2023 ICML Workshop on Computational Biology (Contributed Talk)

Single-cell RNA-seq data imputation using Feature Propagation

Sukwon Yun*, Junseok Lee*, Chanyoung Park

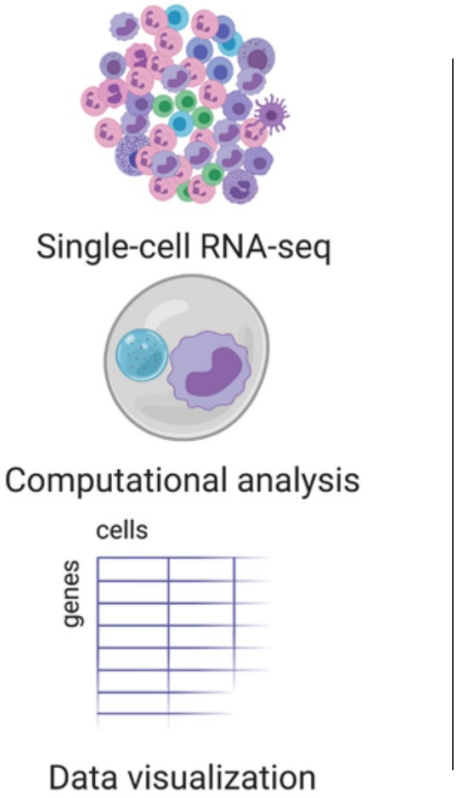
Korea Advanced Institute of Science and Technology (KAIST)

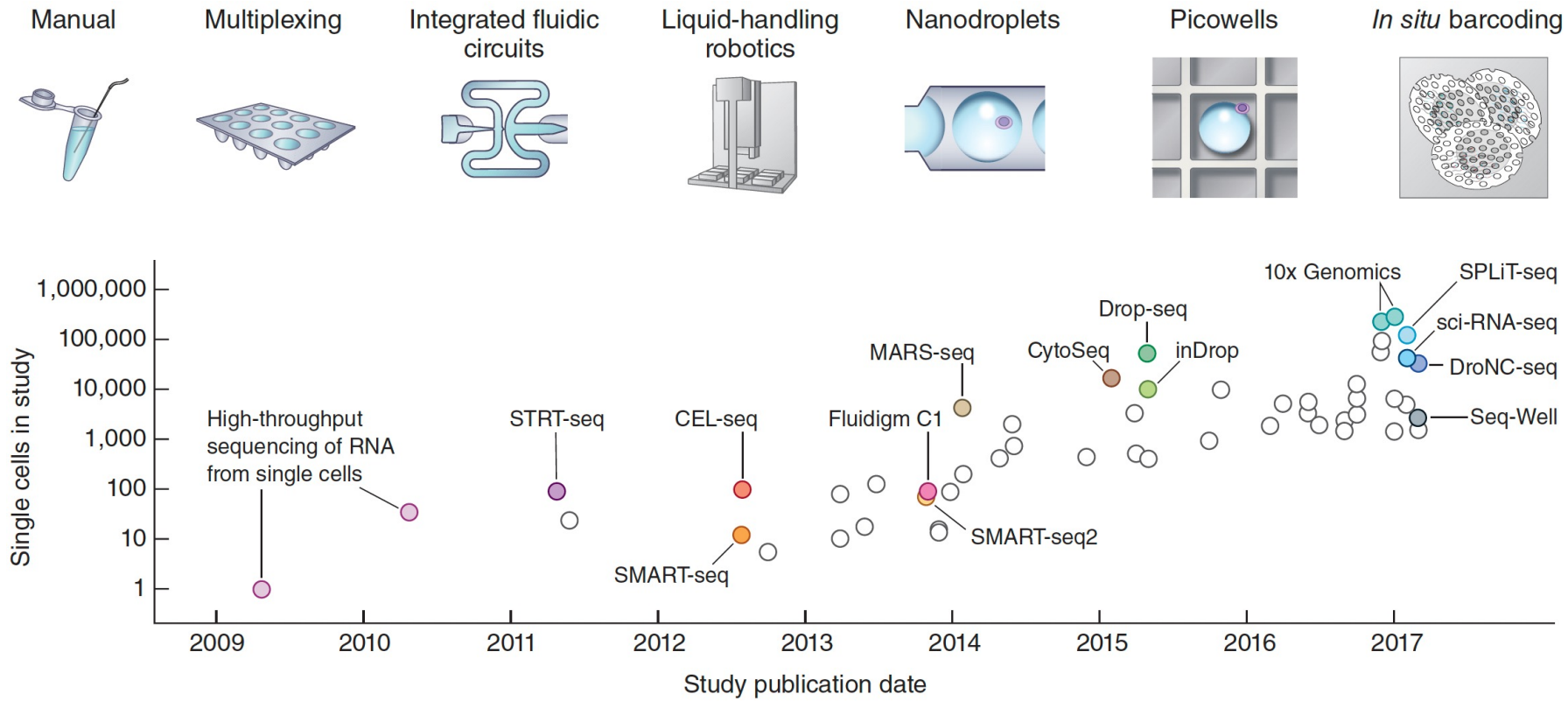
KAIST DSAIL Data Science & Artificial Intelligence Lab.

TABLE OF CONTENTS

BACKGROUND: SINGLE-CELL RNA-SEQ

- **Advances in single-cell RNA-seq**

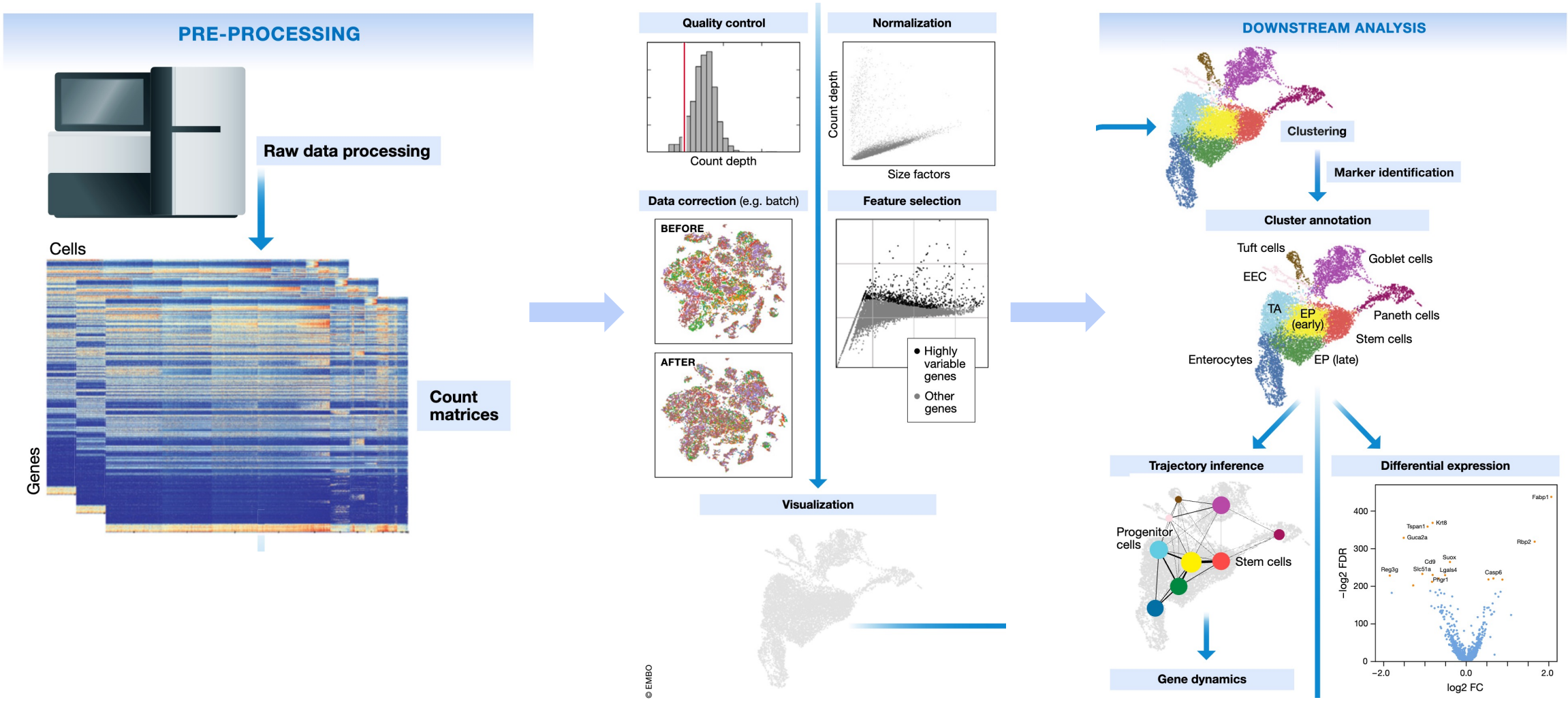




Svensson, Valentine, Roser Vento-Tormo, and Sarah A. Teichmann. "Exponential scaling of single-cell RNA-seq in the past decade." *Nature protocols* 13.4 (2018): 599-604.
Acosta, Jean, Daniel Ssozi, and Peter van Galen. "Single-cell RNA sequencing to disentangle the blood system." *Arteriosclerosis, thrombosis, and vascular biology* 41.3 (2021): 1012-1018.

BACKGROUND: SINGLE-CELL RNA-SEQ

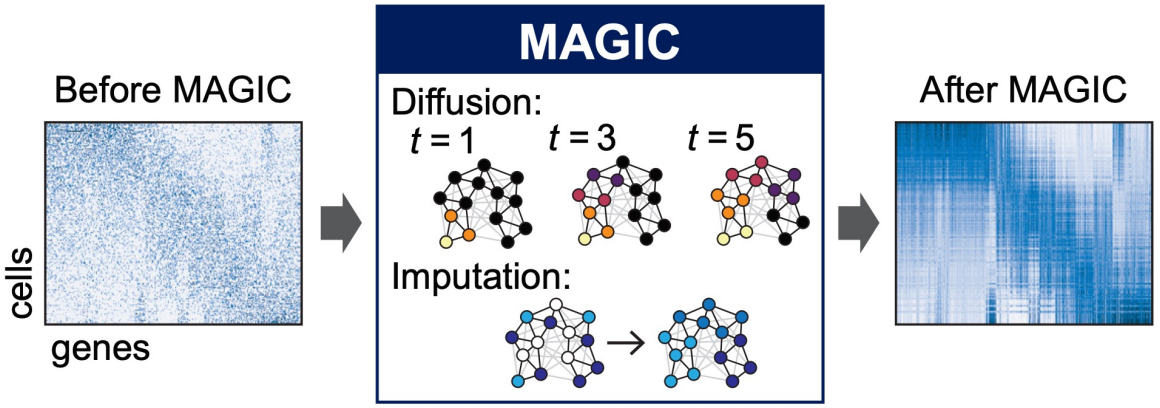
- **Workflow of single-cell RNA-seq (scRNA-seq) analysis**



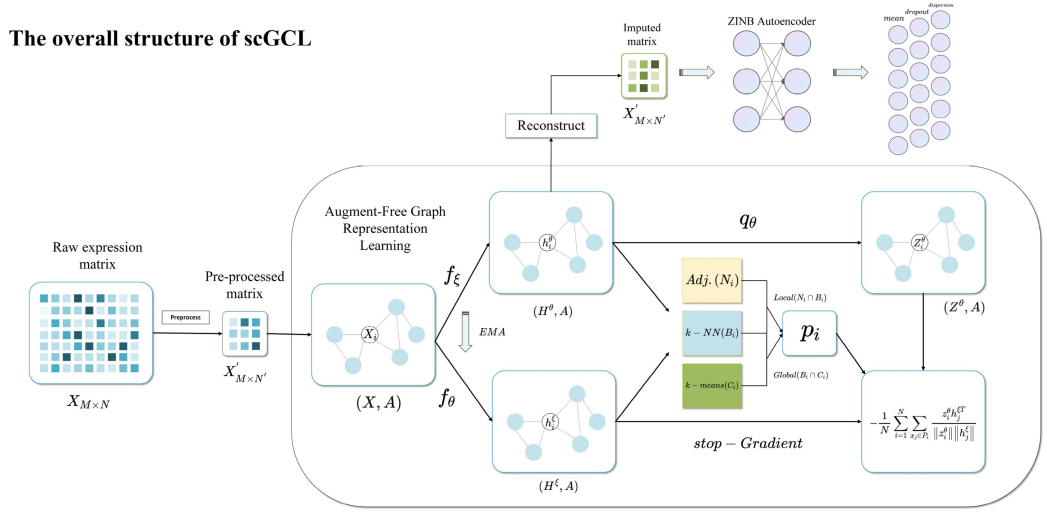
Luecken, Malte D., and Fabian J. Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial." *Molecular systems biology* 15.6 (2019): e8746.

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)

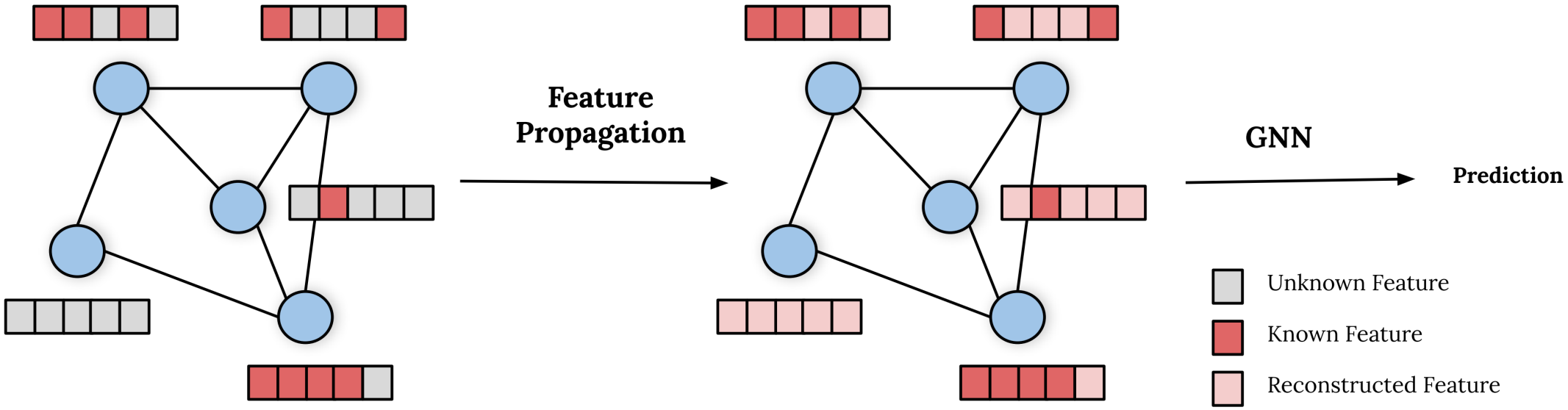


scGCL (Xiong et al., 2023)

Van Dijk, David, et al. "Recovering gene interactions from single-cell data using data diffusion." *Cell* 174.3 (2018): 716-729.
Xiong, Zehao, et al. "scGCL: an imputation method for scRNA-seq data based on graph contrastive learning." *Bioinformatics* 39.3 (2023): btad098.

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



Rossi, Emanuele, et al. "On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features." *Learning on Graphs Conference*. PMLR, 2022.

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

Analytic Approach

Dirichlet Energy

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

Gradient flow

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

Heat Diffusion Equation

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

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

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

Rossi, Emanuele, et al. "On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features." *Learning on Graphs Conference*. PMLR, 2022.

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

Analytic Approach

Dirichlet Energy

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

Gradient flow

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

Heat Diffusion Equaion

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

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

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

Iterative Approach

Euler Scheme

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

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

when $h = 1$,

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

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

Rossi, Emanuele, et al. "On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features." *Learning on Graphs Conference*. PMLR, 2022.

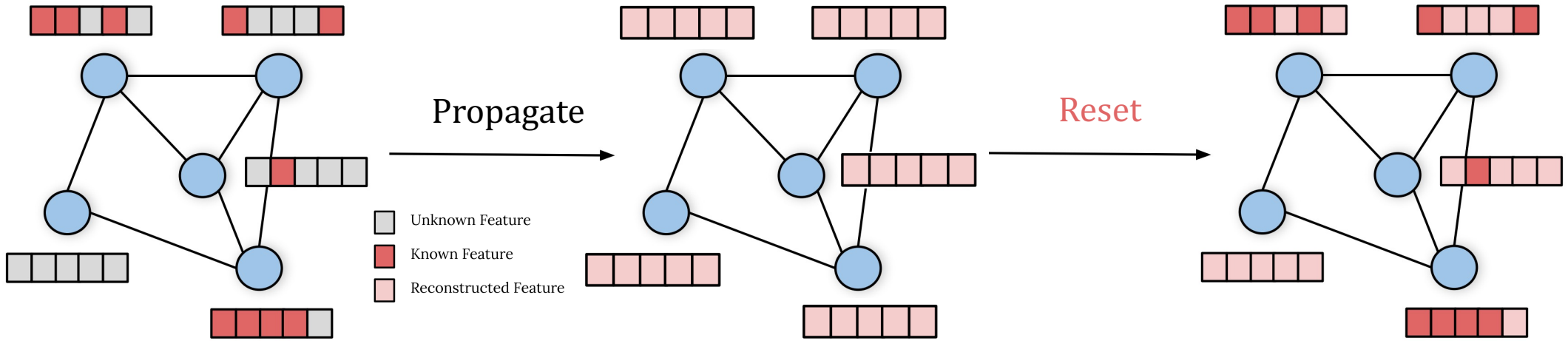
BACKGROUND: AS A GRAPH

- **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

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

1) Propagate: $\mathbf{x}^{(\mathrm{k}+1)} = \tilde{\mathbf{A}}\mathbf{x}^{(\mathrm{k})} = \begin{bmatrix} \tilde{\mathbf{A}}_{kk} & \tilde{\mathbf{A}}_{ku} \\ \tilde{\mathbf{A}}_{uk} & \tilde{\mathbf{A}}_{uu} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_k^{(\mathrm{k})} \\ \mathbf{x}_u^{(\mathrm{k})} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}}_{kk}\mathbf{x}_k^{(\mathrm{k})} + \tilde{\mathbf{A}}_{ku}\mathbf{x}_u^{(\mathrm{k})} \\ \tilde{\mathbf{A}}_{uk}\mathbf{x}_k^{(\mathrm{k})} + \tilde{\mathbf{A}}_{uu}\mathbf{x}_u^{(\mathrm{k})} \end{bmatrix}$

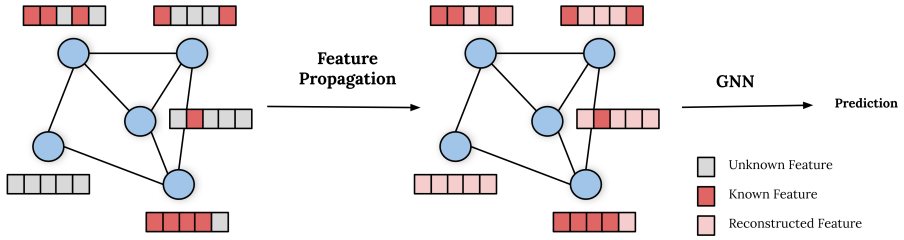
2) Reset: $\mathbf{x}_k^{(\mathrm{k}+1)} = \mathbf{x}_k^{(0)} \rightarrow \mathbf{x}^{(\mathrm{k}+1)} = \begin{bmatrix} \mathbf{x}_k^{(0)} \\ \tilde{\mathbf{A}}_{uk}\mathbf{x}_k^{(\mathrm{k})} + \tilde{\mathbf{A}}_{uu}\mathbf{x}_u^{(\mathrm{k})} \end{bmatrix}$



Rossi, Emanuele, et al. "On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features." *Learning on Graphs Conference*. PMLR, 2022.

MOTIVATION: scRNA-seq DATA IMPUTATION USING FEATURE PROPAGATION

- **Research Direction: Feature Propagation on scRNA-seq data**



Iterative Approach

Euler Scheme

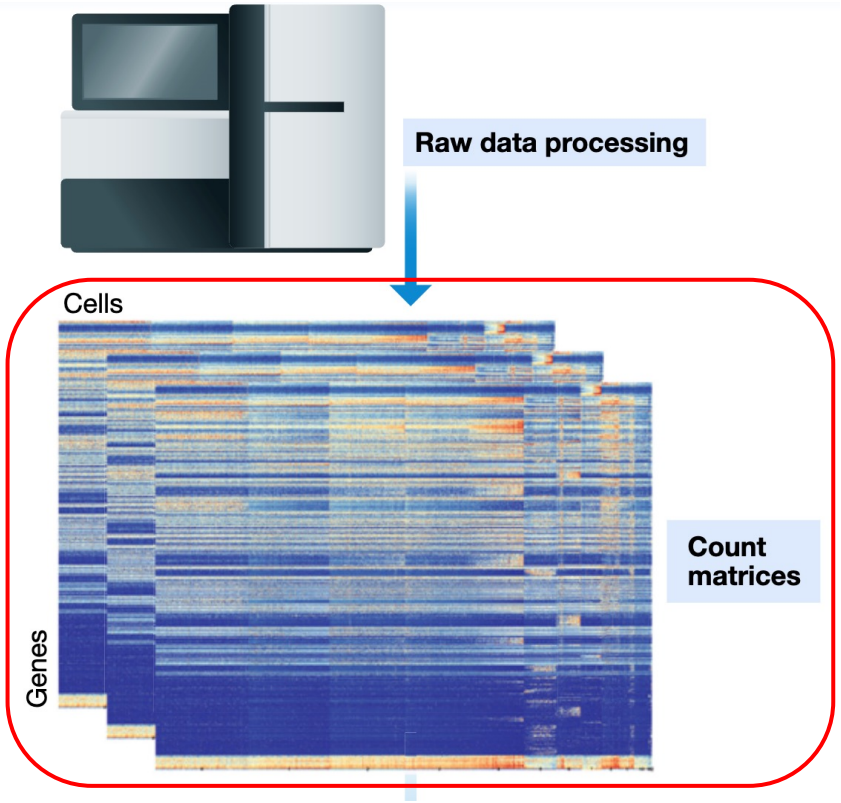
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - h\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \boldsymbol{\Delta}_{uk} & \boldsymbol{\Delta}_{uu}\end{bmatrix}\mathbf{x}^{(k)}$$

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

when $h = 1$,

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

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

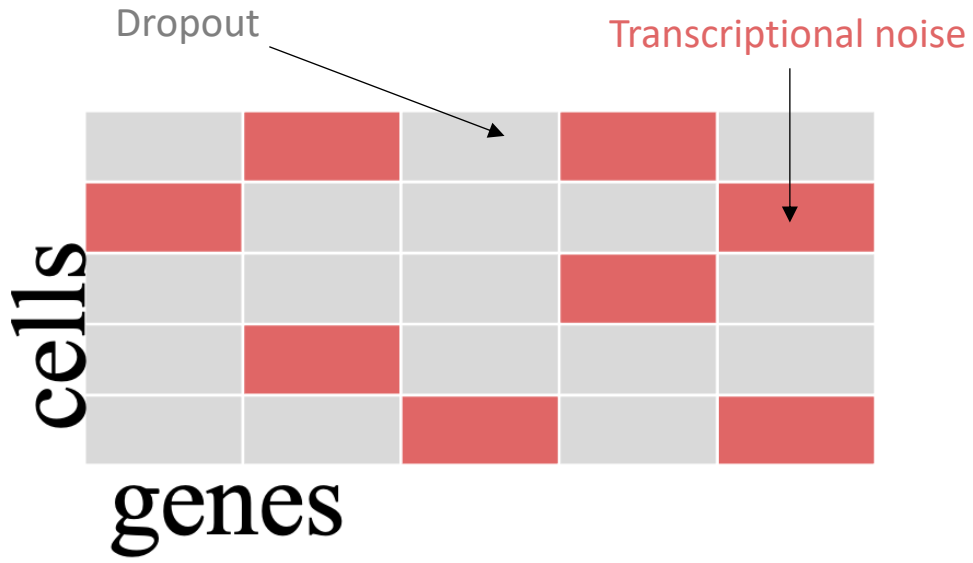


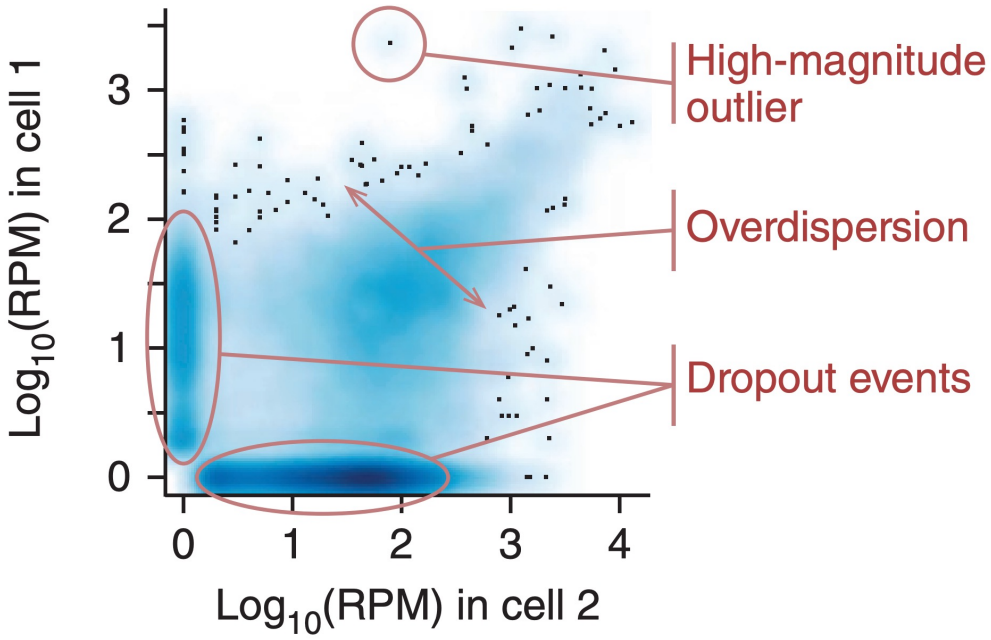
**** Challenges***

1. The information regarding <u>which features are missing or noisy</u> is not provided
2. <u>Biologically relevant graph structure</u> 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)**



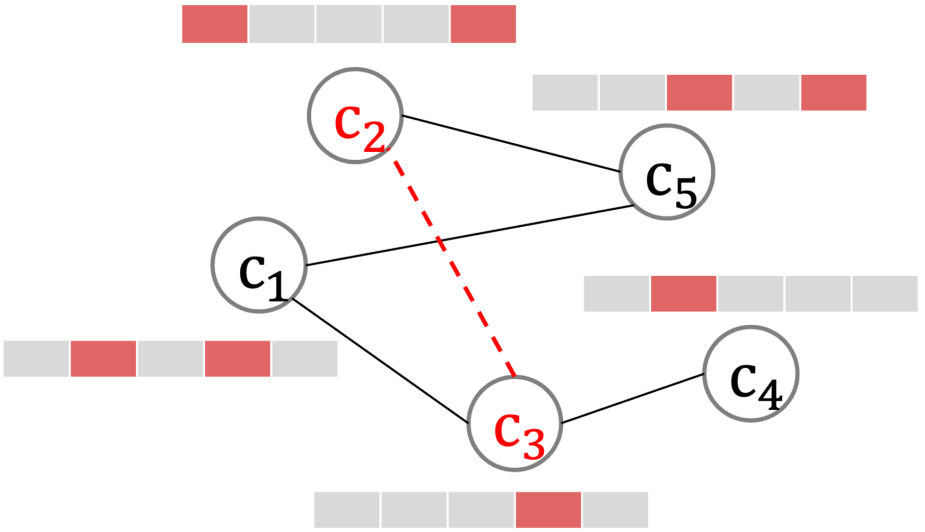


Careful handling of both zero-values and non-zero values is crucial

Kharchenko, Peter V., Lev Silberstein, and David T. Scadden. "Bayesian approach to single-cell differential expression analysis." *Nature methods* 11.7 (2014): 740-742.

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**

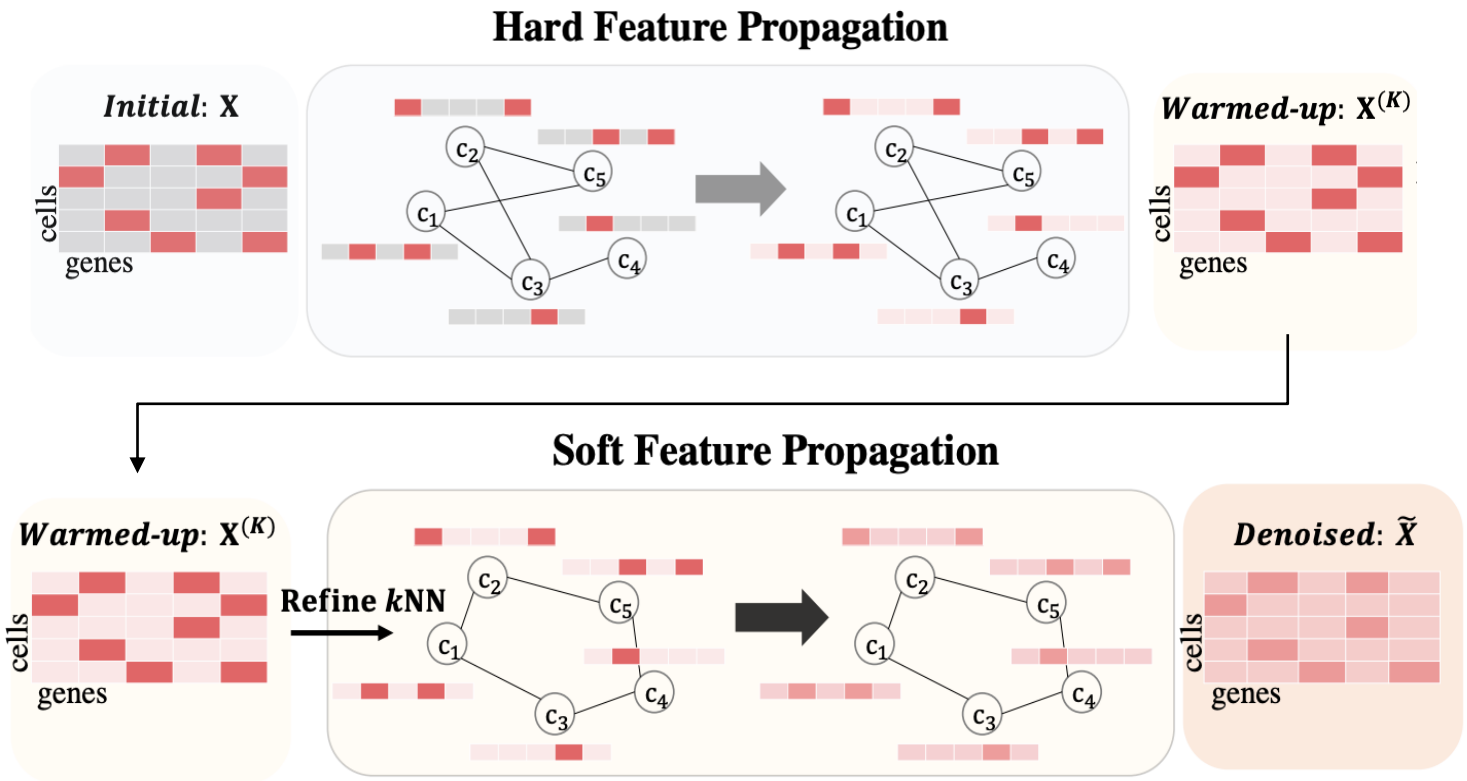


< *k*NN Graph on sparse cell-gene matrix >

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

scFP: single-cell FEATURE PROPAGATION

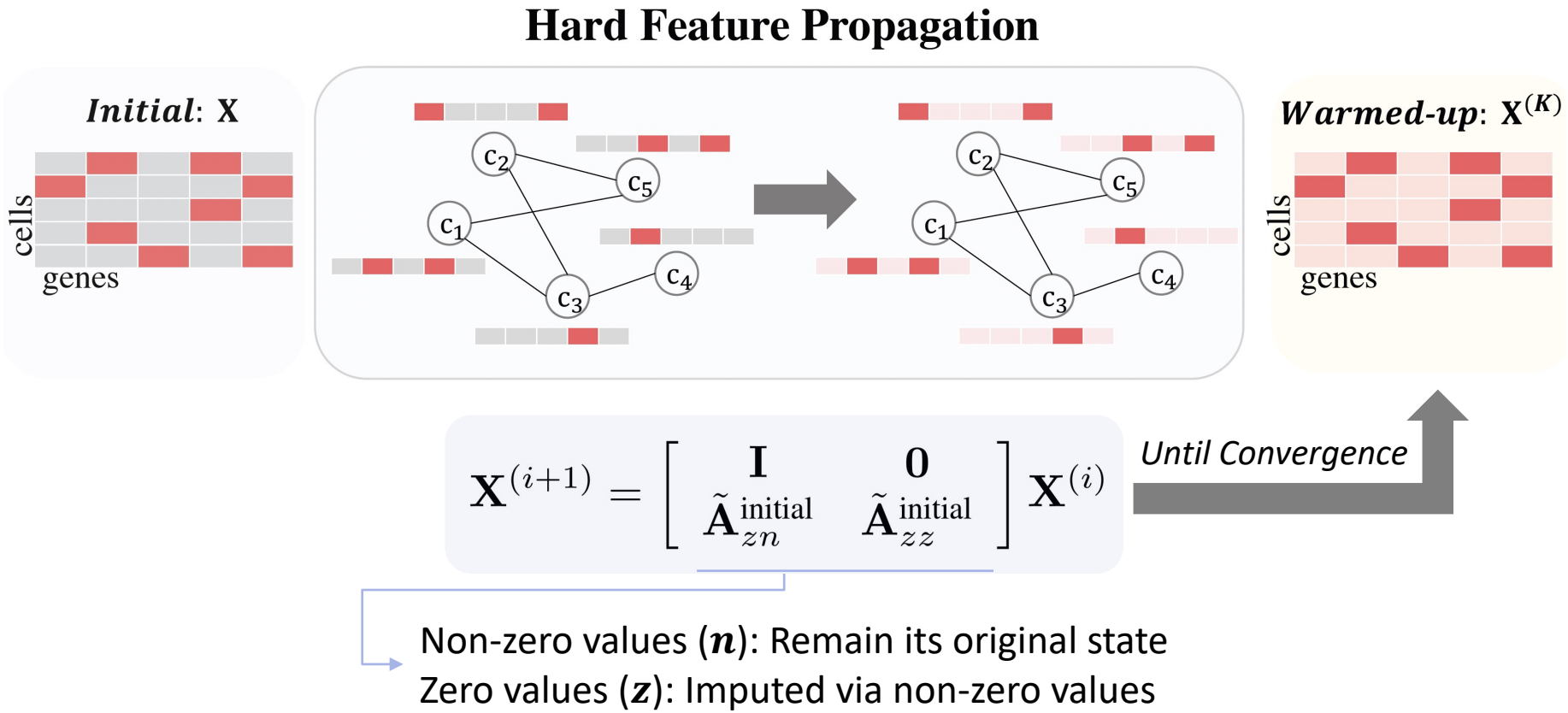
- **Overall Framework of scFP**



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

scFP: single-cell FEATURE PROPAGATION

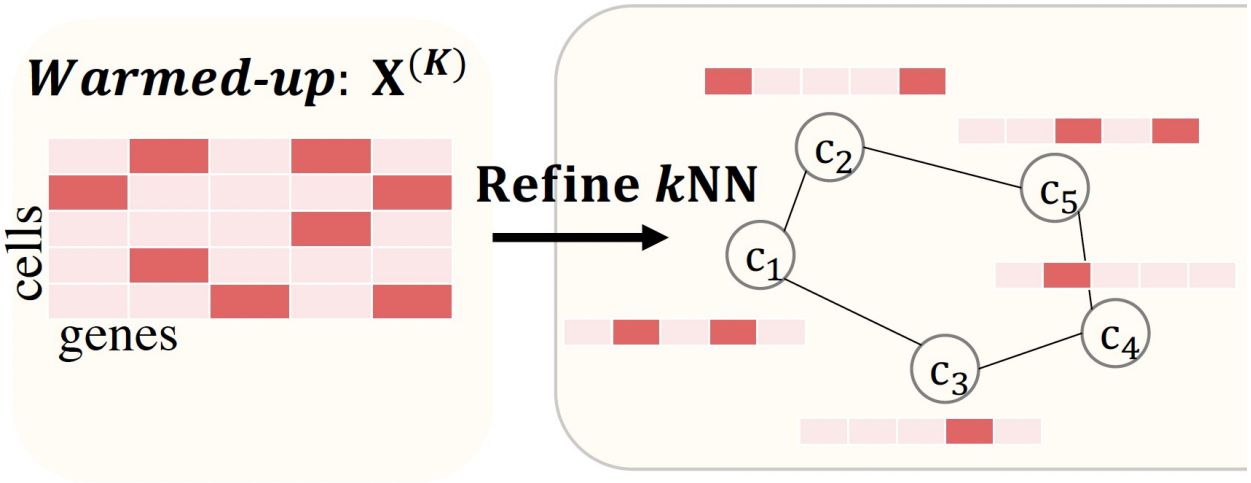
- **1) Hard Feature Propagation**
  - Impute zero values (dropout) via observed gene expression and obtain warmed-up cell-gene matrix
  - *Assumption: Imputing zeros (dropout) is more significant than denoising non-zeros at the initial stage*



$$\mathbf{X}^{(i+1)} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \tilde{\mathbf{A}}^{\text{initial}}_{zn} & \tilde{\mathbf{A}}^{\text{initial}}_{zz} \end{bmatrix} \mathbf{X}^{(i)}$$

scFP: single-cell FEATURE PROPAGATION

- **2) Refine kNN Graph**
  - Build kNN Graph via warmed-up cell-gene matrix
  - Compared to initial kNN Graph, it would potentially reveal hidden or implicit graph structures

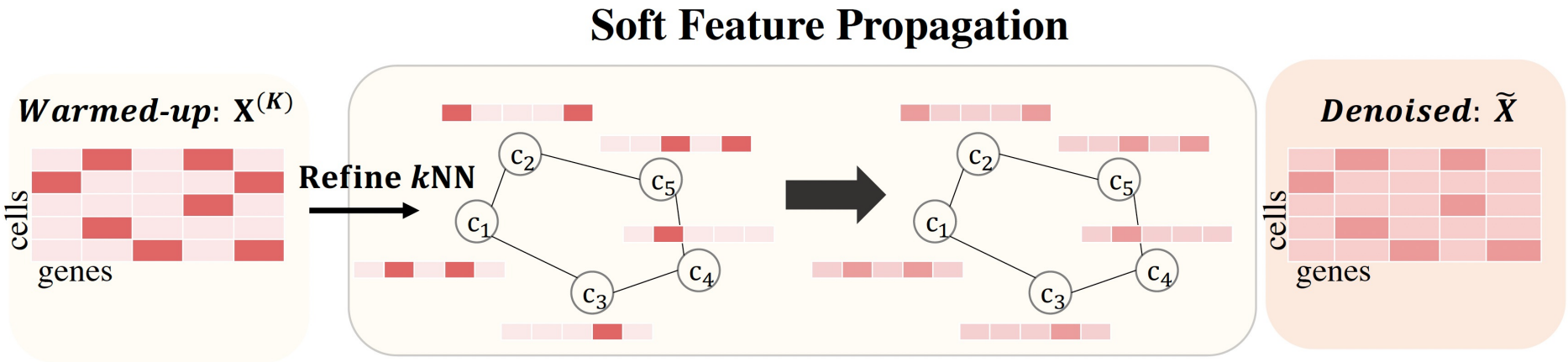


$$\tilde{\mathbf{A}}^{\text{refined}} = k\text{NN}(\mathbf{X}^{(K)})$$

Warmed-up matrix obtained from Hard FP

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



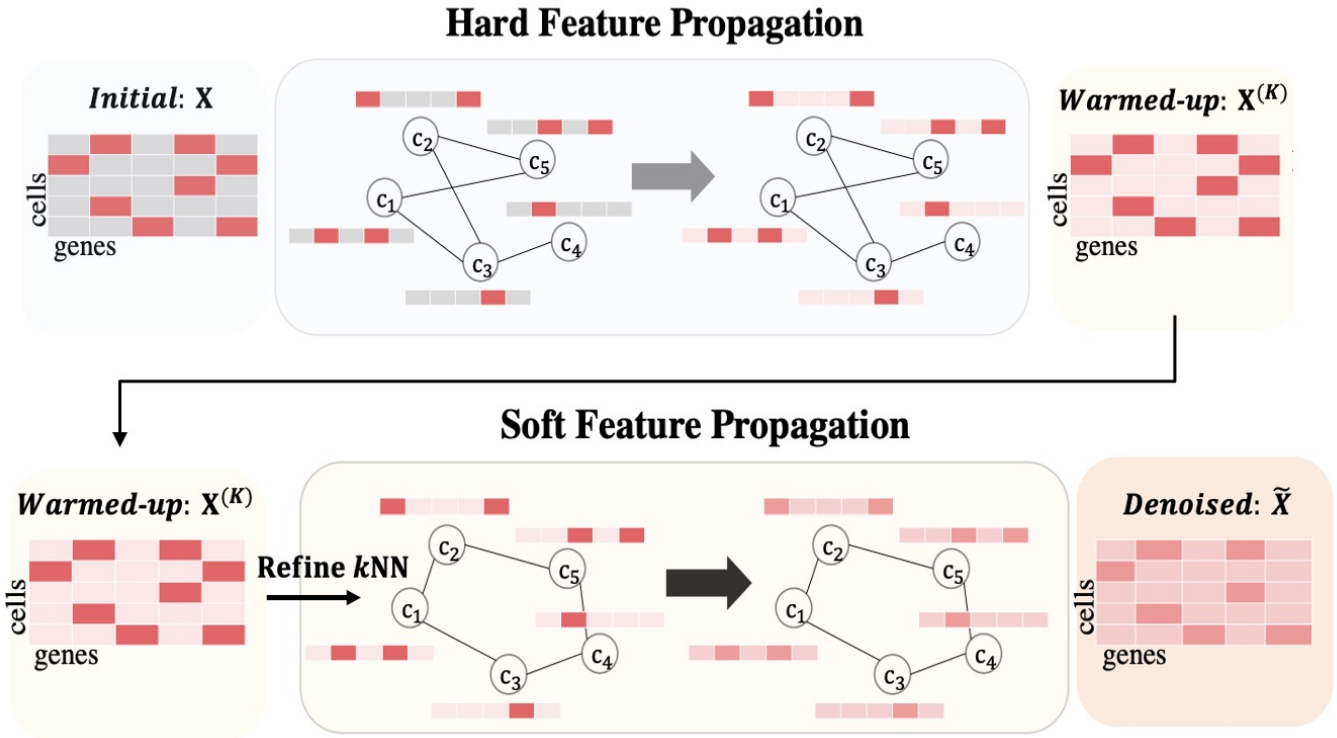
$$\mathbf{X}^{(K+j+1)} = \alpha \tilde{\mathbf{A}}^{\text{refined}} \mathbf{X}^{(K+j)} + (1-\alpha)\mathbf{X}^{(K)}$$

Amount of information from neighbors

Amount of information from initial state

scFP: single-cell FEATURE PROPAGATION

- **In a nutshell,**



Algorithm 1 single-cell Feature Propagation (scFP)

Input: Cell-Gene Matrix $\mathbf{X}$, Initial kNN $\tilde{\mathbf{A}}^{\text{initial}}$
Output: Denoised Cell-Gene Matrix $\tilde{\mathbf{X}}$

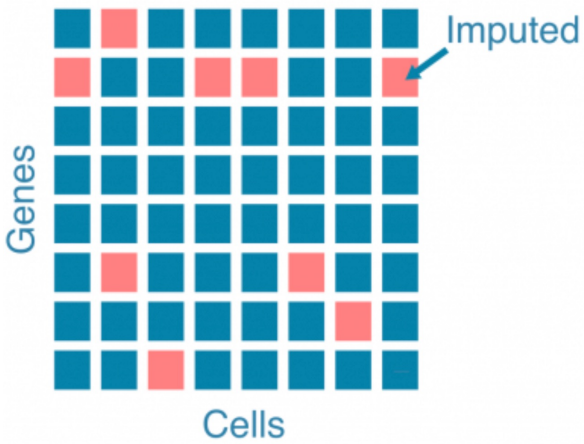
$\mathbf{Y} \leftarrow \mathbf{X}$
while $\mathbf{X}$ has not converged **do**
  $\mathbf{X} \leftarrow \tilde{\mathbf{A}}^{\text{initial}}\mathbf{X}$
  $\mathbf{X}_{k,d} \leftarrow \mathbf{Y}_{k,d} \forall k \in \mathcal{V}_{k,d}, \forall d \leq M$ — **Hard Clamping**
end while
$\tilde{\mathbf{A}}^{\text{refined}} = k\text{NN}(\mathbf{X}^{(K)})$ — **Refine *k*NN**
while $\mathbf{X}^{(K)}$ has not converged **do**
  $\mathbf{X}^{(K)} \leftarrow \alpha\tilde{\mathbf{A}}^{\text{refined}}\mathbf{X}^{(K)} + (1-\alpha)\mathbf{X}^{(K)}$ — **Soft Clamping**
end while

Impute zeros (Dropouts) ⟶ Refine *k*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



Imputation

Cell Clustering

1) Imputation Task

- **Root Mean Square Error (RMSE)**

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

N: # of cells
y_i: ground-truth gene expression i-th cell
$\hat{y}_i$: predicted gene expression i-th cell

2) Cell Clustering Task

- **Adjusted Rand Index (ARI)**

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \quad RI = \frac{a+b}{{}_N C_2}$$

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**

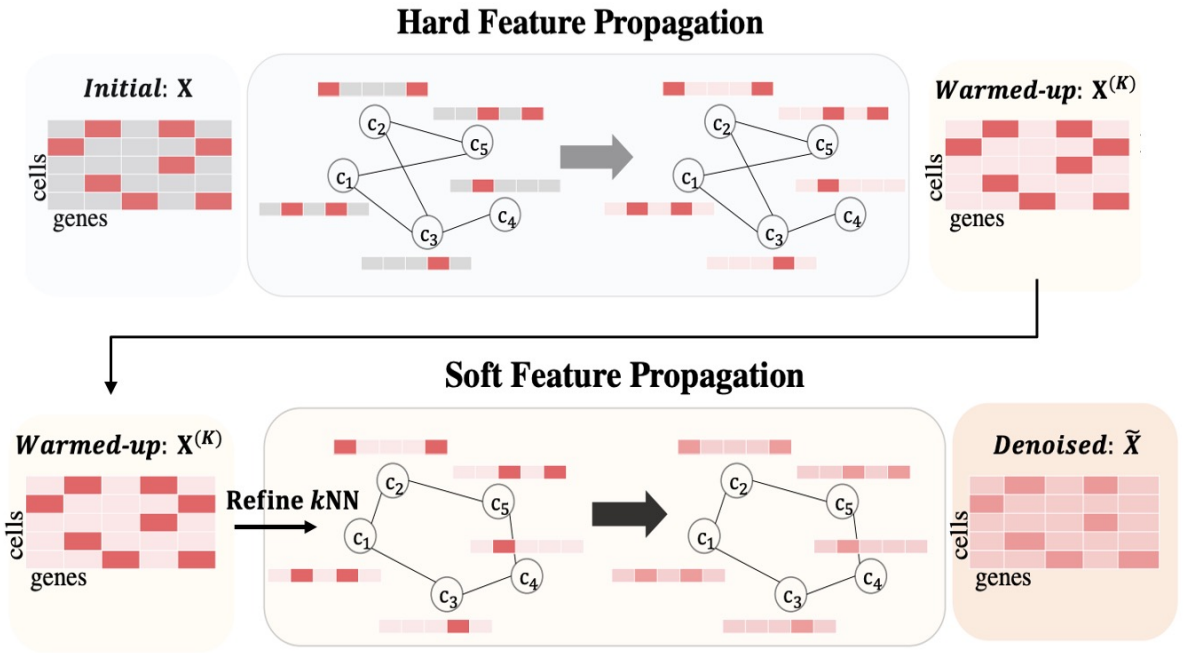
Imputation	Baron Mouse			Mouse ES			Mouse Bladder			Zeisel			Baron Human		
	Dropout Rates			Dropout Rates			Dropout Rates			Dropout Rates			Dropout Rates		
	20%	40%	80%	20%	40%	80%	20%	40%	80%	20%	40%	80%	20%	40%	80%
MAGIC	0.61	0.73	0.99	0.53	0.73	1.21	0.50	0.60	0.82	0.60	0.82	1.31	0.58	0.74	1.06
DCA	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**

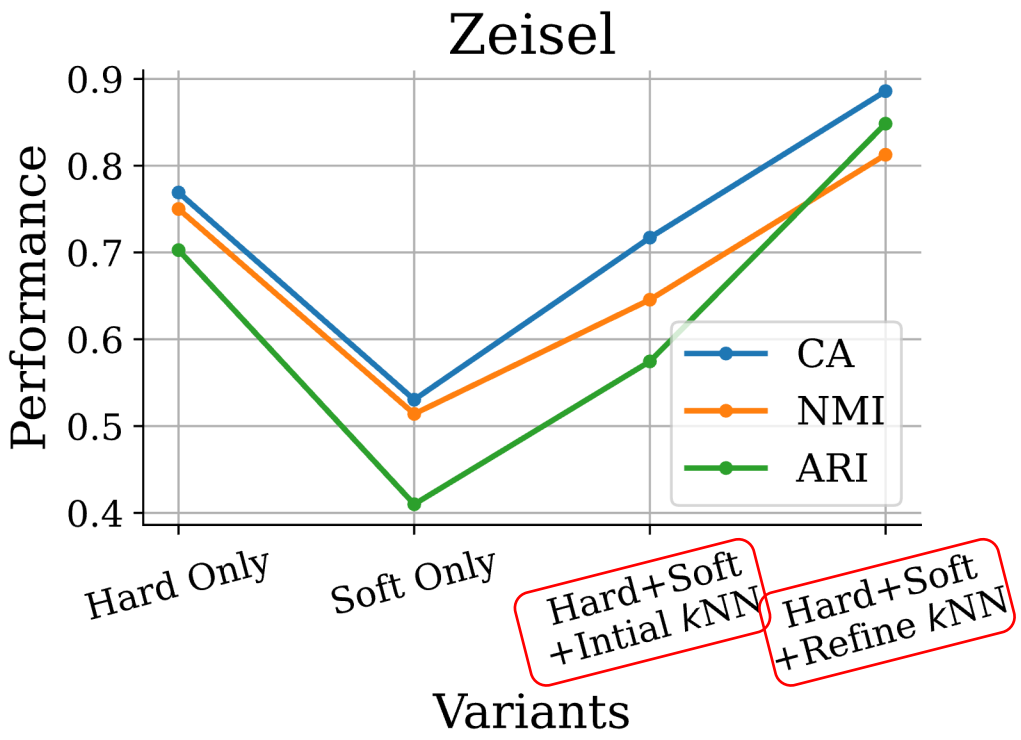
Cell Clustering	Baron Mouse			Mouse ES			Mouse Bladder			Zeisel			Baron Human		
	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA	ARI	NMI	CA
Raw	0.44	0.71	0.56	0.74	0.75	0.79	0.59	0.75	0.68	0.70	0.75	0.77	0.44	0.71	0.56
MAGIC	0.42	0.72	0.57	0.80	**0.85**	0.83	0.55	0.75	0.64	0.70	0.75	0.76	0.56	0.78	0.59
DCA	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**





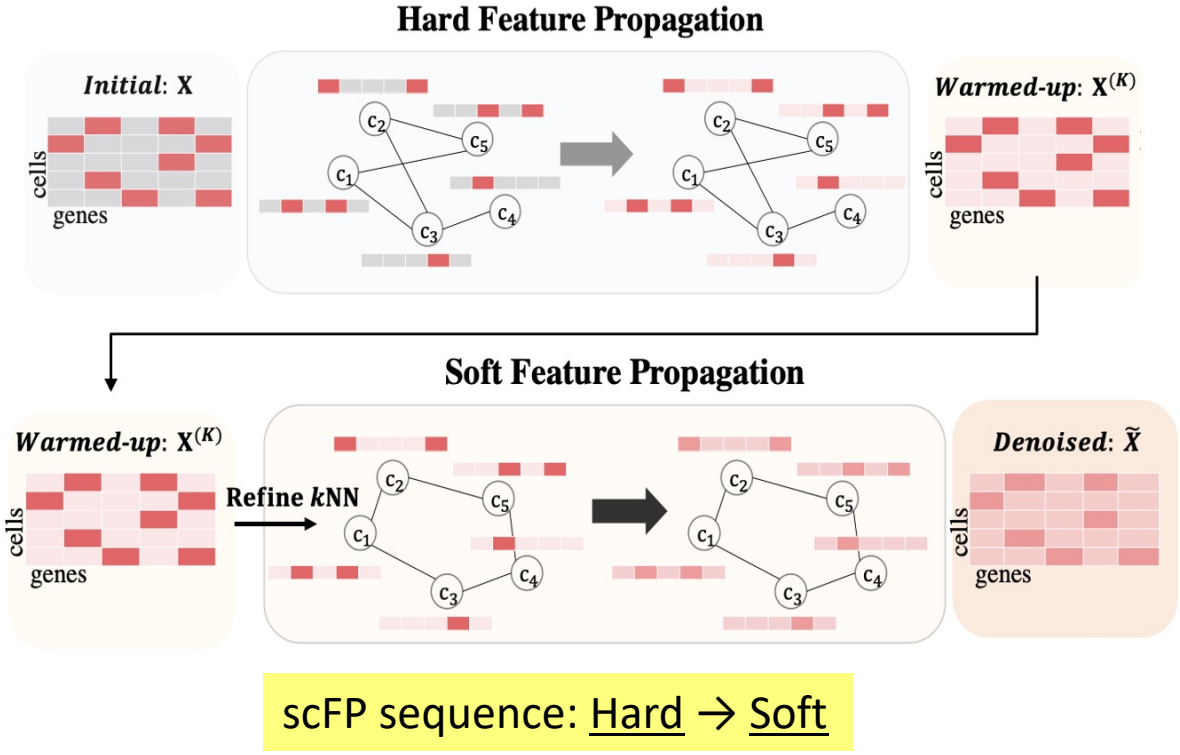
scFP modules: Hard + Soft + Refine kNN

1) (*H+S+Refine kNN* outperforms *H, S only*): **Using both Hard and Soft FP is beneficial**

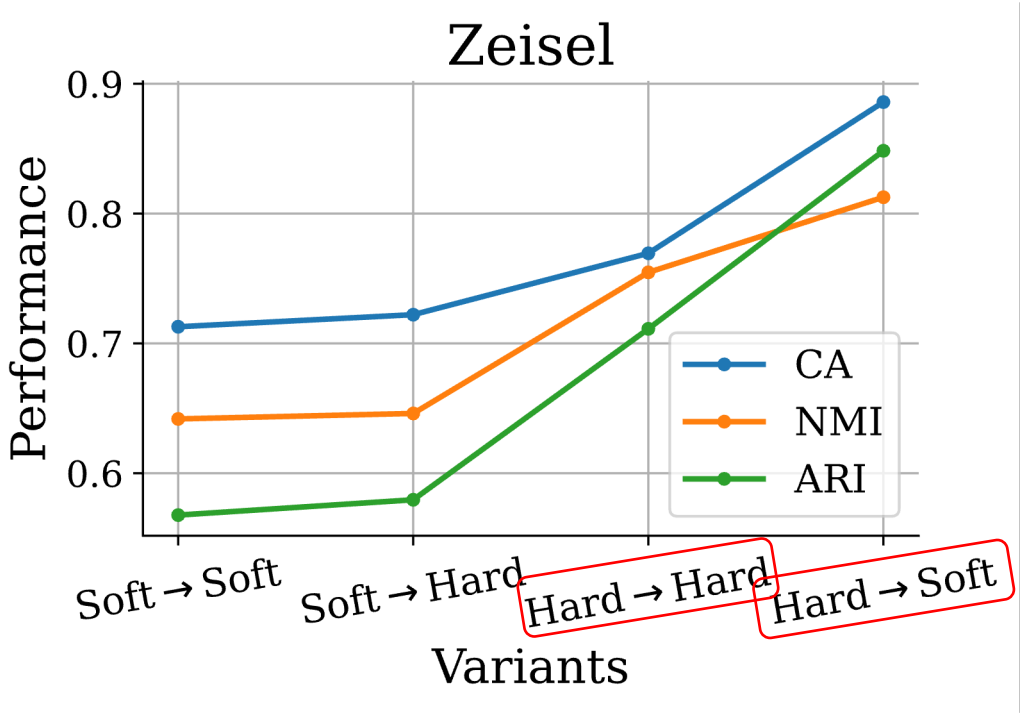
2) (*H+S+Refine kNN* outperforms *H+S+Initial kNN)*: **Utilization of a refined kNN graph prior to applying Soft FP is essential**

EXPERIMENTS

- **Ablation on sequence of scFP**



scFP sequence: Hard → Soft



1) (*H* → • outperforms *S* → •): **Initially, importance of imputing zeros surpasses the significance of denoising non-zeros**
2) (*H* → *S* outperforms *H* → *H*): **Inclusion of Soft FP after Hard FP further enhances obtaining a denoised matrix**

EXPERIMENTS

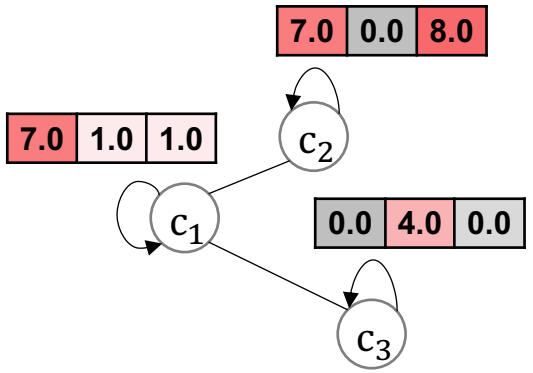
- **Simulation Study: Risk of diffusion of false-zeros (MAGIC vs scFP) – Low dropout rates**

	genes		
cells	7.0	1.0	1.0
	7.0	0.0	8.0
	0.0	4.0	0.0

Same cell type!

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

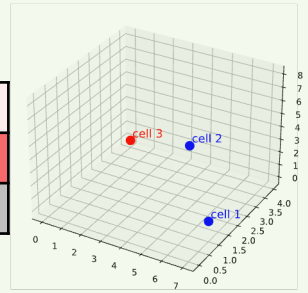
$$X = \begin{bmatrix} 7 & 1 & 1 \\ 7 & 0 & 8 \\ 0 & 4 & 0 \end{bmatrix} \quad \widehat{A} = D^{-1}A = \begin{bmatrix} 0.5 & 0.4 & 0.1 \\ 0.4 & 0.6 & 0 \\ 0.1 & 0 & 0.9 \end{bmatrix}$$



< Cell-Cell Graph $(X, \widehat{A})$ >

MAGIC (van Dijk et al., 2018)

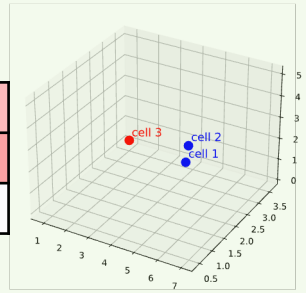
c_1	7.0	1.0	1.0
c_2	7.0	0.0	8.0
c_3	0.0	4.0	0.0



$$\begin{bmatrix} \widehat{A}_{nn} & \widehat{A}_{nz} \\ \widehat{A}_{zn} & \widehat{A}_{zz} \end{bmatrix} \cdot X$$

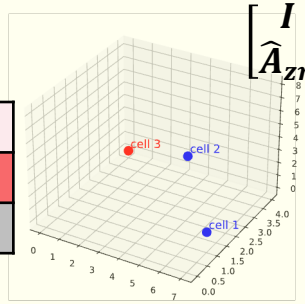
Affine

c_1	6.5	0.8	3.7
c_2	7.0	0.4	5.0
c_3	0.9	3.6	0.1



scFP (Ours)

c_1	7.0	1.0	1.0
c_2	7.0	0.0	8.0
c_3	0.0	4.0	0.0



$$\begin{bmatrix} I & 0 \\ \widehat{A}_{zn} & \widehat{A}_{zz} \end{bmatrix} \cdot X$$

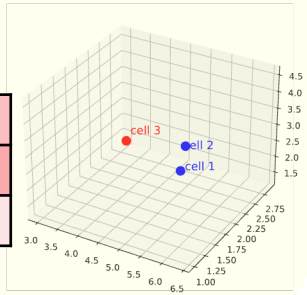
Hard

c_1	7.0	1.0	1.0
c_2	7.0	0.4	8.0
c_3	0.9	4.0	0.1

$$\alpha\widetilde{A}X + (1-\alpha)\widetilde{A}X$$

Soft

c_1	6.0	1.3	3.3
c_2	6.4	1.0	4.5
c_3	3.0	2.9	1.3



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

EXPERIMENTS

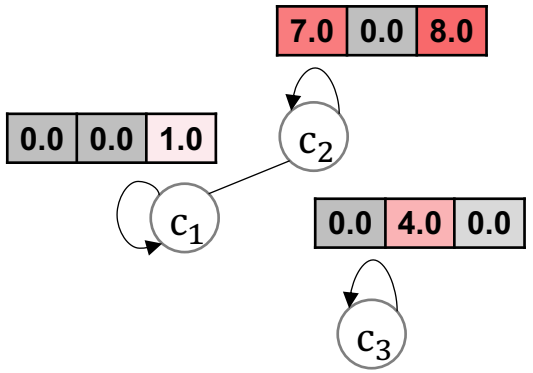
- **Simulation Study: Risk of diffusion of false-zeros (MAGIC vs scFP) – High dropout rates**

cells \ genes			
	0.0	0.0	1.0
	7.0	0.0	8.0
	0.0	4.0	0.0

Same cell type!

< Cell-Gene Count Matrix (Dropout: 55%) >

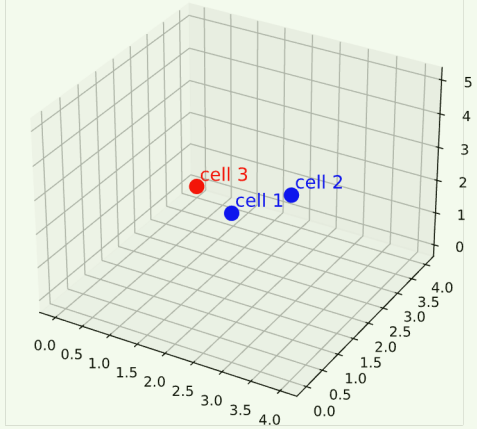
$$X = \begin{bmatrix} 0 & 0 & 1 \\ 7 & 0 & 8 \\ 0 & 4 & 0 \end{bmatrix} \quad \widehat{A} = D^{-1}A = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0.4 & 0.6 & 0 \\ 0 & 0 & 1.0 \end{bmatrix}$$



< Cell-Cell Graph $(X, \widehat{A})$ >

MAGIC (van Dijk et al., 2018)

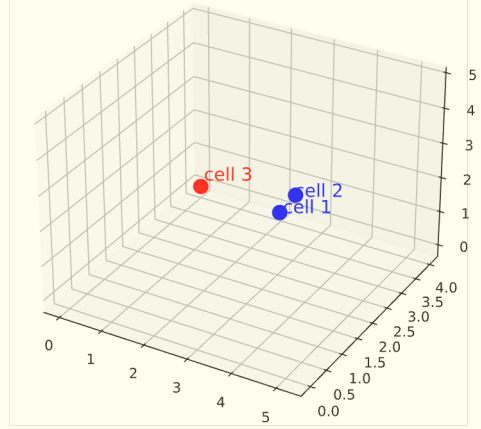
c_1	3.0	0.0	4.0
c_2	4.0	0.0	5.0
c_3	0.0	4.0	0.0



Affine

scFP (Ours)

c_1	4.8	0.0	4.2
c_2	5.2	0.0	4.8
c_3	0.0	4.0	0.0



Hard → Soft

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

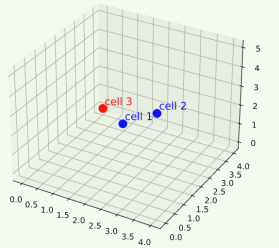
- **Simulation Study: Risk of diffusion of false-zeros in simulation dataset**

Simulation dataset from Splatter[1]

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

MAGIC (van Dijk et al., 2018)

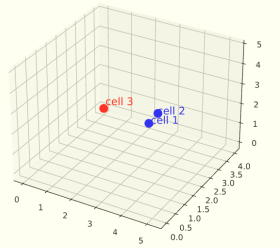
c_1	3.0	0.0	4.0
c_2	4.0	0.0	5.0
c_3	0.0	4.0	0.0



Affine

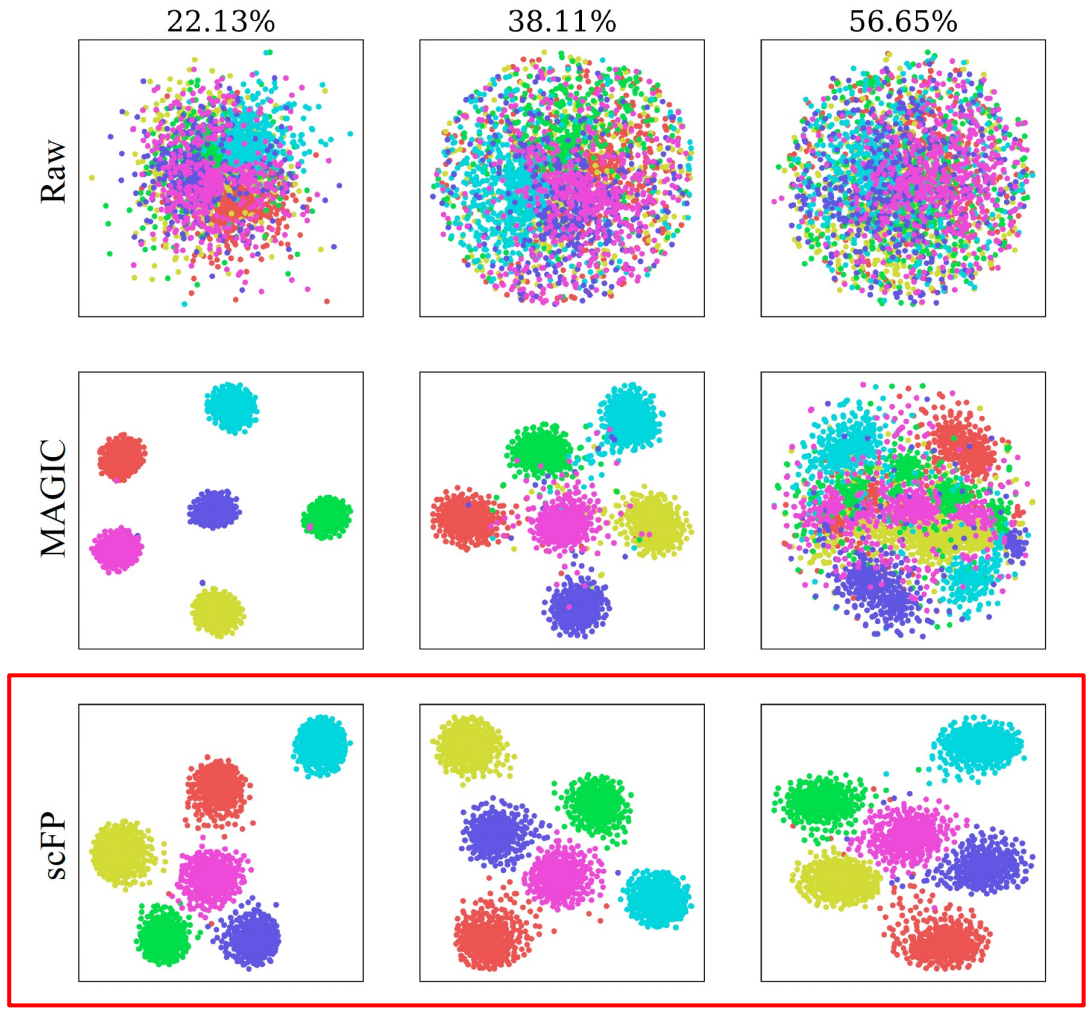
scFP (Ours)

c_1	4.8	0.0	4.2
c_2	5.2	0.0	4.8
c_3	0.0	4.0	0.0



Hard → Soft

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



1) Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol* **18**, 174 (2017). https://doi.org/10.1186/s13059-017-1305-0

EXPERIMENTS

- **Memory & Time Complexity**

Model	Matrices & Parameters in GPU	Big-O
DCA (Eraslan et al., 2019)	$X, f_{enc}, f_{dec}, W_\pi, W_\mu, W_\theta$	$\mathcal{O}(NM) + \mathcal{O}(model)$
AutoClass (Li et al., 2022)	$X, f_{enc}, f_{dec}, f_{cls}$	$\mathcal{O}(NM) + \mathcal{O}(model)$
scGCL (Xiong et al., 2023)	$X, A, f_{o\_enc}, f_{t\_enc}, f_{dec}, W_\pi, W_\mu, W_\theta, q_\theta$	$\mathcal{O}(NM) + \mathcal{O}(N^2) + \mathcal{O}(model)$
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	$\mathcal{O}(NM) + \mathcal{O}(N^2)$

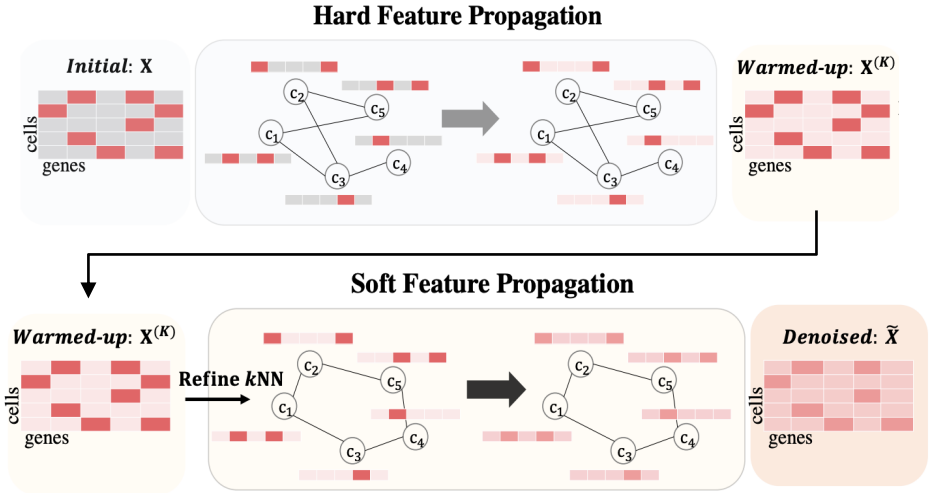
< Memory Complexity $(N: \#\ of\ Cells,\ M: \#\ of\ Genes)$ **>**

< 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



QR Codes

Paper

Code (Github)