DSAIL Data Science & Artificial Intelligence

KAIST

AI for Cellular Systems

From Representation to Perturbation and Delivery

Chanyoung Park

Associate Professor, KAIST

Industrial and Systems Engineering

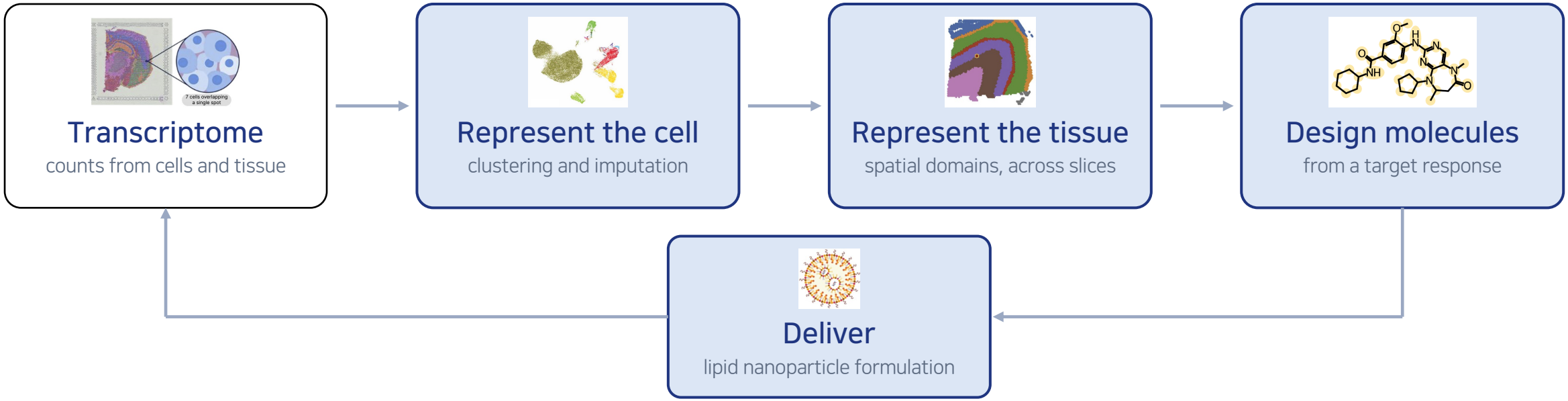
Graduate School of Data Science

Graduate School of AI

cy.park@kaist.ac.kr

One pipeline, from the cell to the delivery vehicle

- The first half reads a transcriptome, the second half writes one

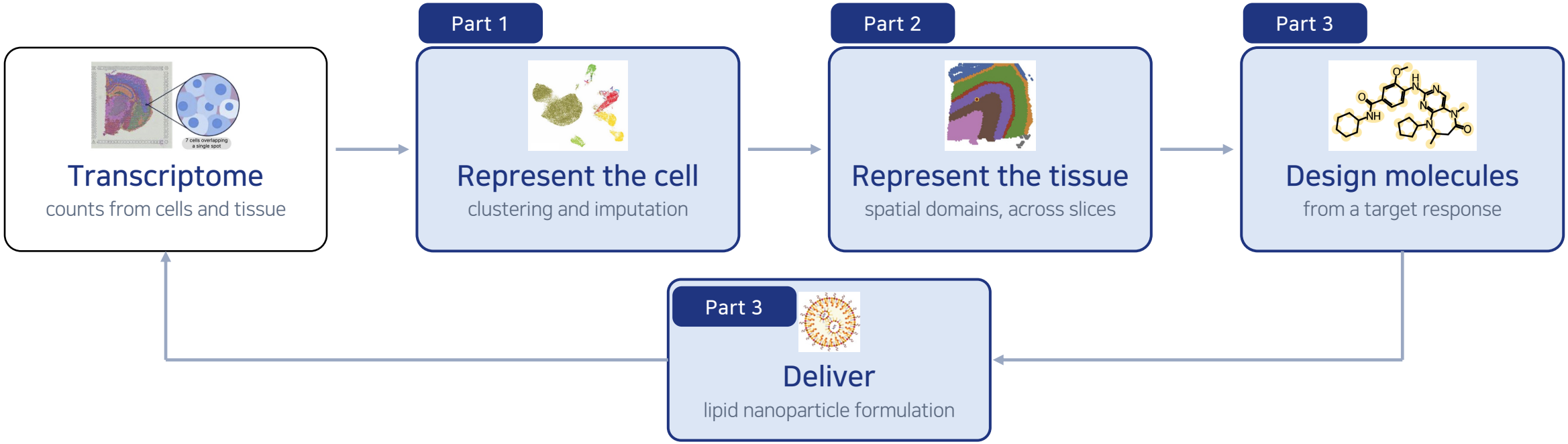


The loop closes: what we deliver is measured again as a transcriptome

- **Read the cell**: group the cells that are alike, and fill in the readings the assay missed
- **Place it in tissue**: the same cell type means something different depending on where it is
- **Design and deliver**: specify the response we want, then find the molecule and the carrier that produce it

One pipeline, five papers

- Five papers from my lab, placed on the same pipeline



The loop closes: what we deliver is measured again as a transcriptome

Part 1: Represent the cell

- scGPCL and scBFP
- Cell clustering, and imputation of the values the assay missed

Part 2: Represent the tissue

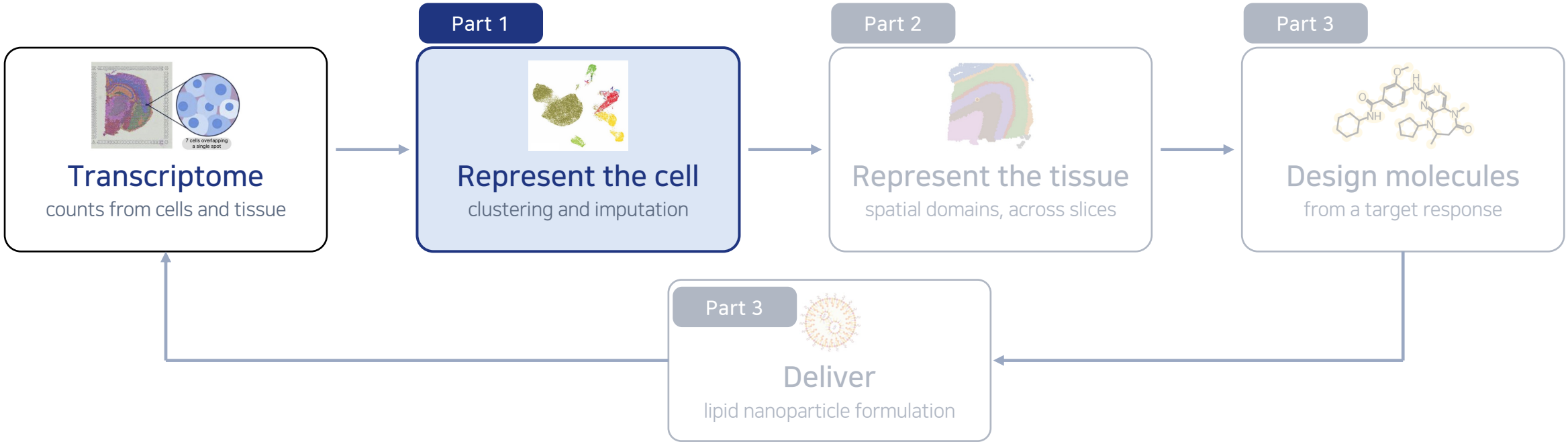
- Spotscape
- Global similarity for spatial transcriptomics, within and across slices

Part 3: Design and deliver

- GxEvo and STRATA
- Transcriptome-conditioned molecular design, and lipid nanoparticle screening

Part 1: Representing the cell

- Part 1: **turning a noisy count matrix into something you can analyze**



Part 1: Represent the cell

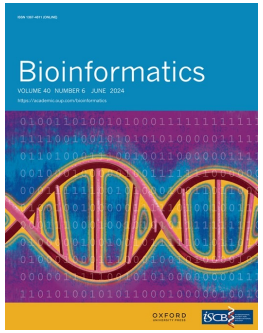
- scGPCL and scBFP
- Cell clustering, and imputation of the values the assay missed

Part 2: Represent the tissue

- Spotscape
- Global similarity for spatial transcriptomics, within and across slices

Part 3: Design and deliver

- GxEvo and STRATA
- Transcriptome-conditioned molecular design, and lipid nanoparticle screening

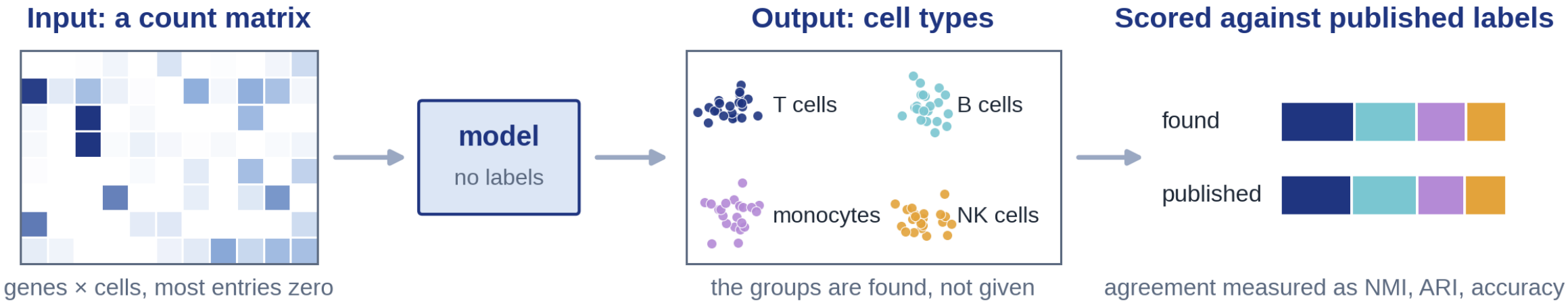


Deep single-cell RNA-seq data clustering with graph prototypical contrastive learning

Bioinformatics 2023

Junseok Lee, Sungwon Kim, Dongmin Hyun, Namkyeong Lee, Yejin Kim, Chanyoung Park

The task: find the cell types, with no labels given



- **Cell typing comes first**, and every later comparison depends on it
- **Nobody tells the model what the groups are**, so it has to find them in the counts alone
- **Most of the table is empty**, and a zero may mean the gene is off or that the assay missed it

The data: public datasets, and controlled simulations

Simulated Data

Assume three situations in which learning cell representations may be challenging

- Case 1 : Gene-expression matrix is highly sparse due to the dropout phenomena
- Case 2 : The expression signal is relatively weak for clustering
- Case 3 : Subgroups of the cells are imbalanced in number

Real Data

Conduct experiments on real scRNA-seq datasets over various sequencing platforms

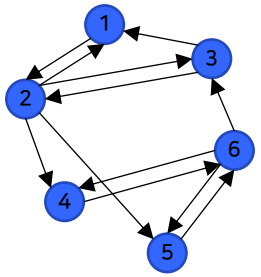
Dataset	Sequencing platform	Cells	Cell types
Mouse ES cells	Droplet barcoding	2,717	4
Mouse bladder cells	Microwell-seq	2,746	16
Zeisel mouse cortex	STRT-seq UMI	3,005	9
Worm neuron cells	sci-RNA-seq	4,186	10
10X PBMC	10X	4,340	8
Human kidney cells	10X	5,685	11
Shekhar mouse retina	Drop-seq	27,499	19

Prior work: learn on a cell-to-cell network

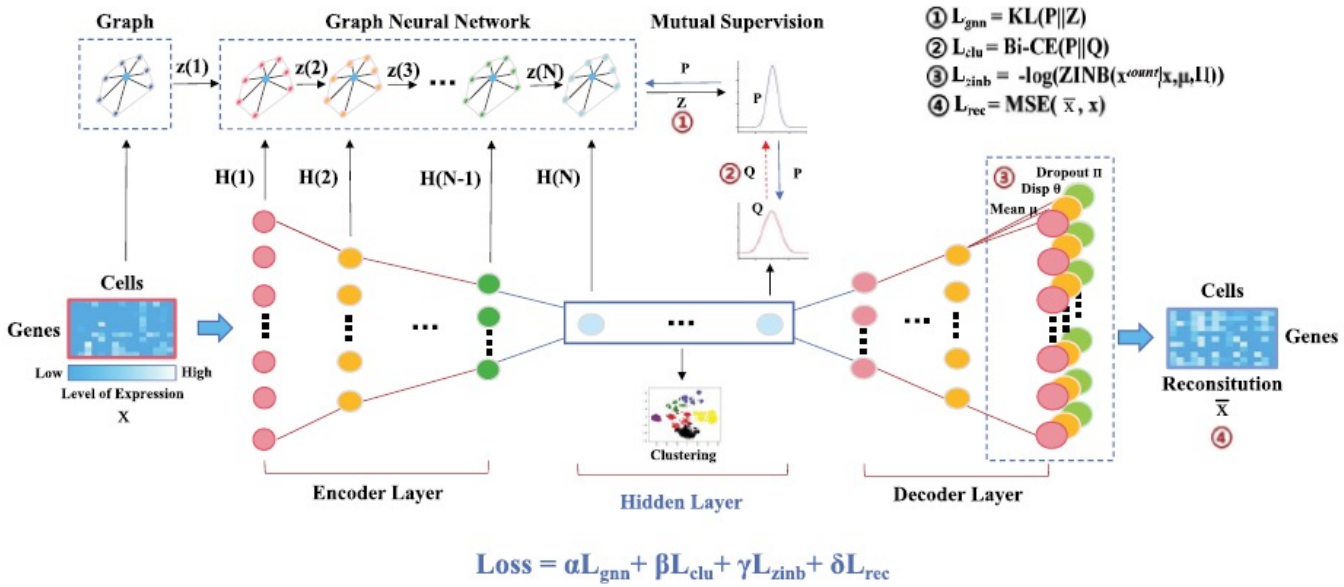
	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	Gene 7
Cell 1	5	0	18	0	0	0	0
Cell 2	10	29	0	5	0	0	0
Cell 3	0	40	5	0	10	0	0
Cell 4	0	0	0	19	0	37	0
Cell 5	9	0	0	0	0	0	3
Cell 6	0	0	0	97	0	0	8

Calculate
Cell-Cell Similarity

KNN Graph



K (=2) Nearest Neighbor Construction

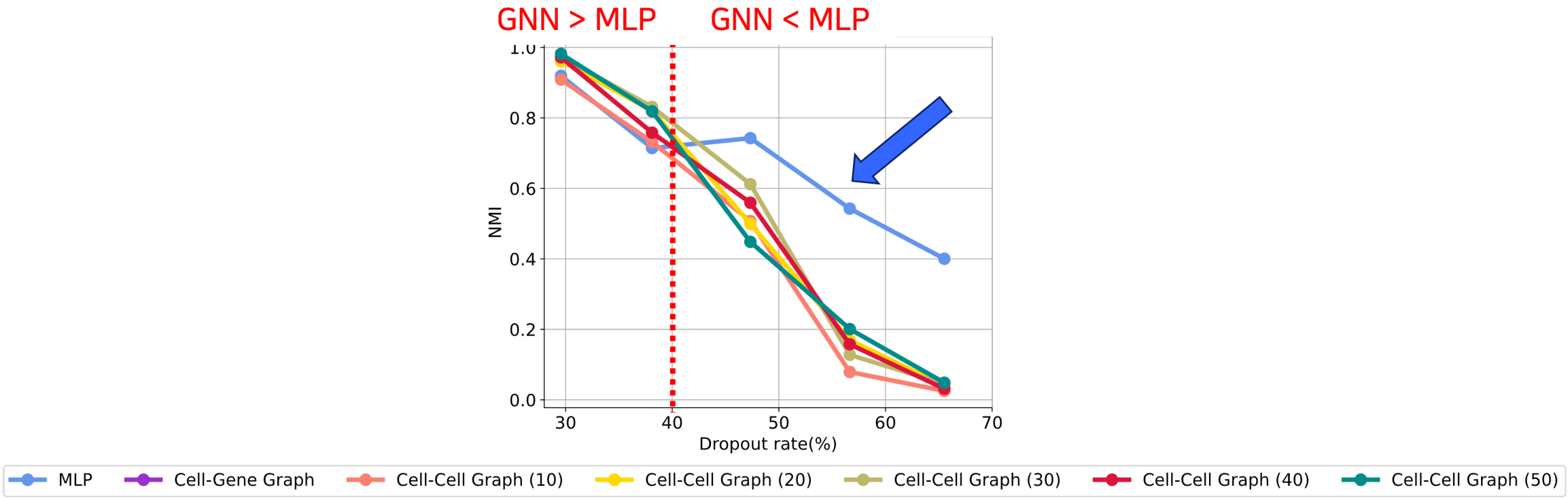


Idea Connect cells that look similar into a graph, and let a GNN share information along it

Limitation Depends on the quality of constructed cell-cell graph
(scRNA-seq data is inherently sparse and noisy)

[Briefings in Bioinformatics 2022] Deep Structural clustering for single-cell RNA-seq data jointly through autoencoder and graph neural networks

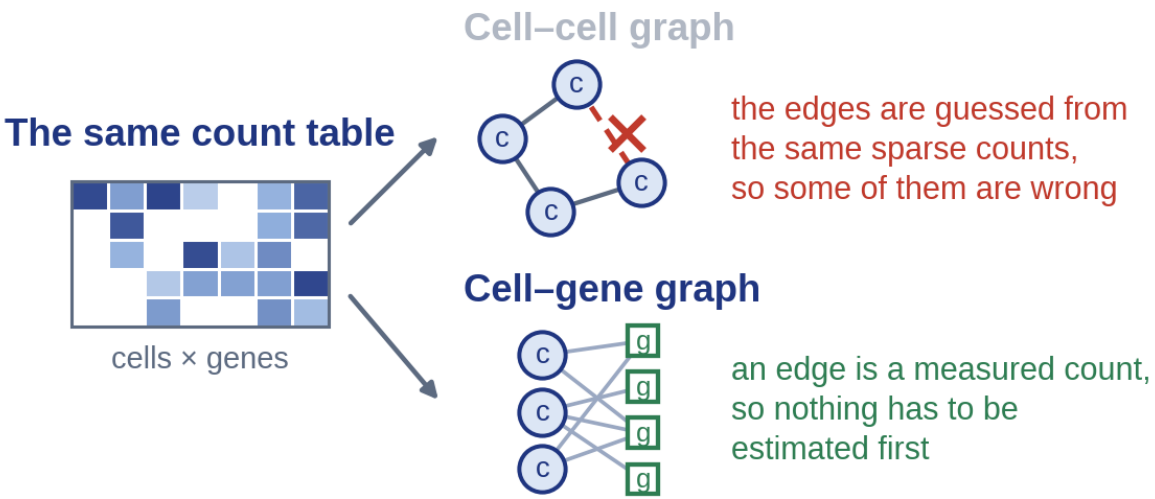
Performance degradation of cell-cell graph with sparse data

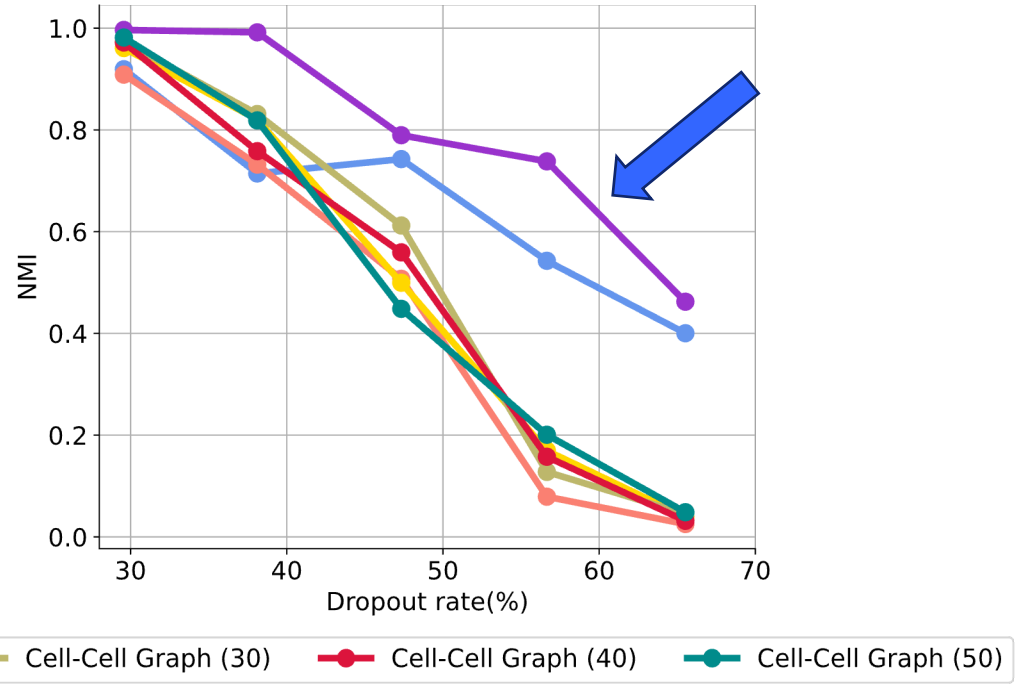


Clustering Performance over various dropout rates on cell-gene matrix

MLP performs better than GNNs as dropout rate increases

The cell-gene graph holds up where the cell-cell graph fails





Clustering Performance over various dropout rates on cell-gene matrix

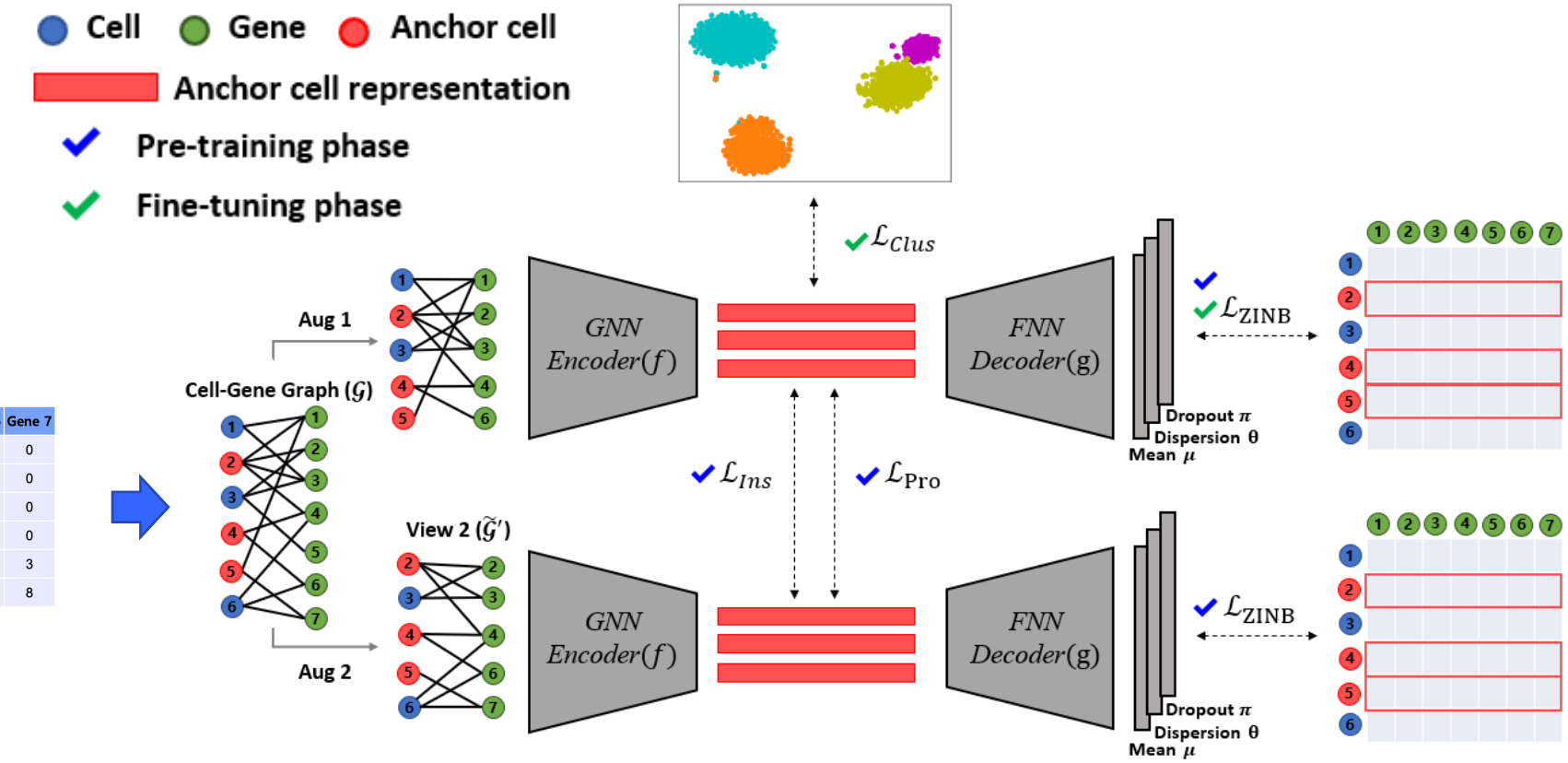
Our Idea Leverage the co-expression-based **cell-gene graph** that preserves the natural relationship

Our idea: the count table is already a network

Key Idea

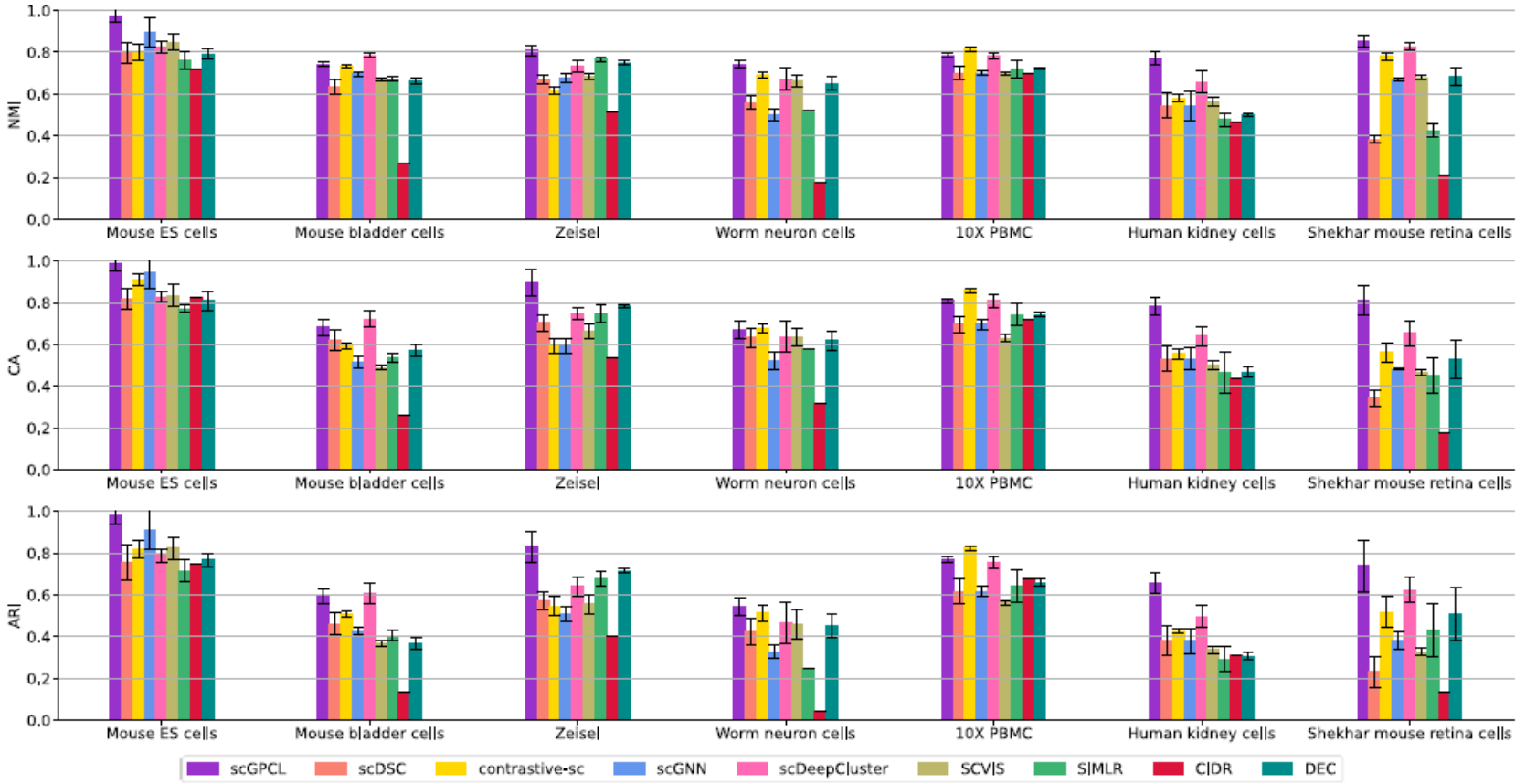
- The cell-gene graph is observed, not inferred, so it adds no new error
- Train on agreement between two augmented views, not on reconstruction
- Compare each cell with its own group, so cells of the same type are not pushed apart

	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	Gene 7
Cell 1	5	0	18	0	0	0	0
Cell 2	10	29	0	5	0	0	0
Cell 3	0	40	5	0	10	0	0
Cell 4	0	0	0	19	0	37	0
Cell 5	9	0	0	0	0	0	3
Cell 6	0	0	0	97	0	0	8



It wins on real data

Comparison on real data



- scGPCL leads on five of the seven datasets and stays close to the best on the other two

→ All of this still assumes the counts can be trusted, and that is where the next paper starts



Single-cell RNA sequencing data imputation using bi-level feature propagation

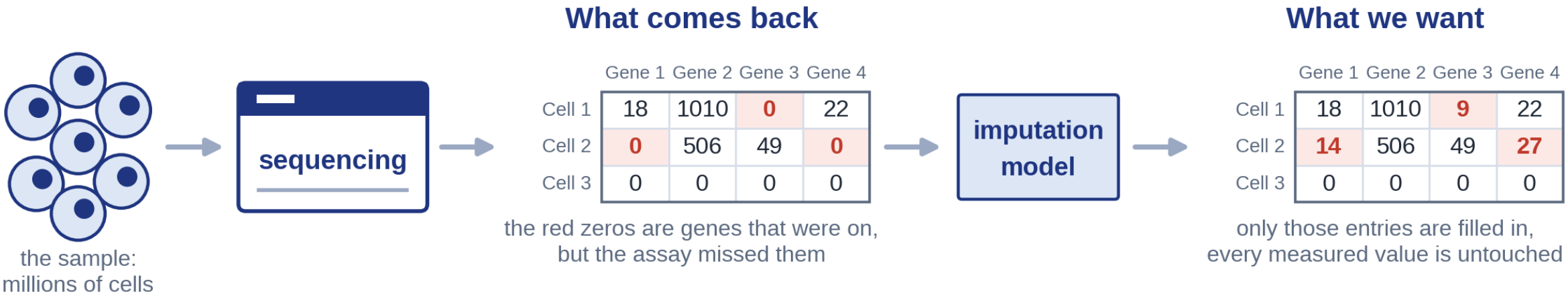
Briefings in Bioinformatics 2024

Sukwon Yun*, Junseok Lee*, Yeongmin Kim, Tianlong Chen, Manolis Kellis, Chanyoung Park

Best Paper Award, ICML 2023 Workshop on Computational Biology

The task: recover the readings the assay missed

Target task Imputation (Denoising) on scRNA-seq data, to improve the analyses that follow



- The true expression is never observed, so to score the result we hide values that were measured and compare after filling them back in

The data: eight datasets, four evaluations

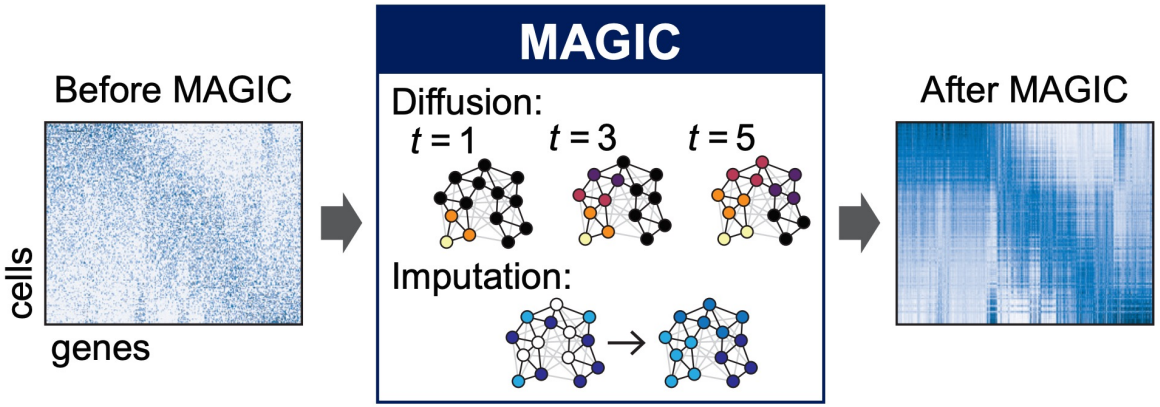
Dataset	Organism and tissue	Platform
Baron Human	human pancreas	inDrop
Baron Mouse	mouse pancreas	inDrop
Pancreas	human pancreas, atlas benchmark	several
Klein	mouse embryonic stem cells	inDrop
Mouse bladder	mouse bladder	Microwell-seq
Zeisel	mouse cortex and hippocampus	STRT-seq
Worm neuron	C. elegans neurons	sci-RNA-seq
Macosko	mouse retina, 44,808 cells	Drop-seq

Gold-standard cell-type labels are available for all eight, so every downstream result can be scored

Evaluation

- **Imputation accuracy**
  - Measured values are hidden, then filled back in and compared
- **Cell clustering**
  - Do the cell types separate more clearly, and do rare types become visible?
- **Differential expression**
  - Genes called differentially expressed are compared with bulk RNA-seq on the same five cell lines
- **Clinical case study**
  - Lung carcinoid tumor against normal lung tissue

Prior work: smooth every value over a cell network



1) Calculate the Exponentiate Markov matrix M (KNN Graph)
2) Iterate diffusion process until convergence

$$D_{imputed} = M^t * D$$

Initial cell-gene matrix

Idea

Graph structure can capture relationships among cells
→ leveraging message-passing schemes for information propagation

Limitation

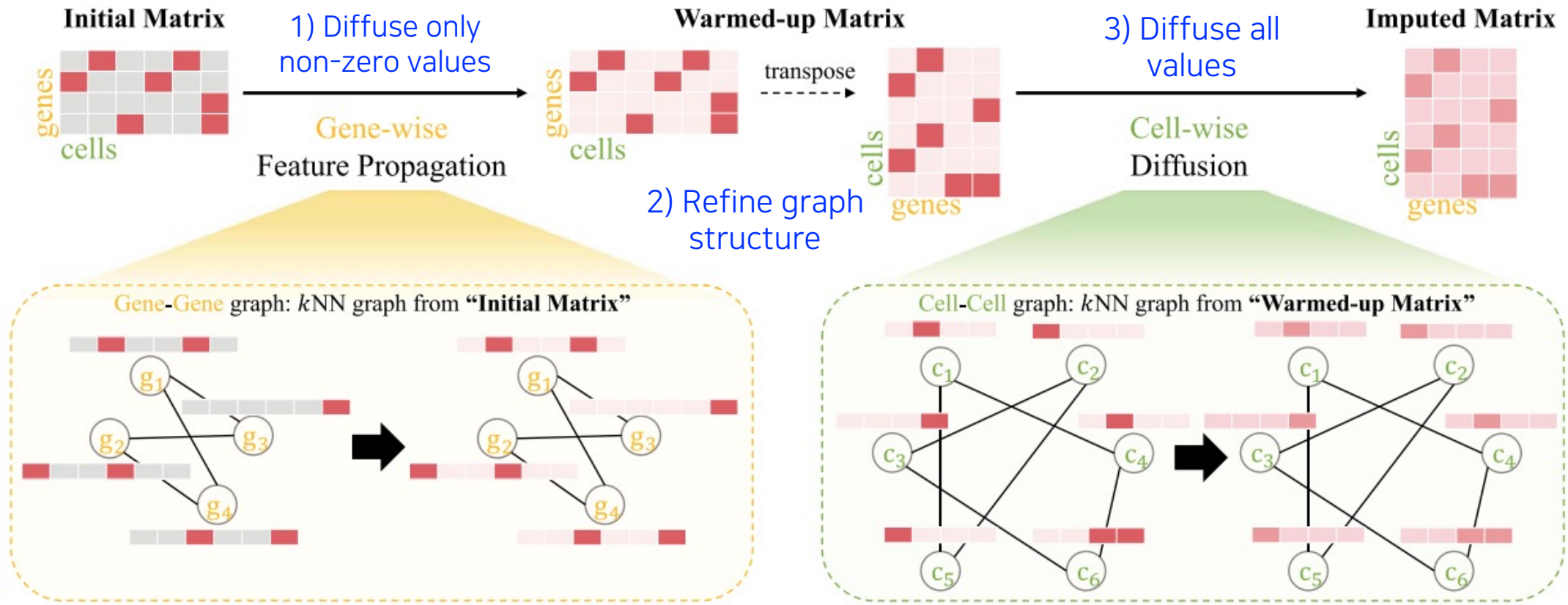
- Depends on the quality of constructed cell-cell graph (scRNA-seq data is inherently sparse and noisy)
- Diffusing zero values involves the risk of propagating noise, when the diffused zeros are false zeros (dropout)
- Does not leverage relationship information between genes

[Cell 2018] Recovering Gene Interactions from Single-Cell Data Using Data Diffusion

Our idea: fill the zeros before you trust the cell graph

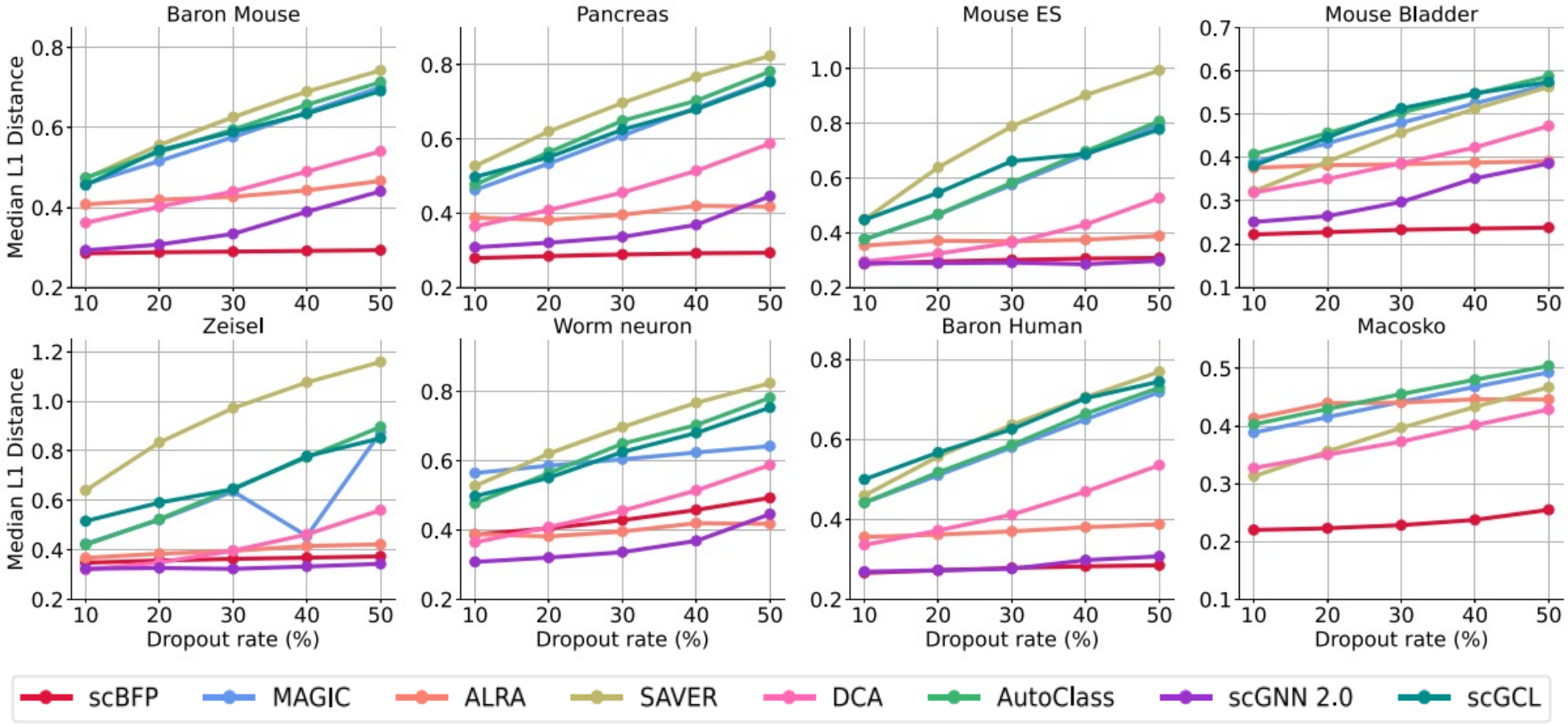
Key Idea

- Step 1: Diffuse only non-zero values to mitigate the risk of propagating false-zeros to non-zero values
- Step 2: Refine the kNN graph after the warm-up phase to construct a cell-cell graph
- Step 3: Leverage cell-cell relationships by conducting diffusion on the refined cell-cell graphs



The filled values are close to the truth

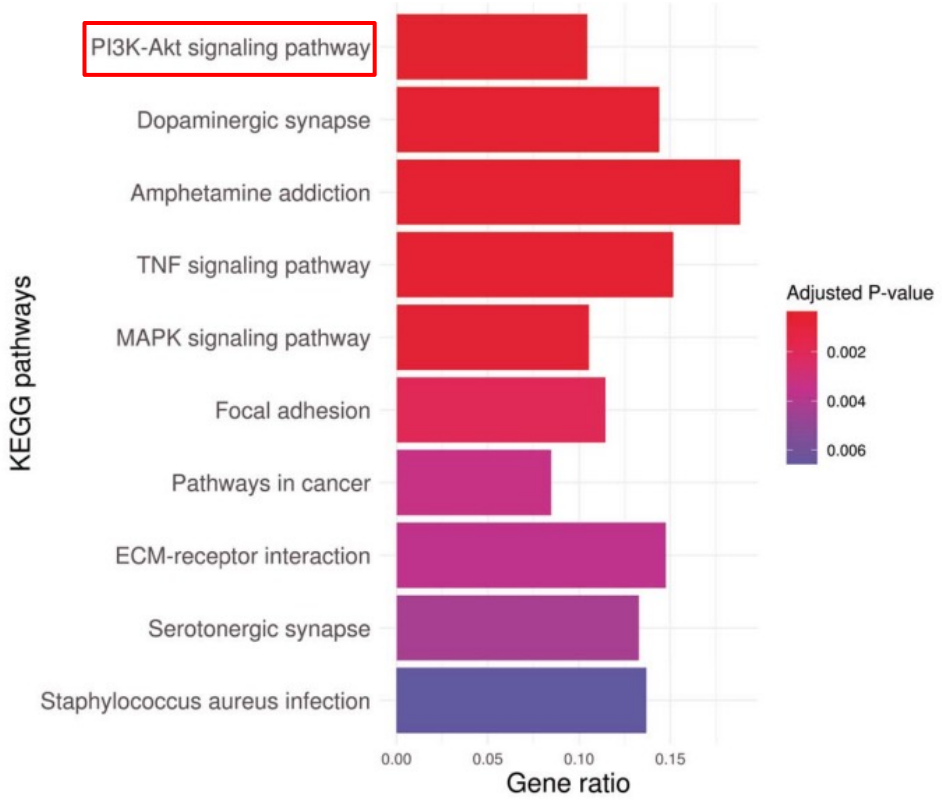
Dropout value recovery



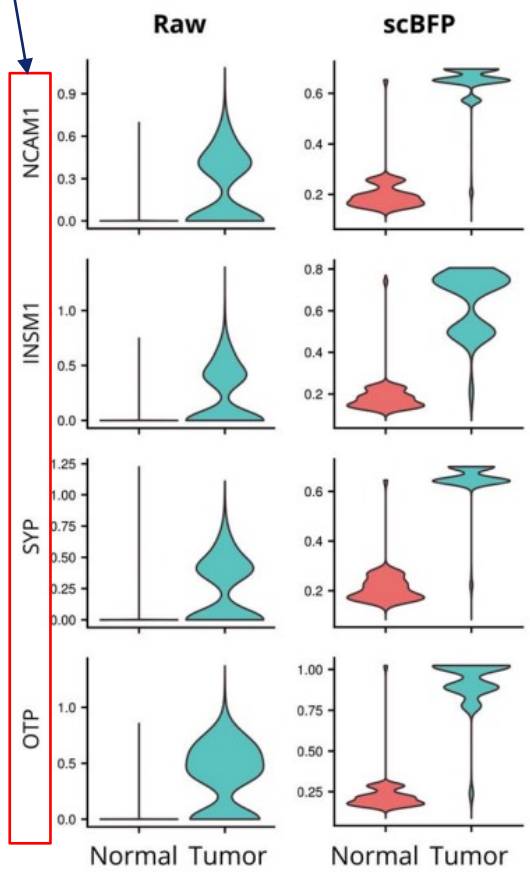
- scBFP succeeds in robustly recovering the original values, even at a high dropout rate

The biology comes through

Enrichment Analysis (Lung carcinoid tumor)



4 marker genes for lung carcinoid



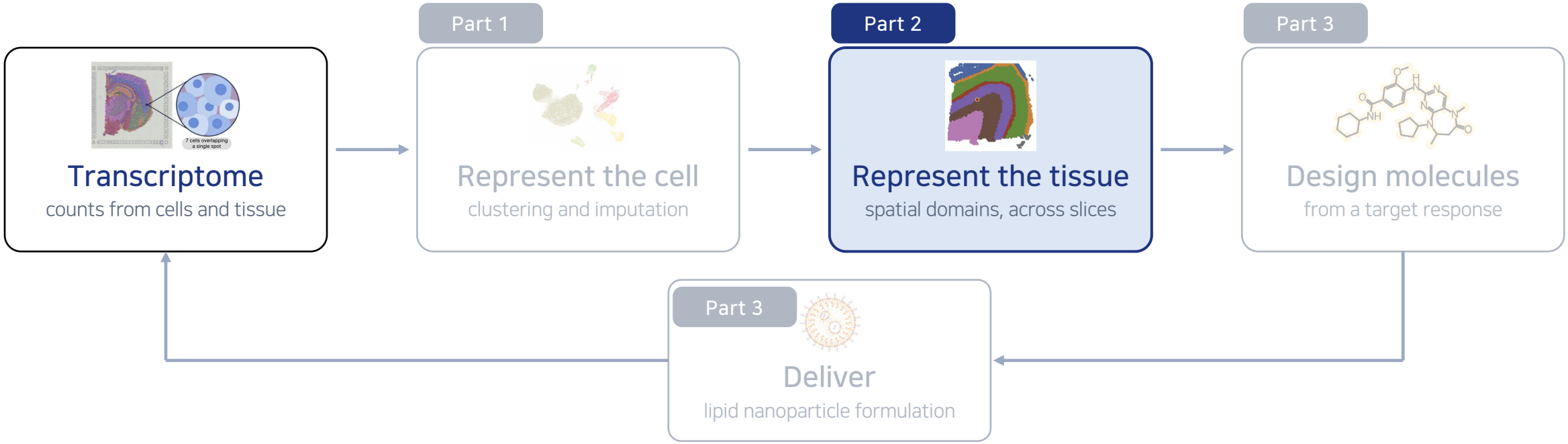
In normal tissue almost every value is zero; filling them in makes the contrast between normal and tumor clear

- DEGs derived from scBFP show a clear enrichment in cancer-relevant KEGG pathways

  → For example, **'PI3K/Akt signaling pathway'** plays a role in promoting tumor cell growth

Part 2: Representing the tissue

- Part 2: **the same cell type means something different depending on where it is**



Adding spatial coordinates helps, and creates a new problem at the borders

Part 1: Represent the cell

- scGPCL and scBFP
- Cell clustering, and imputation of the values the assay missed

Part 2: Represent the tissue

- Spotscape
- Global similarity for spatial transcriptomics, within and across slices

Part 3: Design and deliver

- GxEvo and STRATA
- Transcriptome-conditioned molecular design, and lipid nanoparticle screening



ICML
International Conference
On Machine Learning

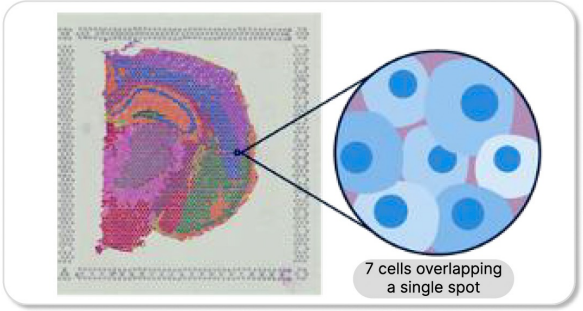
Global Context-aware Representation Learning for Spatially Resolved Transcriptomics

ICML 2025

Yunhak Oh, Junseok Lee, Yeongmin Kim, Sangwoo Seo, Namkyeong Lee, Chanyoung Park

The data: expression with coordinates attached

- Limitation of scRNA-seq data: Loss of the **spatial context** of cells due to tissue dissociation
- Spatially resolved transcriptomics (SRT) incorporates the spatial context and gene expression of cells
  - Allows for the identification of where specific genes are expressed within tissue
  - Enables the study of disease, such as cancer, by observing gene expression patterns in different tissue regions



	Gene 1	Gene 2	Gene 3	Gene 4
Spot 1	18	1010	0	22
Spot 2	0	506	49	0
Spot 3	0	0	0	72

Dataset	Tissue	Technology	Used for
DLPFC	Human prefrontal cortex	10x Visium, 50 μm spots	Spatial domains, slice integration
MTG	Human middle temporal gyrus, control and Alzheimer's	10x Visium, 50 μm spots	Integrating control and Alzheimer's slices
Mouse embryo	Whole mouse embryo, several stages	Stereo-seq, 0.2 μm	Aligning developmental stages
NSCLC	Human lung cancer	NanoString CosMx, subcellular	Spatial domains at subcellular resolution
Breast cancer	Human breast cancer	10x Visium and 10x Xenium	Aligning samples and technologies

Image credit: https://www.10xgenomics.com/analysis-guides/integrating-single-cell-and-visium-spatial-gene-expression-data

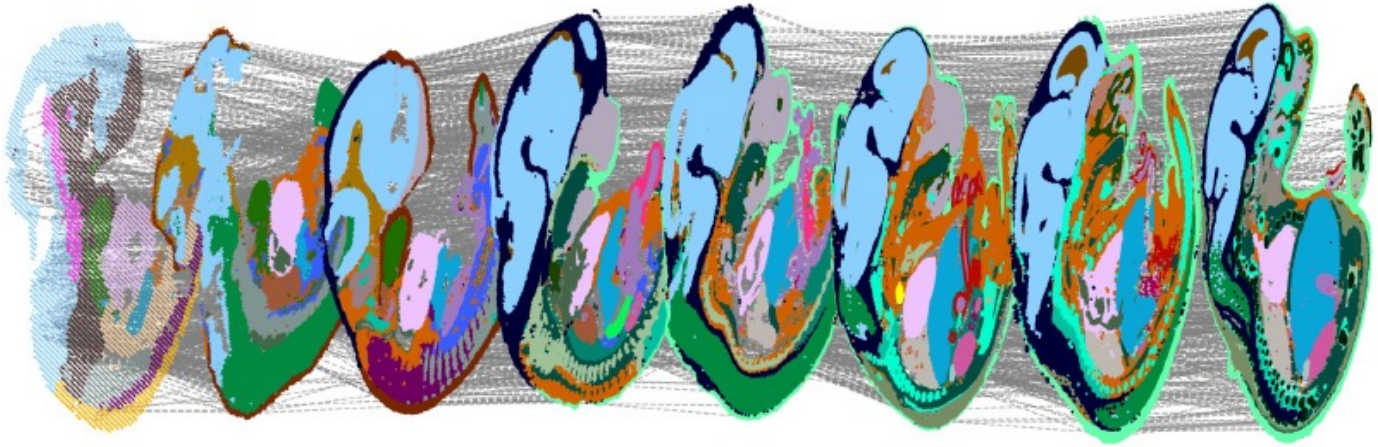
The task: recover tissue structure, and combine slices

Target data Spatially resolved **transcriptomics (SRT)**

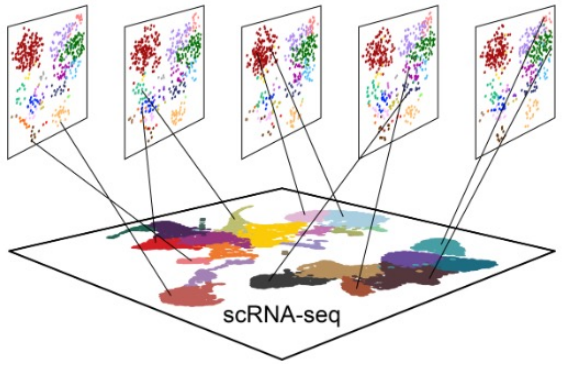
Target task

- Identify spatial domain (**Clustering**)
- **Align** spatial data across multiple slices
- **Integrate** the representations of multiple slices

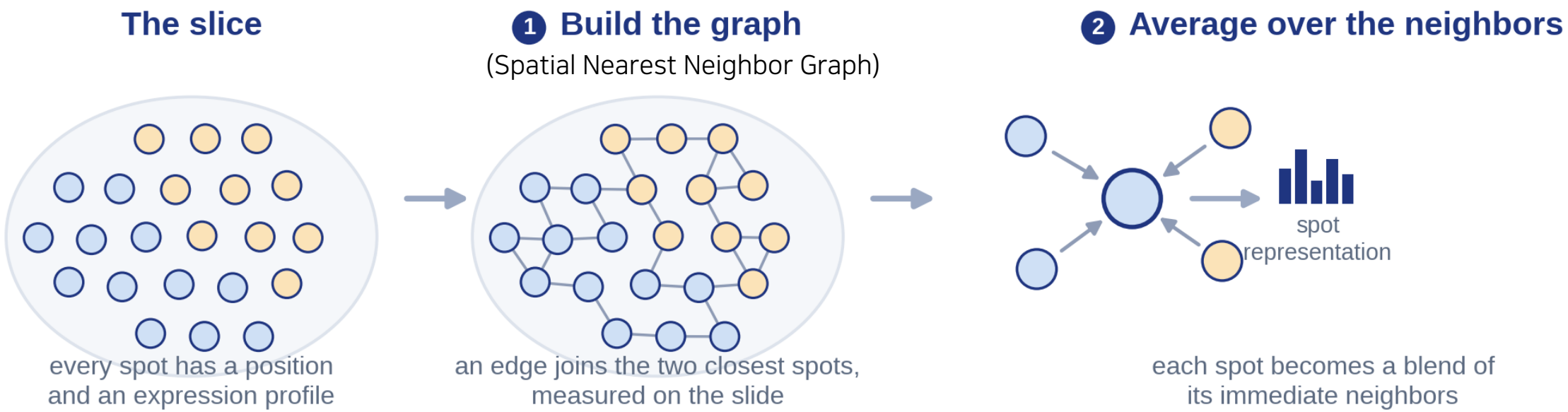
- Alignment across development stages



- Integration of multiple slices



Prior work: average over the neighboring spots



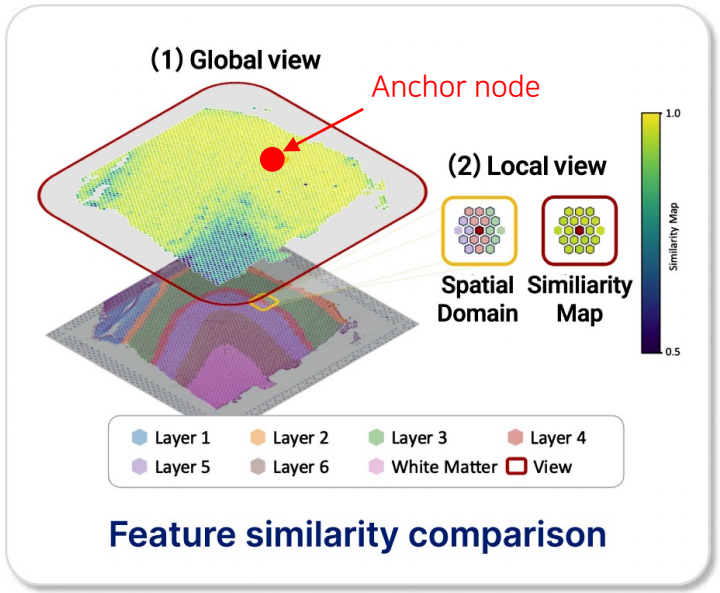
Idea Build a graph from the spot coordinates and let a GNN blend each spot with the spots next to it

Limitation A spot and its neighbors look almost the same, so the blending adds little, and spots on a domain border get smeared into their surroundings

STAGATE, SpaceFlow (2022), STAligner (2023), SLAT (2023), GraphST (2024)

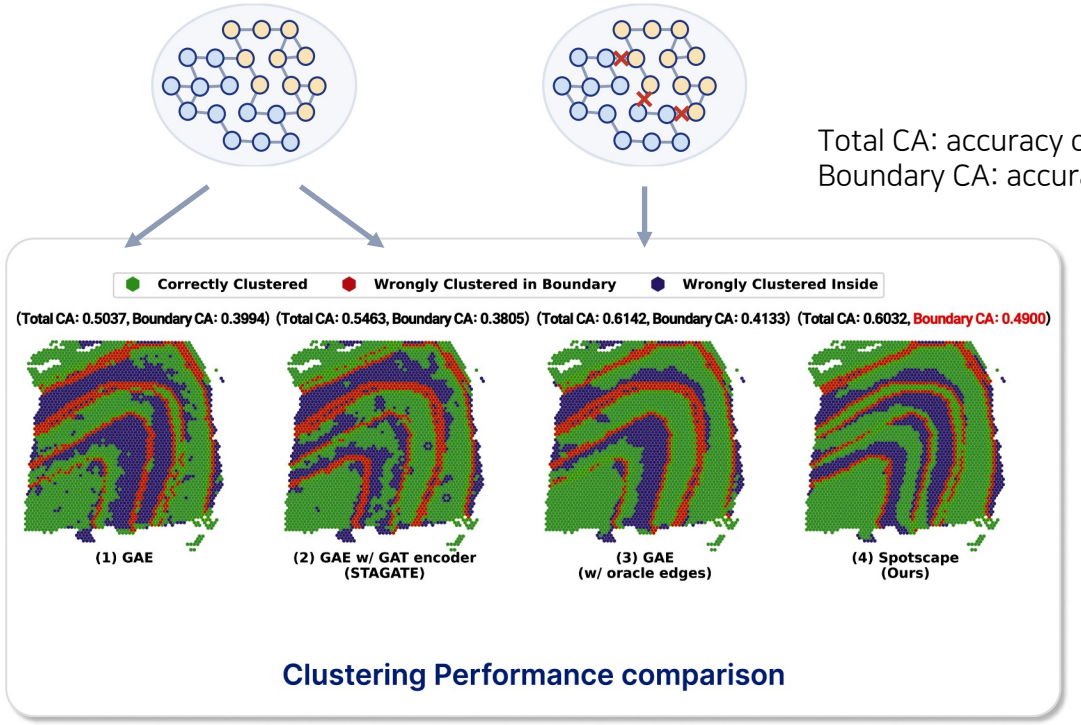
Why that fails at the borders

- Gene expression values vary smoothly along spatial coordinates



Dorsolateral prefrontal cortex (DLPFC) data

Local (spatially close) spots have high similarity, regardless of the spatial domain



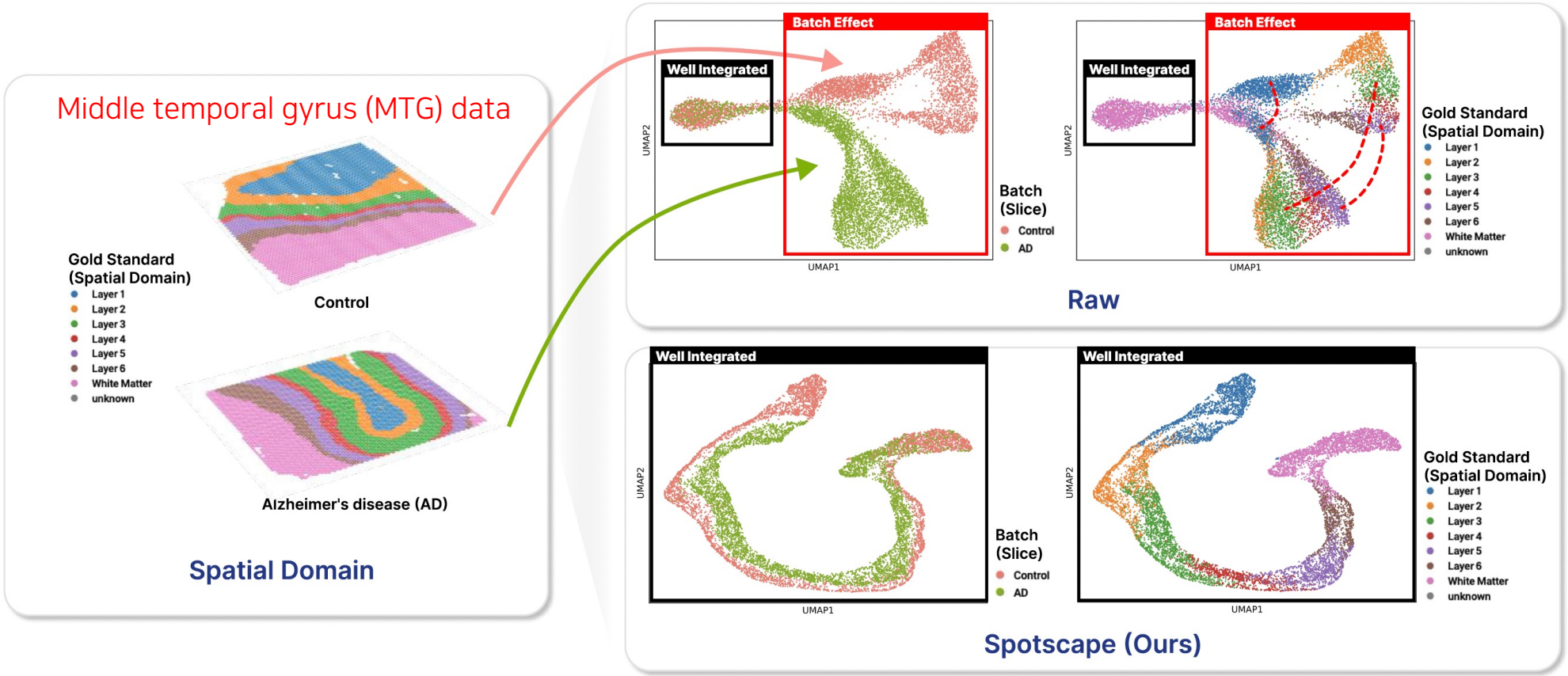
Total CA: accuracy over all spots
Boundary CA: accuracy at the borders

(1) Neighboring spots look almost identical, so averaging adds little
(2) Attention cannot separate them either
(3) Even a perfect neighborhood leaves the border broken
(4) Ours keeps the border accurate without losing overall accuracy

Should capture relationship between spots in global context

And slices do not line up with each other

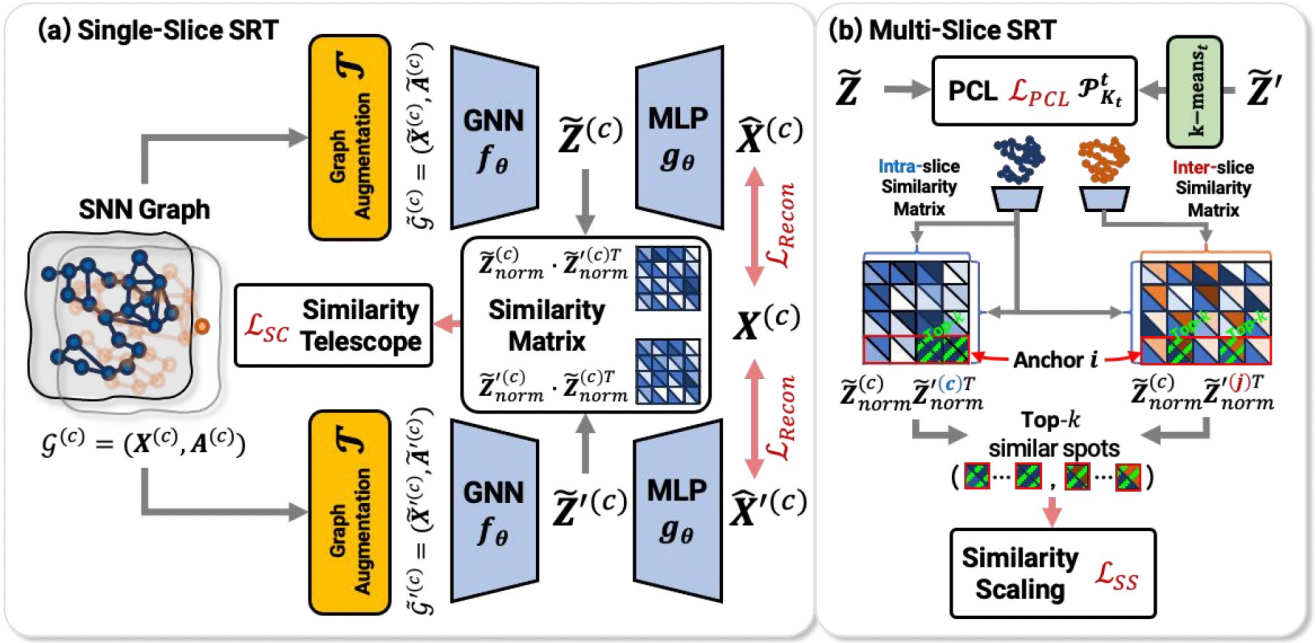
- Gene expression profiles from the same slice cluster together unexpectedly, regardless of their biological relevance

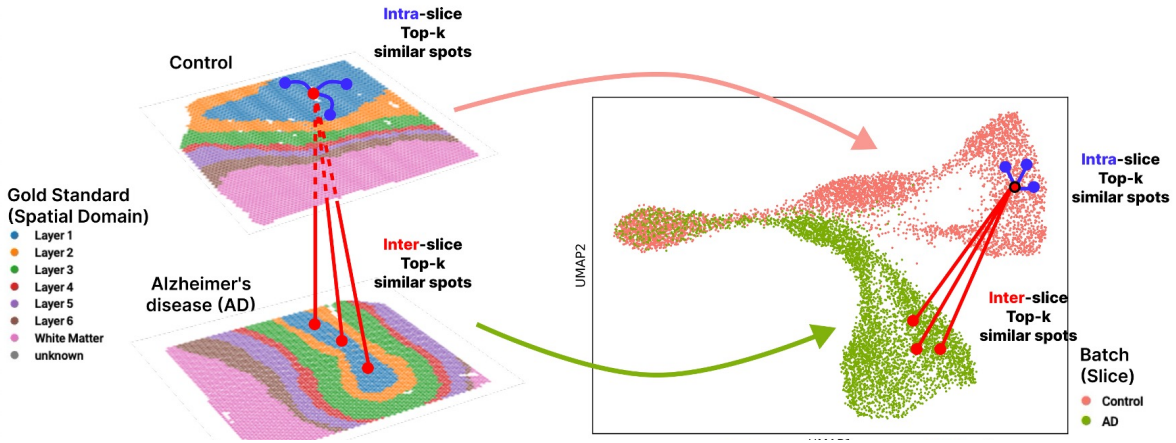


Should alleviate batch effects to extend multi-slice tasks

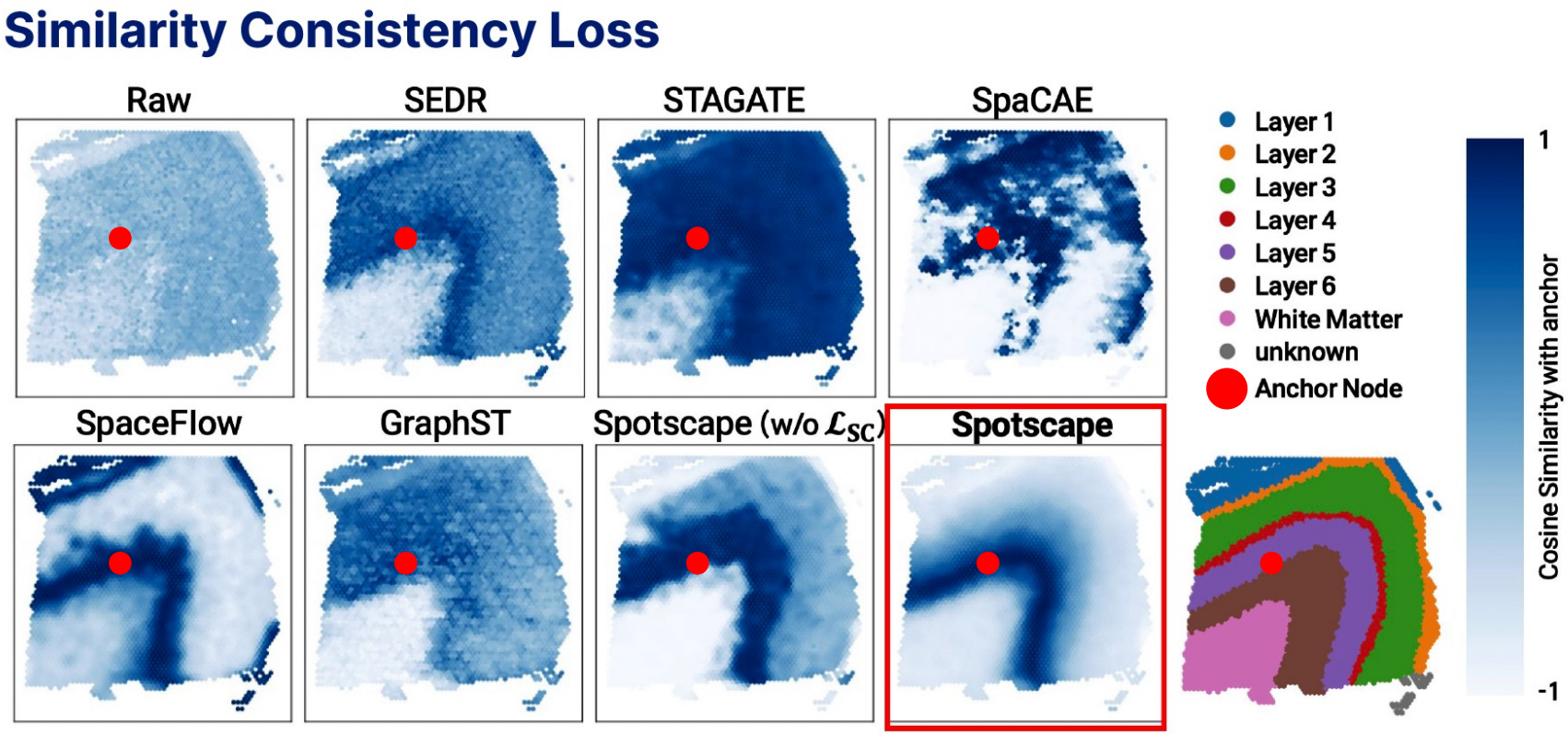
Our idea: compare every spot with every other spot

- Key ideas
  - (1) Capture **global relationships between spots** by reflecting the global context among multiple spots
  - To mitigate batch effects,
    - (2) **balance the scales of intra-slice and inter-slice similarity** between spots
    - (3) **group spots from the same spatial domain** while distancing others





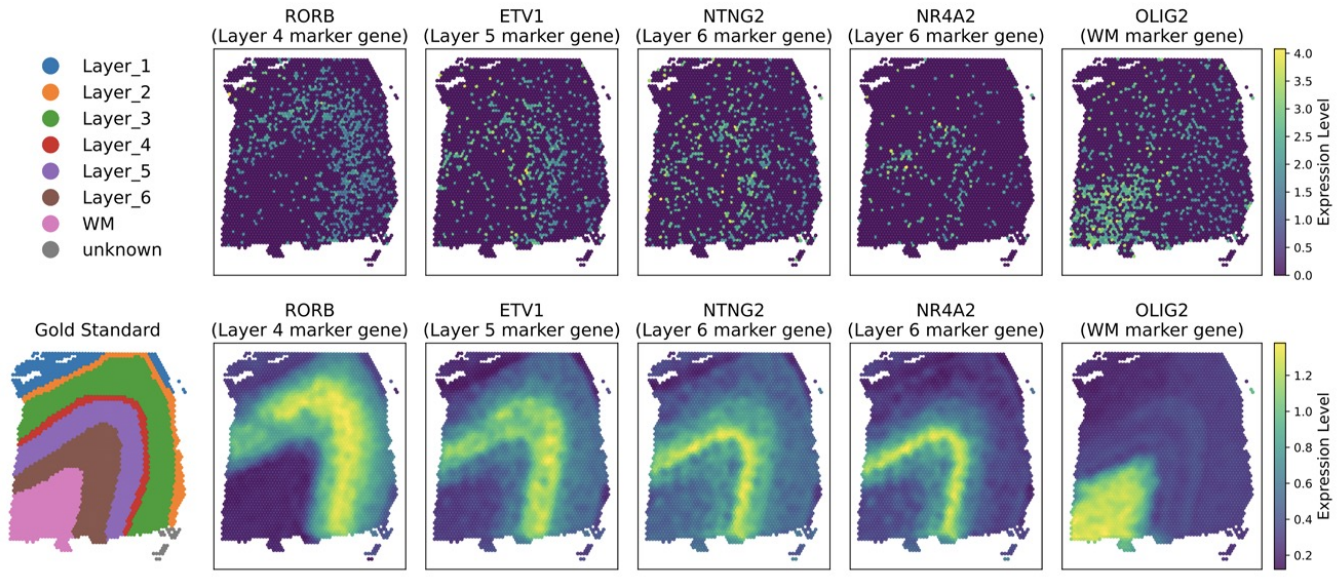
The border becomes visible



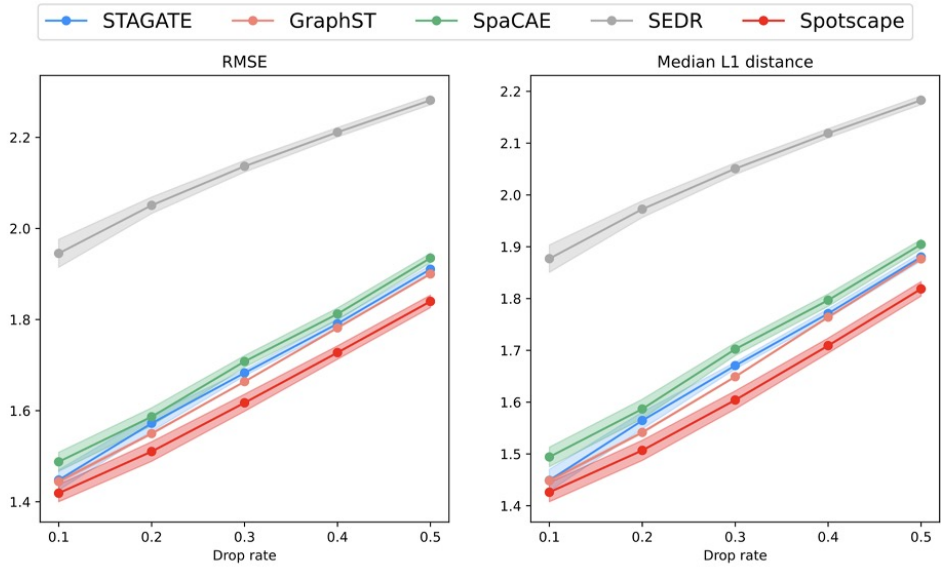
Spotscape captures relative similarity between spots regardless of the actual distance

Denoising and Imputation

Denoising



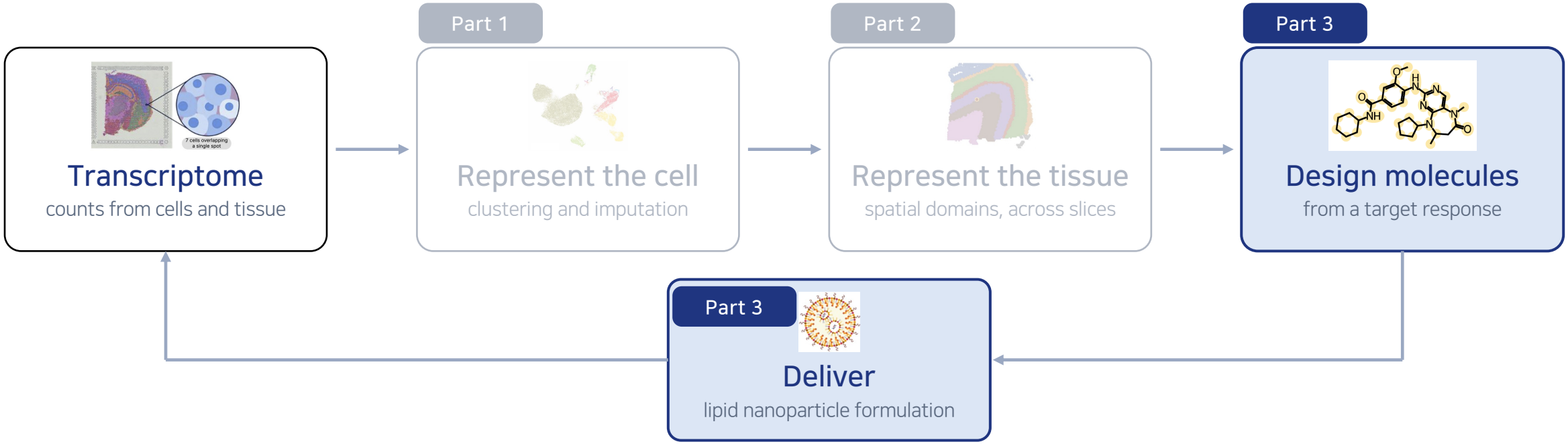
Imputation



- **Denoising**: clean up the values we already have. Markers that were buried in the raw counts become legible
- **Imputation**: fill in values we do not have. We hide measured values, recover them, and score the error at every drop rate

Part 3: Designing and delivering

- Part 3: **from reading a transcriptome to writing one**



The response we want becomes the specification for what to make, and how to deliver it

Part 1: Represent the cell

- scGPCL and scBFP
- Cell clustering, and imputation of the values the assay missed

Part 2: Represent the tissue

- Spotscape
- Global similarity for spatial transcriptomics, within and across slices

Part 3: Design and deliver

- GxEvo and STRATA
- Transcriptome-conditioned molecular design, and lipid nanoparticle screening

GxEvo: Evolutionary Search for Transcriptome-Conditioned Molecular Design

Under review

The task: transcriptome-based drug design

Target-based design



Transcriptome-based design (TBDD)

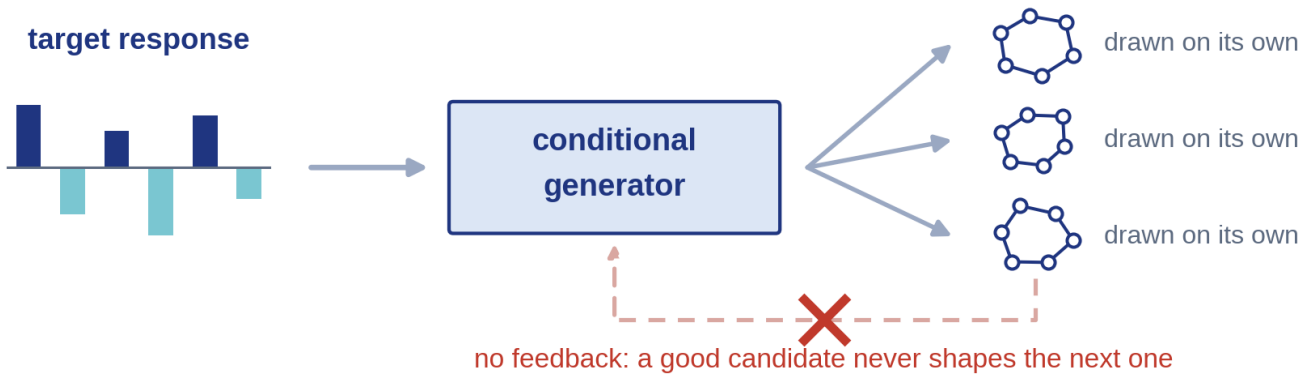


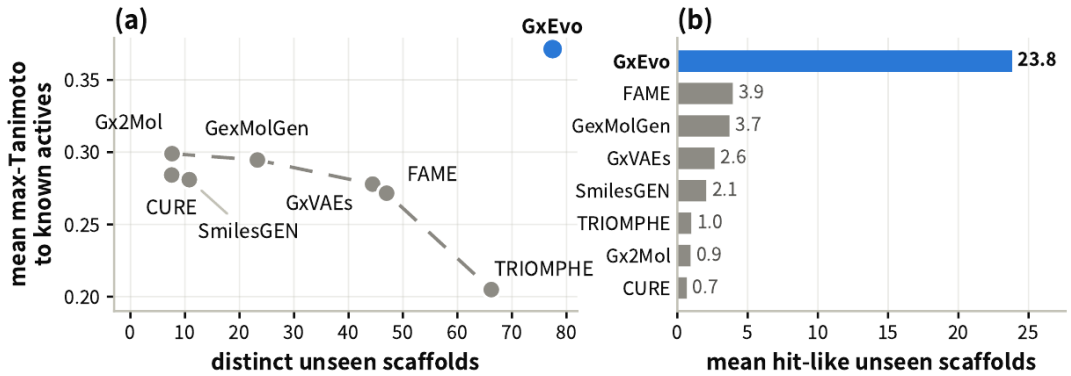
How it is checked

1. Knock down one gene, e.g. AKT1, in MCF7 cells, and record the response
2. Ask for molecules that reproduce that response
3. Good ones should resemble known AKT1 active compounds

- Where the data comes from
  - LINCS L1000: a compound is applied to a cell line and 978 genes are read out. The response is simply the perturbed profile minus the untreated one
  - 85,310 compound and cell-line conditions, 12,313 compounds, 82 cell lines
- What we test on
  - Ten gene-knockdown targets in MCF7

Prior work: every candidate is drawn from scratch

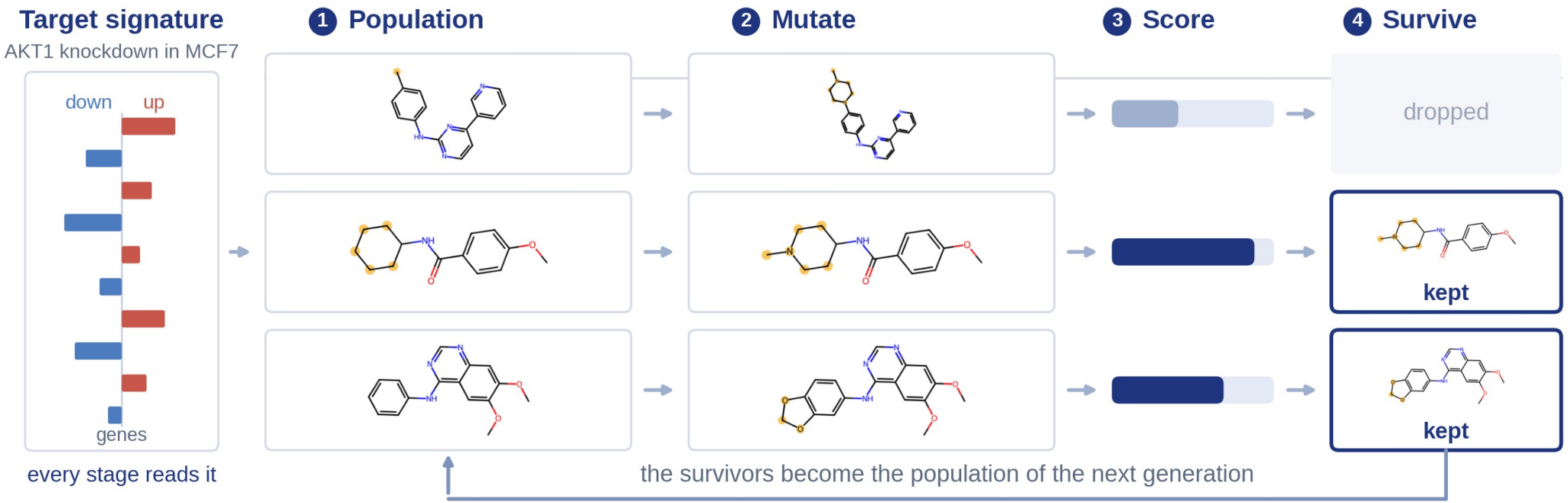




- Why one-shot generation hits a ceiling
  - Every candidate is drawn independently, so the score only ranks the batch at the end
  - So every published method trades one against the other: new scaffolds, or similarity to known actives
- The two things a fix has to do
  - **Raise hit-likeness**: improve a candidate that already scored well, which needs **more than one turn**
  - **Keep scaffold variety**: work from several starting points at once, which needs **a population**

→ Evolutionary search is the one formulation that gives both

Our idea: breed the molecules, do not just sample them



A generic evolutionary algorithm breaks at all three of these steps, so we rebuilt each one

① **Population** a generic search starts from random library molecules → **ours is proposed from the target response**

② **Mutate** a generic search makes a random graph edit → **ours is a learned edit that reads the target**

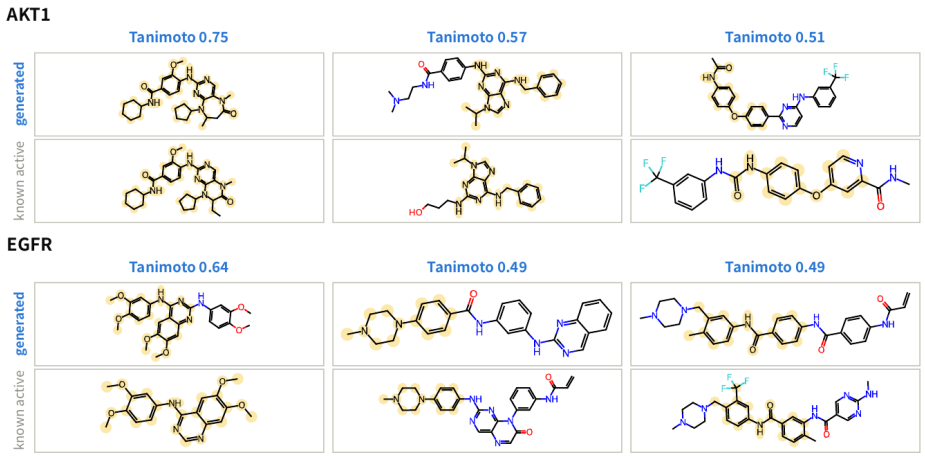
③ **Score** a generic search needs an accurate predictor → **ours only needs to rank candidates correctly**

Novel chemistry that still looks like it works

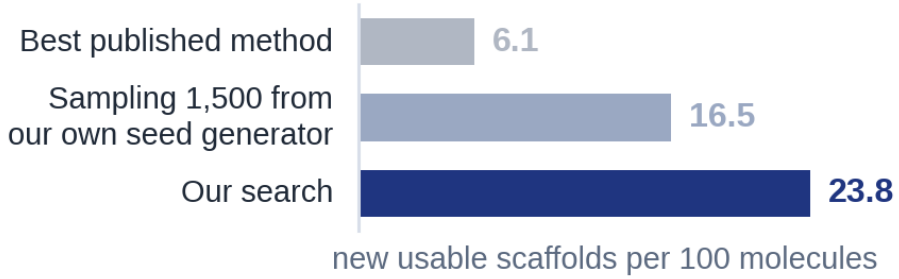
Method	Similar to actives ↑	New usable scaffolds ↑	Copied from corpus ↓
Best prior method	.272	3.9	12%
Most novel prior	.205	1.0	1%
Closest prior match	.299	0.9	85%
GxEvo	**.371**	**23.8**	**3%**

Ten gene-knockdown targets; counts are per 100 valid molecules

Generated, above the nearest known active



Every method gets 1,500 candidates



It finds usable new scaffolds at several times the rate of the strongest published method, without copying the training chemistry.



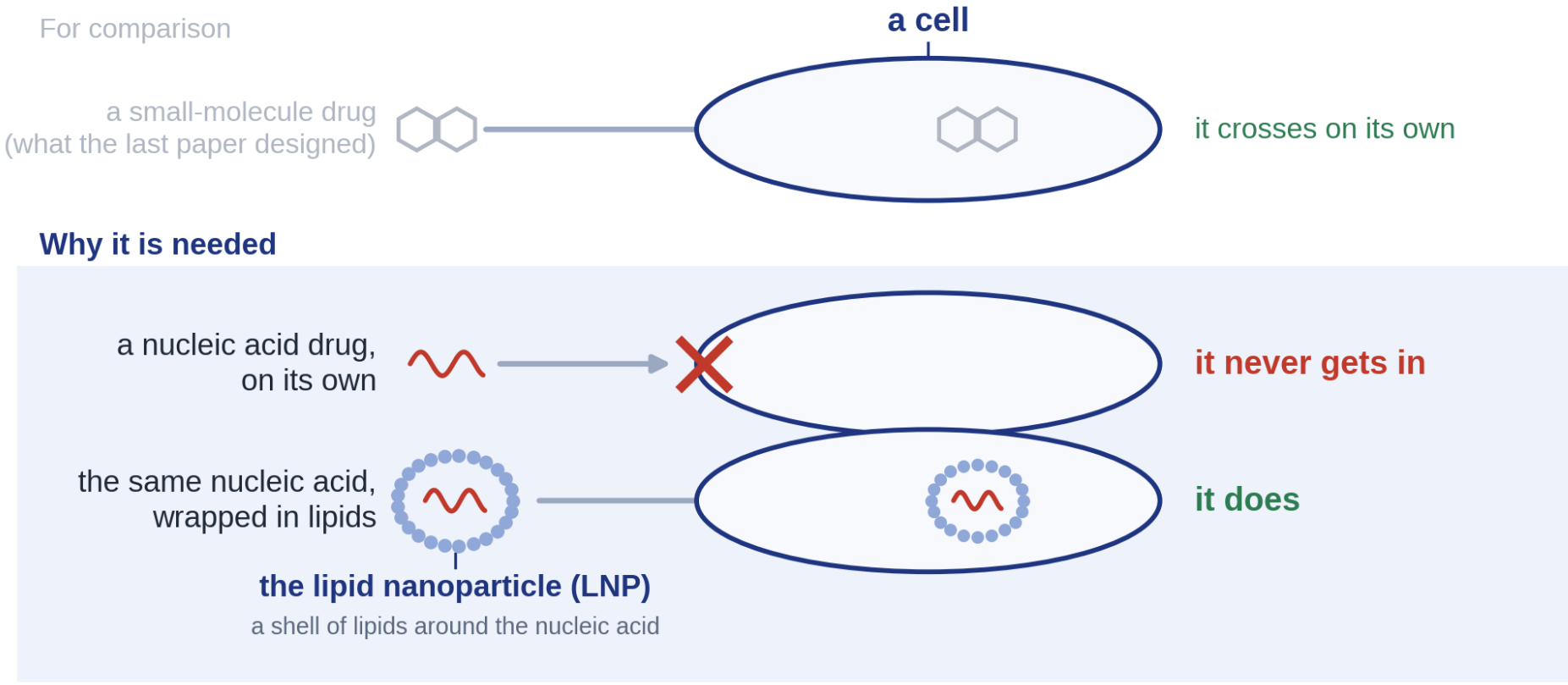
Screening Lipid Nanoparticles through Structure-Ratio Alignment

NeurIPS 2026

Yoonho Lee, Yunhak Oh, Hoyoung Choi, Chanyoung Park

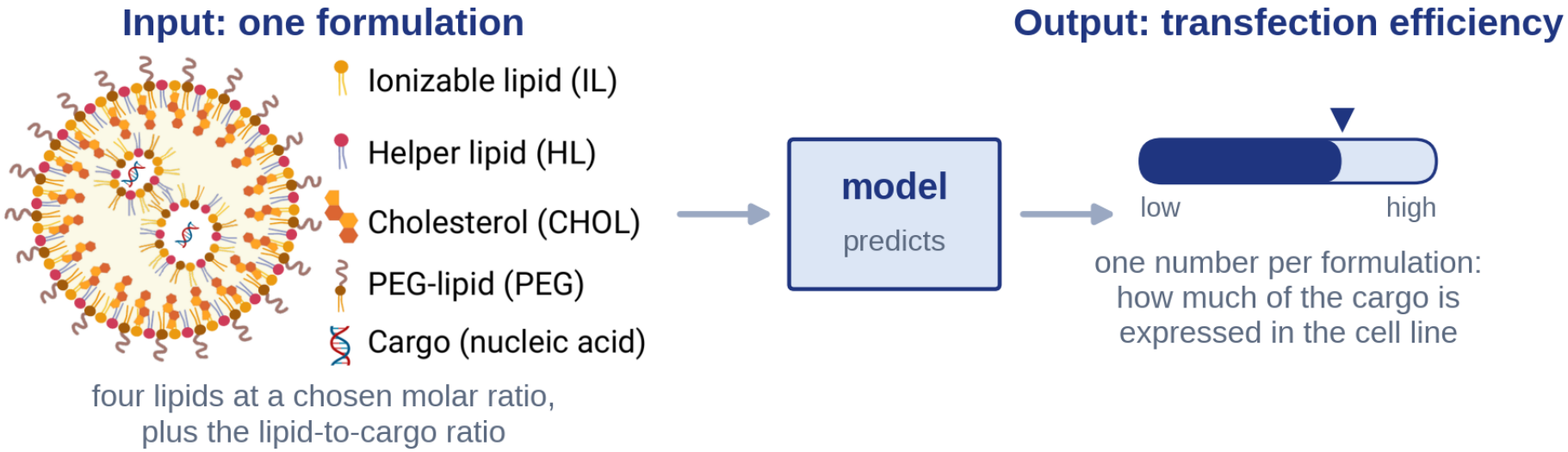
What is a lipid nanoparticle?

A shell of lipids wrapped around a nucleic acid, so that the nucleic acid can get into a cell



→ The lipids and their proportions set how much cargo gets into the cell, and that number still has to be measured

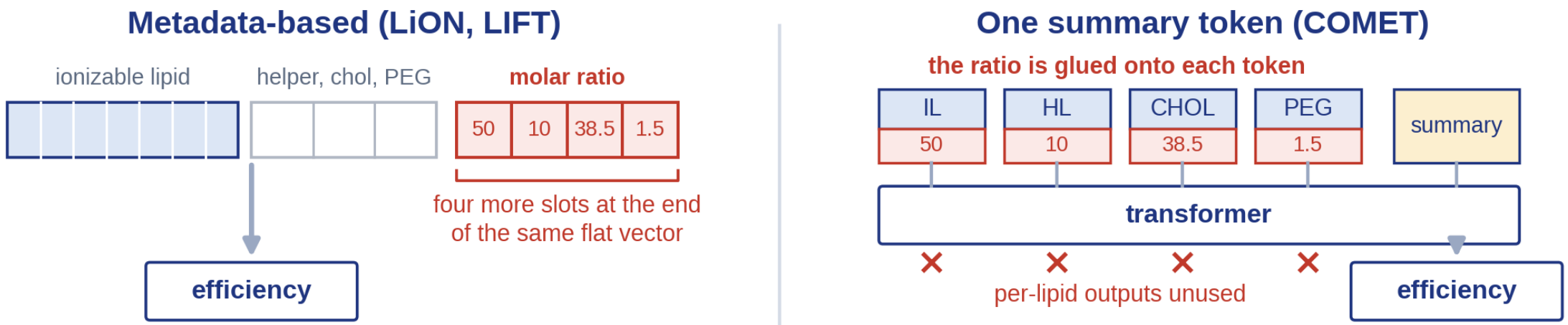
The task: predict whether a formulation will deliver



The ratio matters as much as the lipids: the same four lipids at a different ratio can go from working well to barely working

Dataset	Molecules IL / HL / CHOL / PEG	Formulations
A549	192 / 3 / 1 / 1	1,801
YZ2022	1 / 6 / 1 / 1	1,080
LC2024 (HEK293)	1 / 6 / 1 / 1	1,080
LC2024 (N2a)	1 / 4 / 1 / 1	720

Prior work: the ratio is attached as a number



① The ratio is just a number

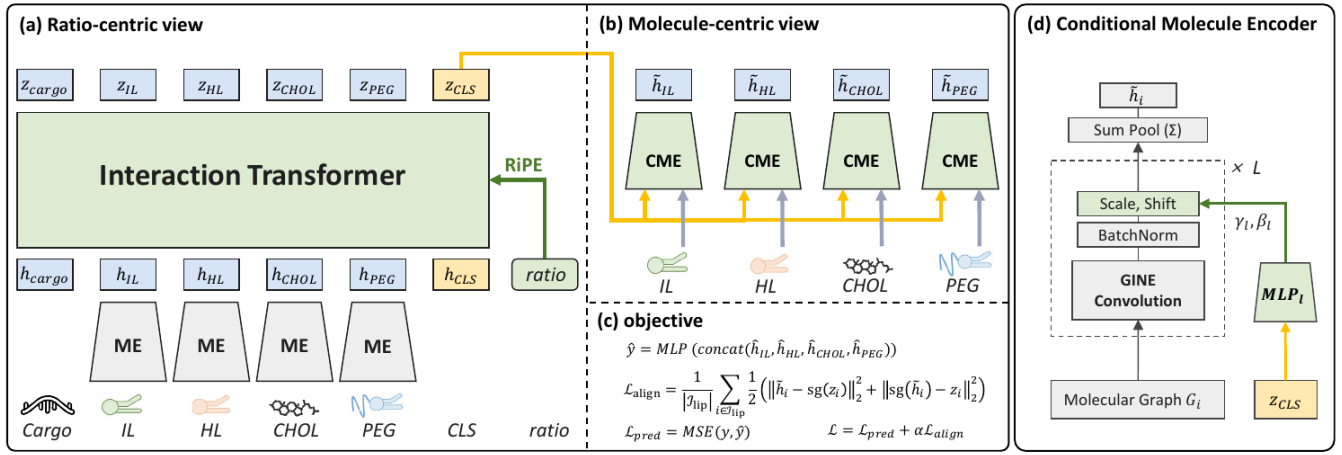
10:20 and 20:40 are the same mixture, and nothing in the model says so

② Each lipid is encoded alone

only the summary of the interaction is used, so no lipid carries its context

→ Make the ratio part of the attention, and encode each lipid inside the interaction context

Our idea: two views of the same formulation



Two views of one formulation, tied together in training

- **The ratio view: how much of each component**
  - The four components attend to one another, and the ratio enters through the positional encoding
- **The chemistry view: each lipid re-encoded in context**
  - The interaction summary is fed back in as the condition
- **The two are tied together in training**
  - Proportions cannot see structure, and structure cannot see proportion

Why the ratio, not the difference

IL : HL		RoPE: the difference what transformers normally use	RiPE: the ratio ours
10 : 20	the same mixture	20 - 10 = 10	20 / 10 = 2
20 : 40		40 - 20 = 20	40 / 20 = 2
20 : 30	different mixture	30 - 20 = 10	30 / 20 = 1.5
		same reads different, different reads same	same reads same, different reads different

RiPE puts each component at the log of its share, so the difference attention computes is

$$\log r_i - \log r_j = \log (r_i / r_j)$$

It holds up on lipids and ratios never measured

Four datasets, with the test set either drawn at random or built from molecule types held out of training

Metric	MSE (↓)				PCC (↑)			
dataset	A549	YZ2022	LC2024 (N2a)	LC2024 (HEK293)	A549	YZ2022	LC2024 (N2a)	LC2024 (HEK293)
random split								
AGILE-Metadata	0.5644 ± 0.0263	0.6908 ± 0.0783	2.1385 ± 0.3542	3.8305 ± 0.2649	0.7610 ± 0.0180	0.5500 ± 0.0729	0.6660 ± 0.0698	0.4884 ± 0.0528
AGILE-MLP	0.5325 ± 0.0399	0.4007 ± 0.0830	1.5726 ± 0.3292	3.7015 ± 0.2307	0.7792 ± 0.0139	0.7748 ± 0.0460	0.7683 ± 0.0577	0.5176 ± 0.0460
AGILE-COMET	1.0022 ± 0.0657	0.5127 ± 0.0466	1.5963 ± 0.2253	3.3464 ± 0.2654	0.5019 ± 0.0340	0.7040 ± 0.0507	0.7595 ± 0.0426	0.5814 ± 0.0535
STRATA (ours)	**0.4873 ± 0.0290**	**0.3698 ± 0.0667**	**0.7953 ± 0.1295**	**2.5155 ± 0.4417**	**0.8030 ± 0.0128**	**0.8015 ± 0.0452**	**0.8863 ± 0.0225**	**0.7269 ± 0.0558**
gain	8.4883%	7.7115%	49.4277%	24.8297%	3.0544%	3.4461%	15.3586%	25.0258%
unseen molecule split								
AGILE-Metadata	0.7433 ± 0.1001	0.9685 ± 0.6554	3.2135 ± 1.6978	4.6804 ± 1.6085	0.6801 ± 0.0523	0.2831 ± 0.3055	0.4388 ± 0.2666	0.3534 ± 0.2689
AGILE-MLP	0.7390 ± 0.0533	1.1061 ± 0.6109	4.4688 ± 1.3858	6.5759 ± 1.5044	0.6904 ± 0.0262	0.1370 ± 0.3085	-0.0649 ± 0.3489	0.2134 ± 0.3933
AGILE-COMET	1.0772 ± 0.1110	0.9545 ± 0.4914	3.5356 ± 1.7032	5.0261 ± 1.1157	0.4748 ± 0.0570	0.2553 ± 0.2574	0.4065 ± 0.1782	0.3018 ± 0.1582
STRATA (ours)	**0.6558 ± 0.0966**	**0.8652 ± 0.3153**	**2.8728 ± 1.1652**	**4.5784 ± 0.8397**	**0.7415 ± 0.0326**	**0.3451 ± 0.2211**	**0.5051 ± 0.1682**	**0.5209 ± 0.0977**
gain	11.2585%	9.3557%	10.6021%	2.1793%	7.4015%	21.9004%	15.1094%	47.3967%

→ Best on nearly every dataset and split, including the splits where the molecule type was never in training

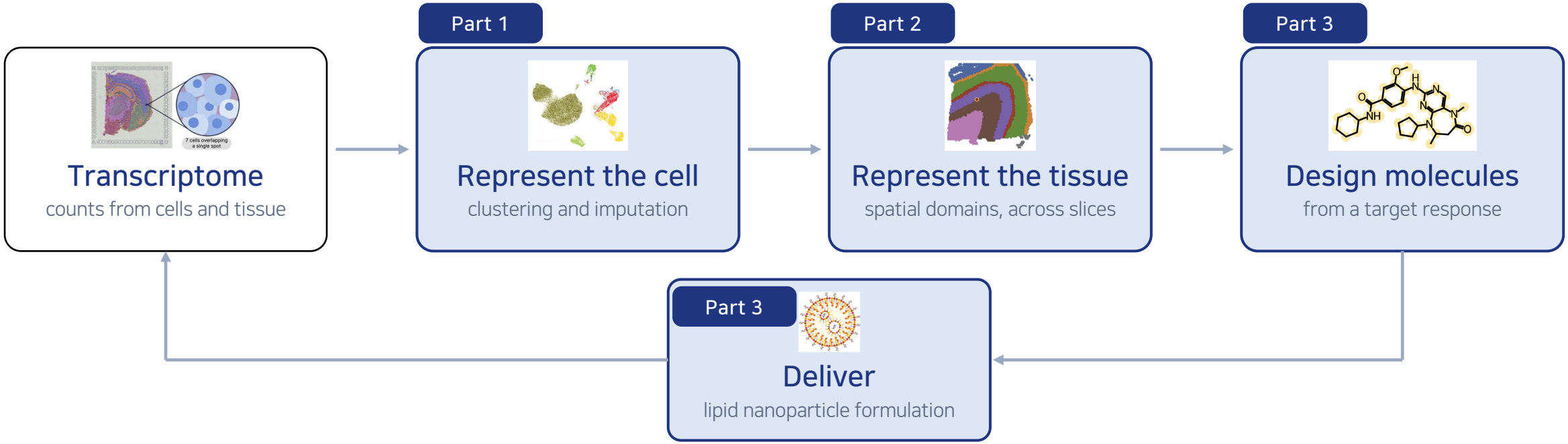
Trained on one cell type, then only the prediction head is retrained on four cell types never seen

Metric	MSE (↓)				PCC (↑)			
dataset	LC2024 (ARPE19)	LC2024 (B16)	LC2024 (N2a)	LC2024 (PC3)	LC2024 (ARPE19)	LC2024 (B16)	LC2024 (N2a)	LC2024 (PC3)
AGILE-Metadata	2.4359 ± 0.1597	2.6538 ± 0.3008	2.7428 ± 0.3376	3.3561 ± 0.3849	0.5364 ± 0.0594	0.5816 ± 0.0348	0.4872 ± 0.0653	0.5913 ± 0.0319
AGILE-MLP	2.1809 ± 0.1685	2.3719 ± 0.3126	2.6853 ± 0.4117	2.8947 ± 0.3418	0.5718 ± 0.0620	0.6275 ± 0.0282	0.5384 ± 0.0625	0.6227 ± 0.0353
AGILE-COMET	2.7186 ± 0.1230	3.1666 ± 0.3438	2.9284 ± 0.3890	4.1724 ± 0.3720	0.3905 ± 0.0492	0.3690 ± 0.0488	0.4603 ± 0.0880	0.2950 ± 0.0579
STRATA (ours)	1.5560 ± 0.1752	1.2414 ± 0.1789	1.4281 ± 0.2137	1.8796 ± 0.2653	0.7232 ± 0.0442	0.8193 ± 0.0206	0.7866 ± 0.0389	0.7725 ± 0.0317

→ The representation is about the formulation, not the cell type, so it carries over

Summary: one pipeline, five papers

- What we deliver becomes the next transcriptome we measure



Part 1: Represent the cell

- **scGPCL**: the count table is already a network
- **scBFP**: propagate measurements before denoising

Part 2: Represent the tissue

- **Spotscape**: compare every spot with every other one, not only the ones next to it

Part 3: Design and deliver

- **GxEvo: the score steers the next candidate**
- **STRATA**: proportions and chemistry, read at once

References

- **Part 1: Representing the cell**
  - Deep single-cell RNA-seq data clustering with graph prototypical contrastive learning. J. Lee, S. Kim, D. Hyun, N. Lee, Y. Kim, C. Park. **Bioinformatics** 39(6), btad342, 2023
  - Single-cell RNA sequencing data imputation using bi-level feature propagation. J. Lee, S. Yun, Y. Kim, T. Chen, M. Kellis, C. Park. **Briefings in Bioinformatics** 25(3), bbae209, 2024
- **Part 2: Representing the tissue**
  - Global Context-aware Representation Learning for Spatially Resolved Transcriptomics. Y. Oh, J. Lee, Y. Kim, S. Seo, N. Lee, C. Park. **ICML 2025**
- **Part 3: Designing and delivering**
  - GxEvo: Evolutionary Search for Transcriptome-Conditioned Molecular Design. Under review
  - Screening Lipid Nanoparticles through Structure-Ratio Alignment. Y. Lee, Y. Oh, H. Choi, C. Park. **NeurIPS 2026**

KAIST

Thank you

cy.park@kaist.ac.kr dsail.kaist.ac.kr

DSAIL Data Science & Artificial Intelligence