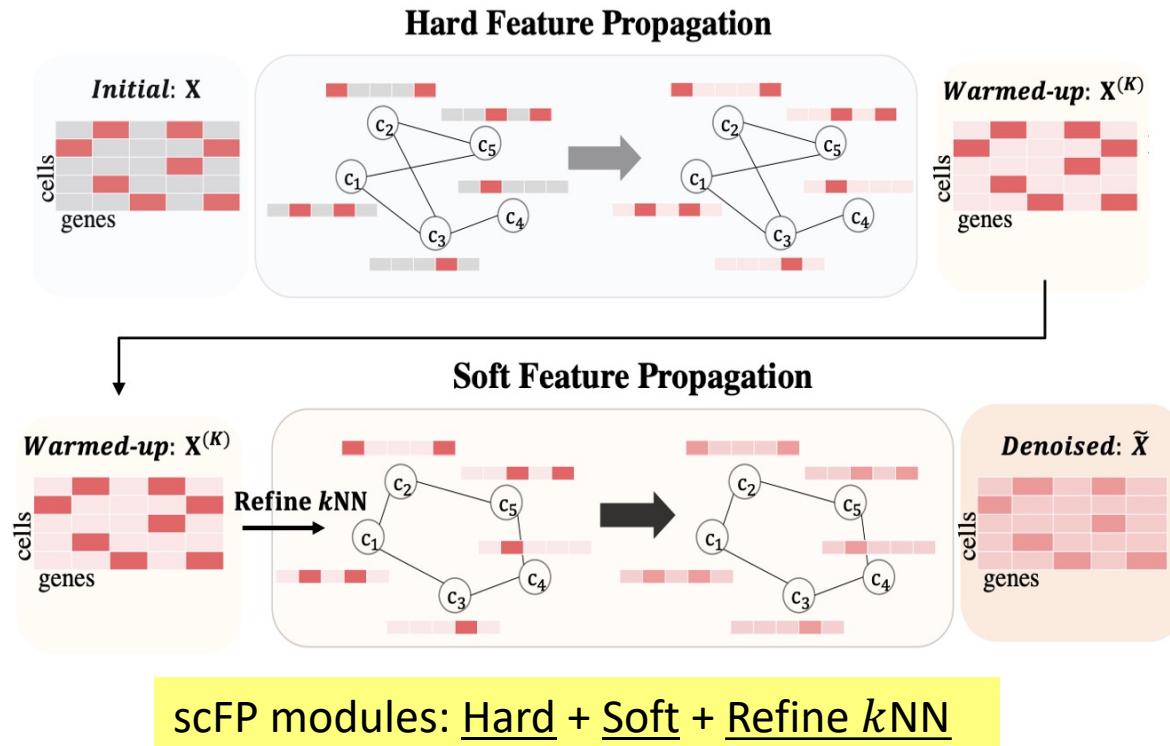
Imputation	Baron Mouse			Mouse ES			Mouse Bladder			Zeisel			Baron Human		
	Dropout Rates														
	20%	40%	80%	20%	40%	80%	20%	40%	80%	20%	40%	80%	20%	40%	80%
MAGIC	0.61	0.73	0.99	0.53	0.73	1.21	0.50	0.60	0.82	0.60	0.82	1.31	0.58	0.74	1.06
DCA	0.42	0.43	0.49	0.35	0.35	0.36	0.37	0.38	0.41	0.39	0.42	0.44	0.41	0.43	0.47
AutoClass	0.63	0.76	0.98	0.53	0.75	1.23	0.52	0.64	0.82	0.60	0.84	1.32	0.59	0.76	1.08
scGCL	0.64	0.74	0.97	0.59	0.75	1.16	0.51	0.62	0.81	0.66	0.82	1.29	0.63	0.77	1.08
scFP (Ours)	0.36	0.37	0.43	0.32	0.32	0.36	0.26	0.26	0.31	0.39	0.40	0.44	0.33	0.34	0.39

Cell Clustering	Baron Mouse			Mouse ES			Mouse Bladder			Zeisel			Baron Human		
	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA
Raw	0.44	0.71	0.56	0.74	0.75	0.79	0.59	0.75	0.68	0.70	0.75	0.77	0.44	0.71	0.56
MAGIC	0.42	0.72	0.57	0.80	0.85	0.83	0.55	0.75	0.64	0.70	0.75	0.76	0.56	0.78	0.59
DCA	0.46	0.69	0.59	0.76	0.78	0.81	0.39	0.59	0.54	0.67	0.72	0.75	0.53	0.74	0.55
AutoClass	0.44	0.71	0.52	0.74	0.75	0.81	0.51	0.75	0.64	0.71	0.75	0.77	0.44	0.71	0.52
scGCL	0.43	0.72	0.54	0.73	0.75	0.79	0.53	0.75	0.64	0.65	0.70	0.73	0.50	0.78	0.62
kNN-smoothing	0.43	0.72	0.55	0.72	0.74	0.79	0.59	0.76	0.68	0.68	0.73	0.76	0.56	0.78	0.56
scFP (Ours)	0.61	0.82	0.76	0.82	0.83	0.85	0.65	0.77	0.73	0.85	0.81	0.89	0.68	0.83	0.73

The denoised matrix obtained via scFP shows promising results in both imputation and the cell clustering task

EXPERIMENTS

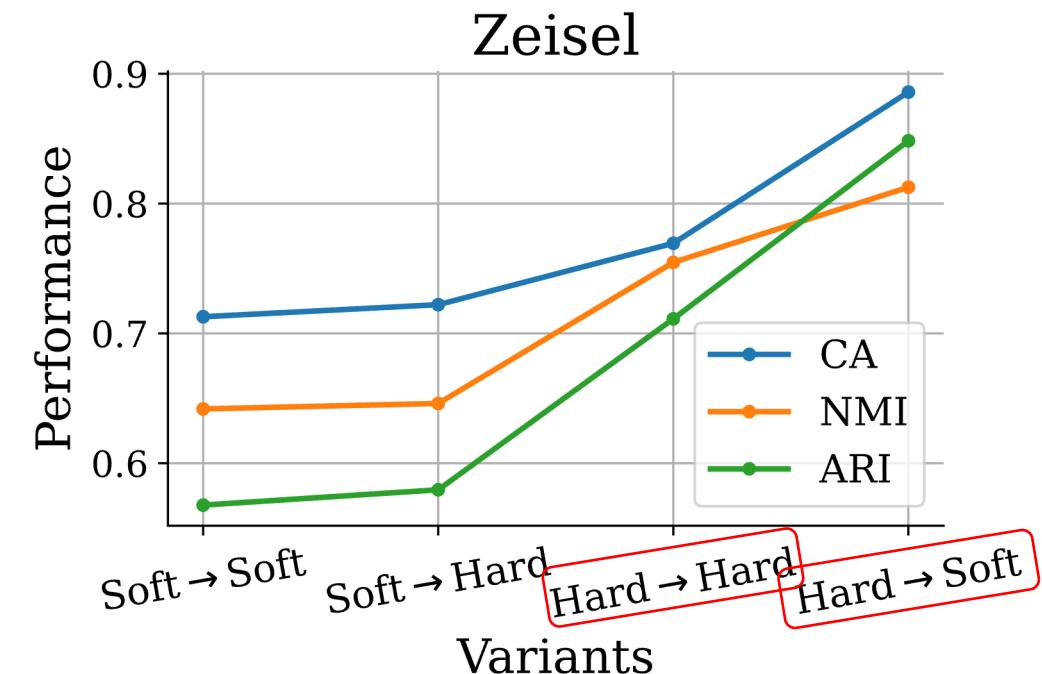
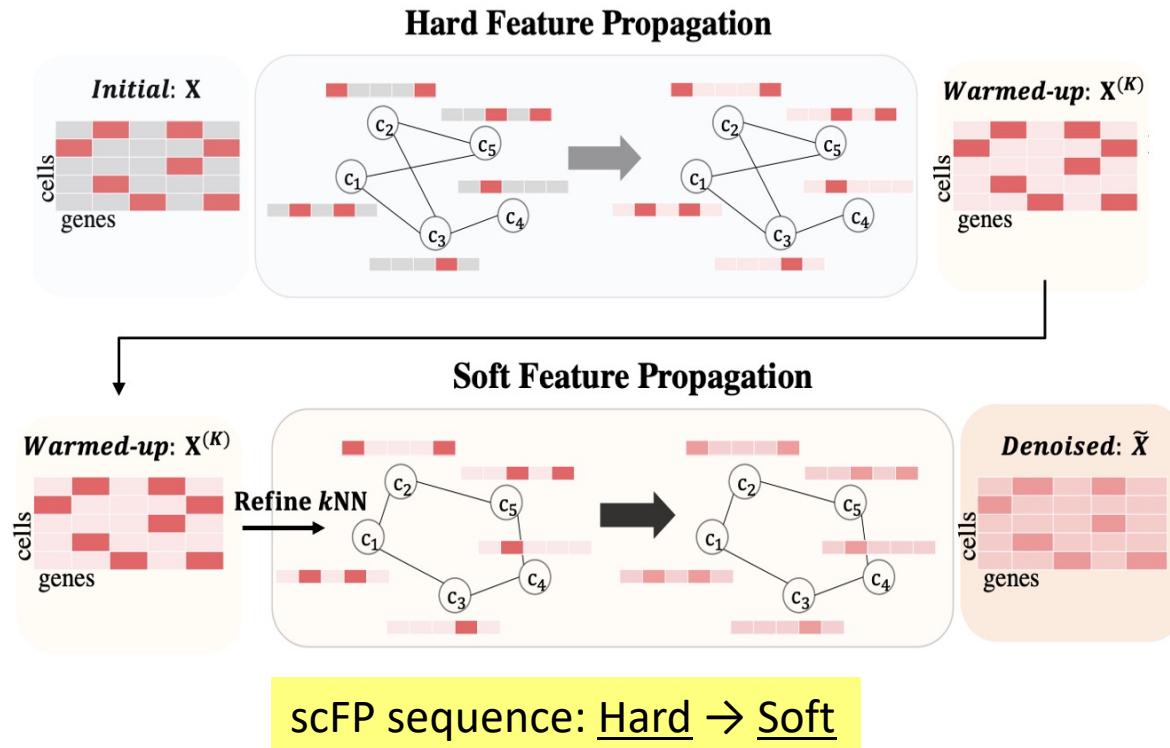
- Ablation on each module in scFP



- 1) (*H+S+Refine kNN* outperforms *H, S only*): Using both Hard and Soft FP is beneficial
- 2) (*H+S+Refine kNN* outperforms *H+S+Initial kNN*): Utilization of a refined kNN graph prior to applying Soft FP is essential

EXPERIMENTS

- Ablation on sequence of scFP



- ($H \rightarrow \bullet$ outperforms $S \rightarrow \bullet$): Initially, importance of imputing zeros surpasses the significance of denoising non-zeros
- ($H \rightarrow S$ outperforms $H \rightarrow H$): Inclusion of Soft FP after Hard FP further enhances obtaining a denoised matrix

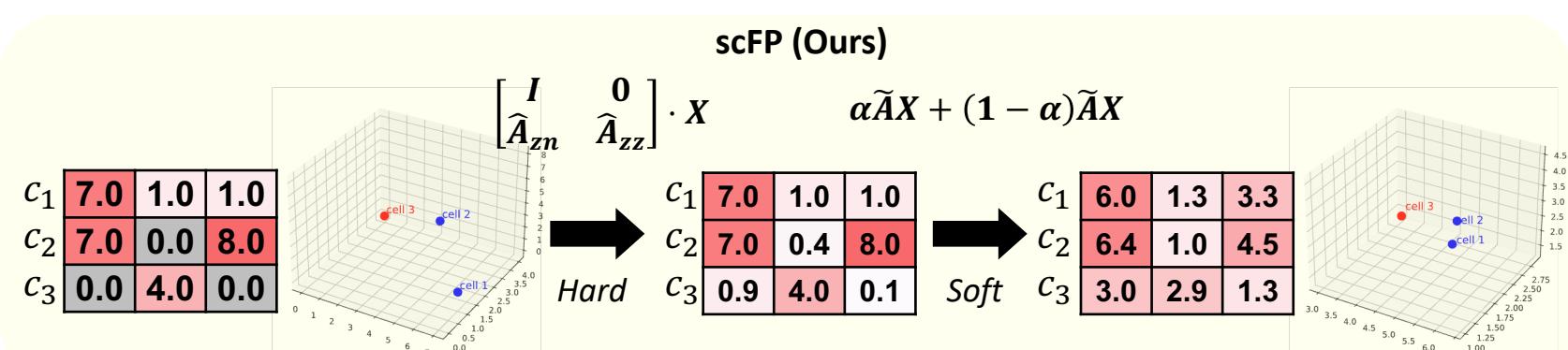
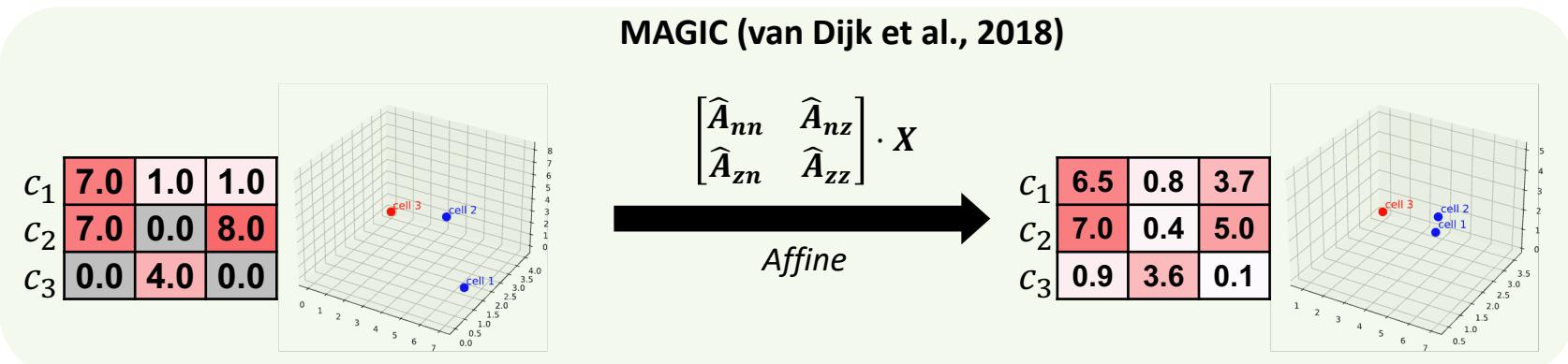
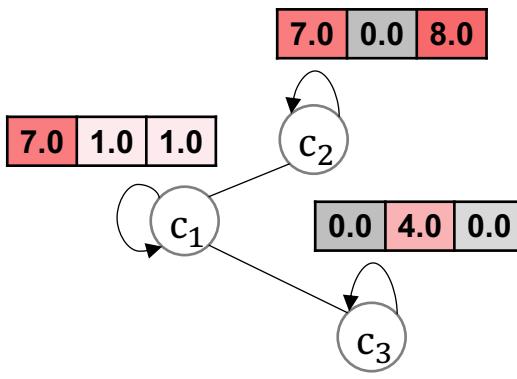
EXPERIMENTS

- Simulation Study: Risk of diffusion of false-zeros (MAGIC vs scFP) – Low dropout rates

7.0	1.0	1.0
7.0	0.0	8.0
0.0	4.0	0.0

genes
cells
< Cell-Gene Count Matrix (Dropout: 33%) >

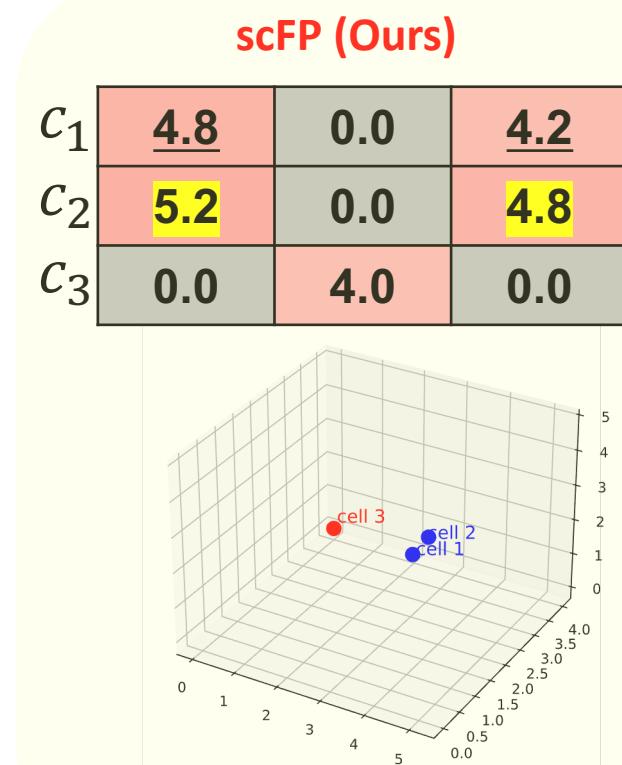
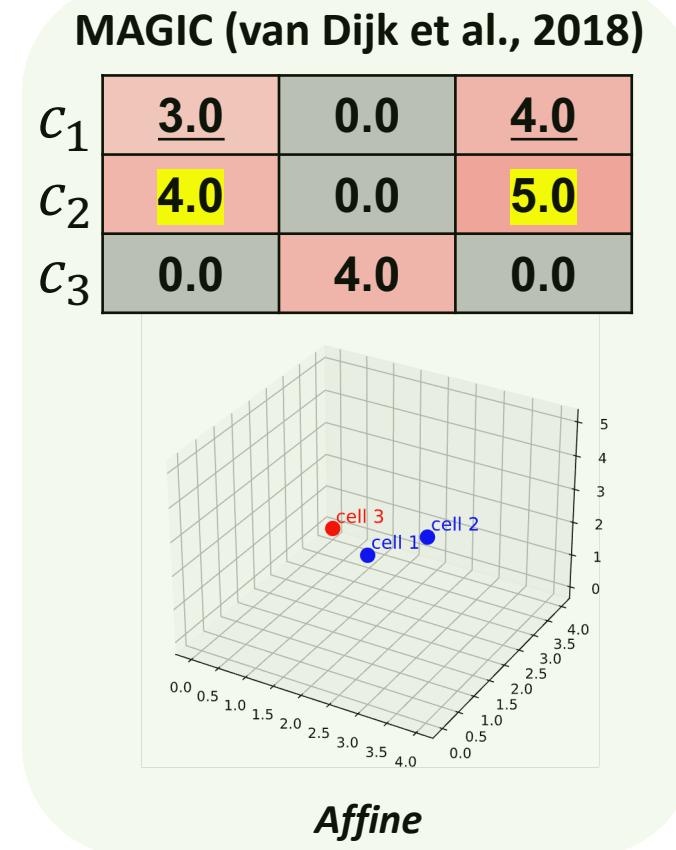
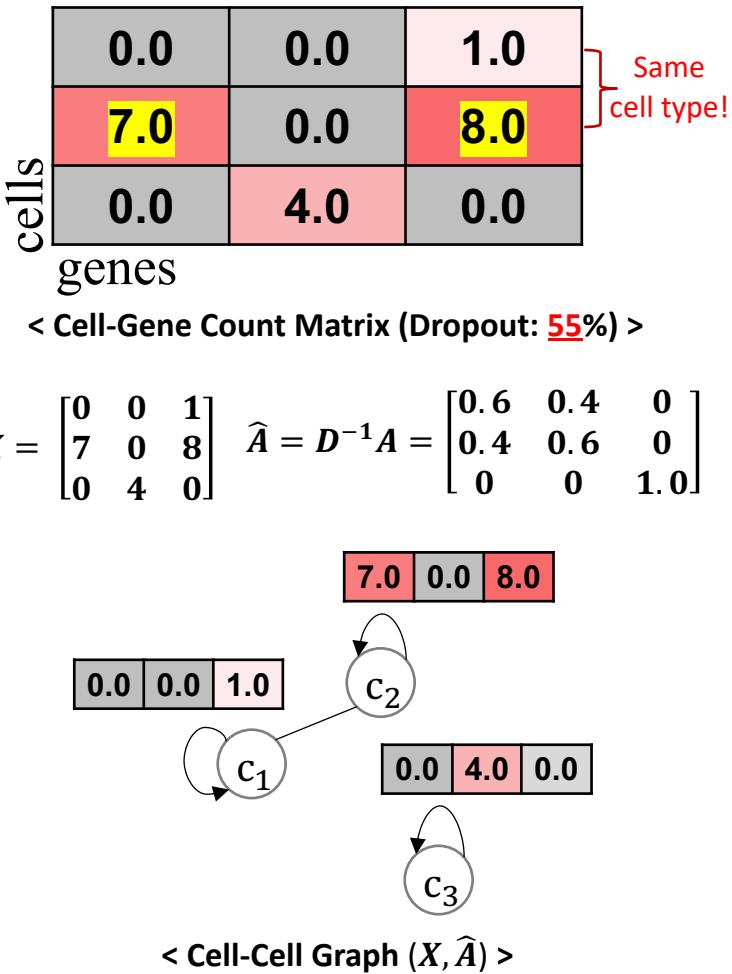
$$X = \begin{bmatrix} 7 & 1 & 1 \\ 7 & 0 & 8 \\ 0 & 4 & 0 \end{bmatrix} \quad \widehat{A} = D^{-1}A = \begin{bmatrix} 0.5 & 0.4 & 0.1 \\ 0.4 & 0.6 & 0 \\ 0.1 & 0 & 0.9 \end{bmatrix}$$



At low dropout rates, both MAGIC and scFP perform well on cell type clustering

EXPERIMENTS

- Simulation Study: Risk of diffusion of false-zeros (MAGIC vs scFP) – **High** dropout rates



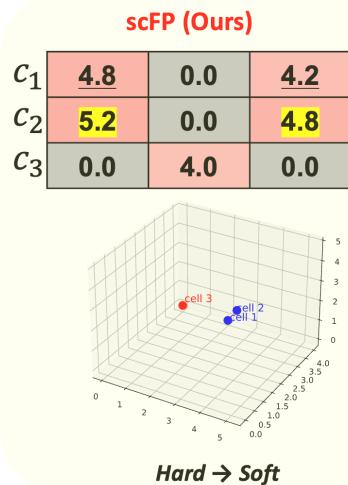
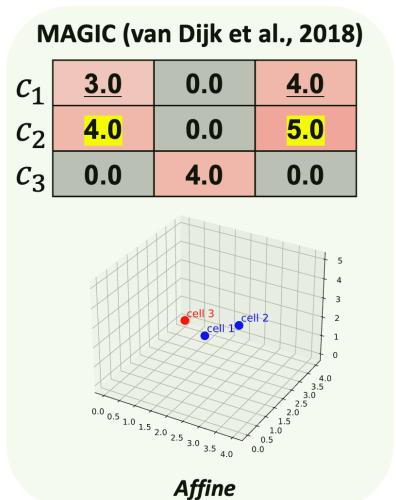
At **high** dropout rates, scFP better perseveres the scale of original non-zero values thanks to *Hard Feature Propagation*, while MAGIC is more vulnerable to false-zeros

EXPERIMENTS

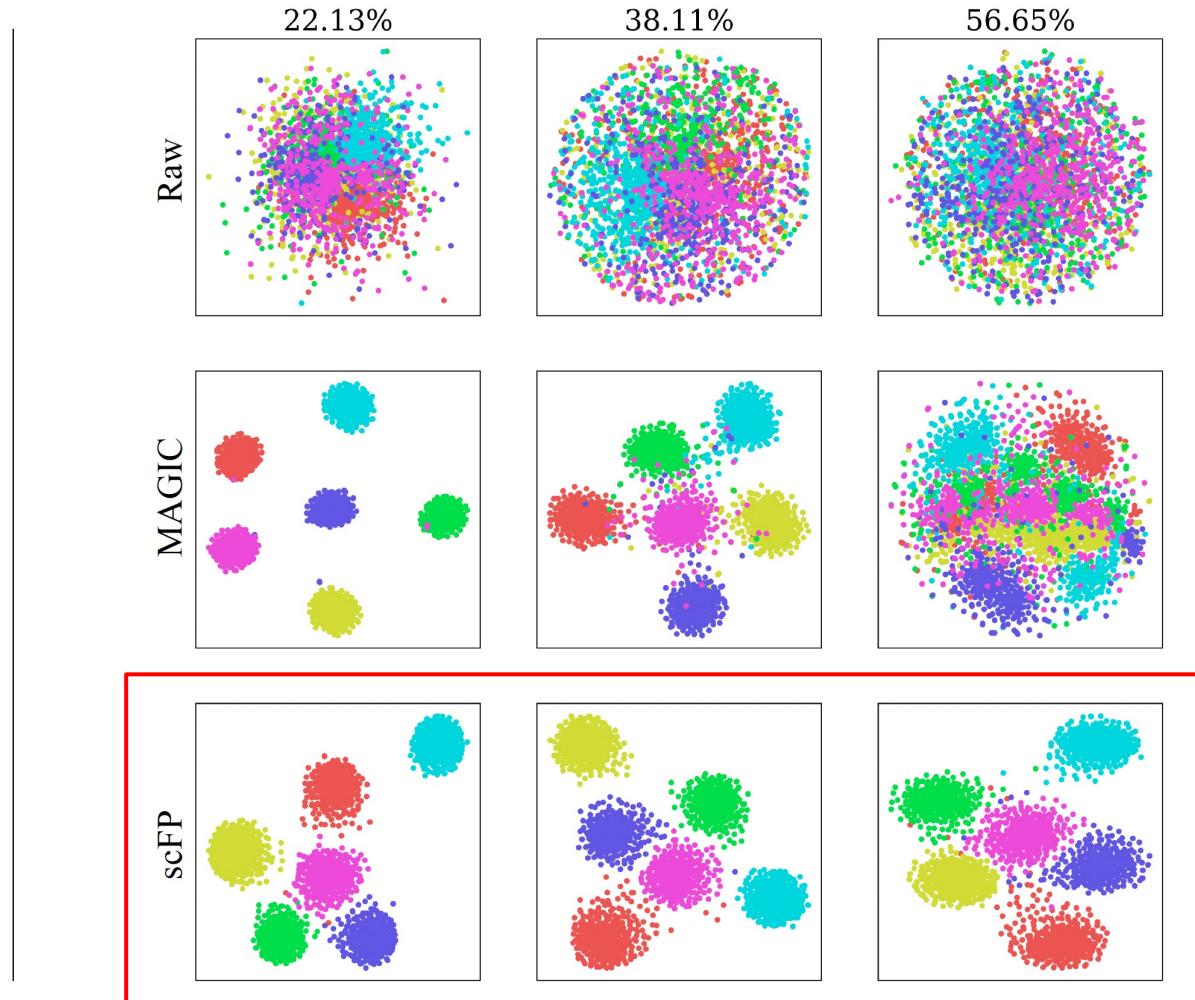
▪ Simulation Study: Risk of diffusion of false-zeros in simulation dataset

Simulation dataset from Splatter¹

- # of Cells: 3918
- # of Genes: 11786
- # of subgroups: 6
- Dropout Rate: 22.13%, 38.11%, 56.65%



At high dropout rates, scFP better perseveres the scale of original non-zero values thanks to Hard Feature Propagation, while MAGIC is more vulnerable to false-zeros



EXPERIMENTS

▪ Memory & Time Complexity

Model	Matrices & Parameters in GPU	Big-O
DCA (Eraslan et al., 2019)	$X, f_{enc}, f_{dec}, W_\pi, W_\mu, W_\theta$	$\mathcal{O}(NM) + \mathcal{O}(\text{model})$
AutoClass (Li et al., 2022)	$X, f_{enc}, f_{dec}, f_{cls}$	$\mathcal{O}(NM) + \mathcal{O}(\text{model})$
scGCL (Xiong et al., 2023)	$X, A, f_{o_enc}, f_{t_enc}, f_{dec}, W_\pi, W_\mu, W_\theta, q_\theta$	$\mathcal{O}(NM) + \mathcal{O}(N^2) + \mathcal{O}(\text{model})$
kNN-Smoothing (Wagner et al., 2018)	X, A	$\mathcal{O}(NM) + \mathcal{O}(N^2)$
MAGIC (van Dijk et al., 2018)	X, A	$\mathcal{O}(NM) + \mathcal{O}(N^2)$
scFP (Ours)	X, A	$\mathcal{O}(NM) + \mathcal{O}(N^2)$

< Memory Complexity (N : # of Cells, M : # of Genes) >

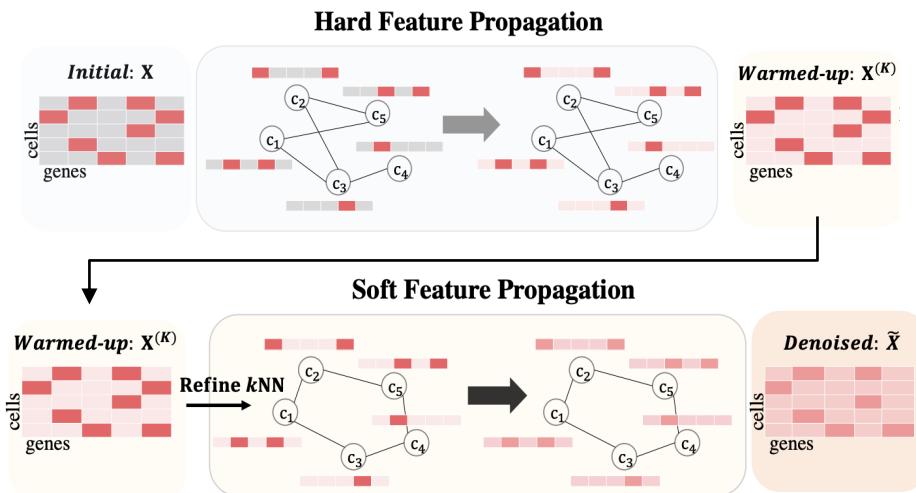


< Time Complexity >

Proposed scFP shows notably lower computational cost compared to other graph-based baselines thanks to the absence of PCA computation and the use of sparse matrix multiplication

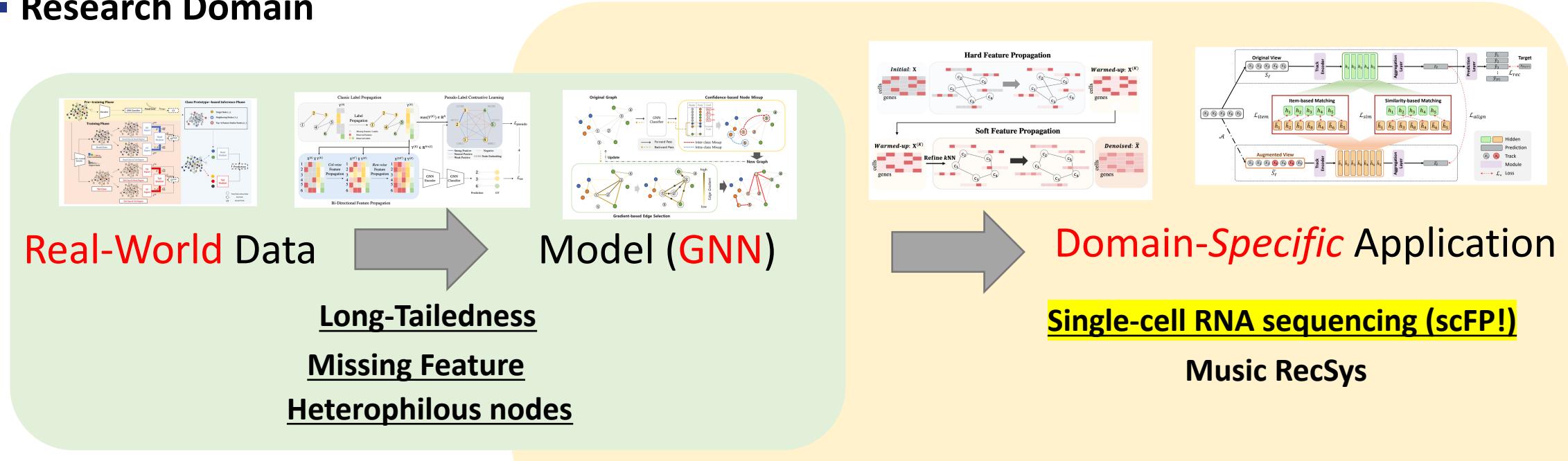
CONCLUSION

- Nut: scRNA-seq meets Feature Propagation!
- Challenges lie in the presence of missing and noise in the cell-gene matrix and the lack of a biologically relevant graph
- To this end, we introduce scFP, a method that effectively denoises both zero values and non-zero values
- Experimental results on real-world datasets show the promising performance of scFP in imputation and cell clustering
- **Keywords for scFP**
 - Hard Feature Propagation
 - Refine kNN
 - Soft Feature Propagation



Seeking Collaborative Opportunities and Ph.D. Positions for Fall 2024

■ Research Domain



Sukwon Yun
(M.S. in August 2023)

Research Keywords

1. GNN modeling with real-world data (Tackling inductive bias)
2. Domain-specific application (Biomedical domain)