Querying Counterfactuals on Tissue Graphs with Supervised Disentanglement

Abdul Moeed[1,2] **Stefan Schrod**[1,3] **Martin Rohbeck**[1] **Marc Jan Bonder**[4,5]
Pavlo Lutsik[6] **Oliver Stegle**[1,3†] **Daniel Dimitrov**[1,3†]

[1]DKFZ, Heidelberg [2]HIDSS4Health [3]EMBL, Heidelberg [4]UMCG, Groningen
[5]Oncode Institute, Utrecht [6]KU Leuven

Abstract

Tissue graph counterfactuals ask what a cell would express if its spatial neighborhood were altered, a question central to predicting cell behavior in tissues. Existing methods, however, present only specific instances of such queries. In this work, we first formalize *tissue graph counterfactuals* as a class of spatial interventions that either rewire connections between cells (*edge perturbation*) or modify the expression of their neighbors (*node perturbation*). We then introduce *Cellina* [‡], a framework that uses supervised disentanglement to decompose a cell's intrinsic state from its spatial context, using the latter as a conditioning input for in-silico perturbation predictions. Across benchmarks spanning over 2.5 million spatially-resolved cells in observational and interventional screens, *Cellina* outperforms both spatially-informed and non-spatial competitors in in-silico perturbations, while training as fast as non-spatial methods. Moreover, we show that *Cellina* reveals cancer subdomains with distinct signaling programs in an unsupervised manner and enables targeted neighbor perturbation simulations.

1 Introduction

A central goal of computational single-cell biology is to predict how cells respond to perturbations, including in cell types or other contexts that have not been measured directly [Bunne et al., 2024, Roohani et al., 2025, Dimitrov et al., 2026]. Existing single-cell perturbation methods typically rely on at least one of two assumptions: (i) that perturbations act as shared stimuli applied uniformly across cells, and (ii) that cells are conditionally independent, giving rise to an effectively i.i.d. learning problem. Tissues violate both assumptions. Within a tissue, a cell's transcriptional state is shaped by its local neighborhood, that is, the composition of nearby cells and the signals they emit [Armingol et al., 2021]. Consequently, modeling tissues requires methods that reason about neighbor-driven stimuli, which are unique to every cell. This motivates two complementary prediction tasks: what would a cell express if it were surrounded by a different set of neighbors, and what would it express if its existing neighbors changed the activity of specific genes or pathways? To address these tasks, a first generation of spatially-informed perturbation models has recently emerged, ranging from label-swapping flow models [Akbarnejad et al., 2025] and coordinate-aware covariate models [Lin et al., 2025] to neighbor-propagating graph neural networks [Sun et al., 2025]. Yet, each method targets a specific intervention type and is commonly evaluated in-domain. Here, we formalize *tissue graph counterfactuals*: interventions on either the neighborhood connections of a focal cell (*edge perturbation*) or the expression of its neighbors (*node perturbation*), corresponding to the two mutable components of the tissue graph (see Section 3.3). Each operator corresponds to a distinct class of biological intervention. An edge perturbation asks about composition and location: the infiltration of an immune population into a tumor, the depletion of a cell type that sustains its growth, or, prospectively, how an administered therapeutic cell would behave once it lodges in one region

[†]Joint senior and corresponding authors: daniel.dimitrov@embl.de, o.stegle@dkfz-heidelberg.de
[‡]https://cellina.readthedocs.io

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

rather than another. A node perturbation asks about signaling in a fixed tissue, as when an agonist, an inhibitor, or a knockdown shifts a coherent gene program across a cell's existing neighbors. We evaluate an instance of each: transplanting cells between perturbed and healthy regions, and shifting pathway activity across their neighbors.

We present *Cellina* (Figure 1), a (graph) variational autoencoder (VAE) that addresses tissue graph counterfactuals by decomposing each cell's gene expression into an intrinsic representation z encoding cell identity and an extrinsic (spatial) representation s encoding the effect of its microenvironment. Because such disentangled representations cannot be reliably recovered without supervision [Locatello et al., 2019], we use cell-type and spatial-domain labels as a soft inductive bias without formal identifiability guarantees [Khemakhem et al., 2020], anchoring z to cell-type identity and adversarially removing spatial-domain information, which routes microenvironmental variation to s. We validate this separation under out-of-distribution regimes via in-silico neighborhood alterations, tasks at which a model that conflates the drivers of variation cannot succeed [Schölkopf et al., 2021].

Contributions:

1. We formalize *tissue graph counterfactuals* as a class of spatial interventions with two operators, edge and node perturbations, and use them to jointly define and evaluate neighborhood-driven cell responses.
2. We introduce *Cellina*, a dual-encoder graph VAE with supervised disentanglement. On in-silico perturbations derived from observational data, it outperforms spatial and non-spatial baselines on colorectal cancer ($+0.18$ Pearson over the closest baseline) and across held-out mouse brain domains. On *in vivo* spatial interventional data, where it predicts how T cells respond to specific CRISPR knockouts in neighboring cancer cells, it leads the closest competitor by $+0.17$ ($+0.24$ with *Cellina-GAT*) Pearson.
3. We use *Cellina*'s spatial latents to identify biologically distinct cancer subdomains without supervision, and to simulate pathway-targeted neighbor perturbations using existing priors.

2 Related Work

Perturbations and context transfer. scGen [Lotfollahi et al., 2019] and CPA [Hetzel et al., 2022, Lotfollahi et al., 2023] are standard methods for predicting cellular responses to perturbations. Both models assume i.i.d. data and neither represents continuous neighbor composition or cell-specific spatial contexts. More recent methods based on optimal transport [Bunne et al., 2023] and flow matching [Klein et al., 2025] model individual cell trajectories, yet still work on the same shared-stimulus intervention assumption. Extending this paradigm to tissue perturbations requires disentangling intrinsic cellular state from extrinsic influence, and reasoning about continuous variation in individual neighborhoods rather than shared or discretized stimuli.

Spatially-informed disentanglement. A related line of work leverages spatial information to separate intrinsic cell states from extrinsic tissue influences. For example, MISTy [Tanevski et al., 2022] and NCEM [Fischer et al., 2023] model neighborhood effects through multi-view regression and graph neural networks, respectively, while SIMVI [Dong et al., 2025] uses a graph VAE with unsupervised disentanglement to isolate spatially-induced variation. These approaches yield interpretable decompositions of spatial scales, but do not support in-silico perturbations.

Tissue graph perturbations. The state-of-the-art methods most directly related to simulating tissue graph counterfactuals in spatial omics are MintFlow [Akbarnejad et al., 2025], Concert [Lin et al., 2025], Celcomen [Megas et al., 2025], and SpatialProp [Sun et al., 2025]. Celcomen models spatial in-silico perturbations through learned gene-gene interactions, but learns a global interaction matrix shared across the tissue, perturbing gene values rather than nodes or edges. SpatialProp proposed modeling the downstream effects of neighbor perturbations, making it directly related to our *node perturbation* task. MintFlow and Concert both perform in-silico perturbations via label conditioning, but MintFlow approaches it via an operation on the graph itself, enabling adaptation to our *edge perturbation* task. *Cellina* supports both edge and node perturbations, disentangles intrinsic from spatial factors under label supervision, and focuses on predicting cell type $\times$ domain pairs unseen during training (Table A3).

Graph counterfactuals. The broader graph ML literature offers a complementary perspective: counterfactual reasoning over graphs has been explored via instance-level adjacency perturbations

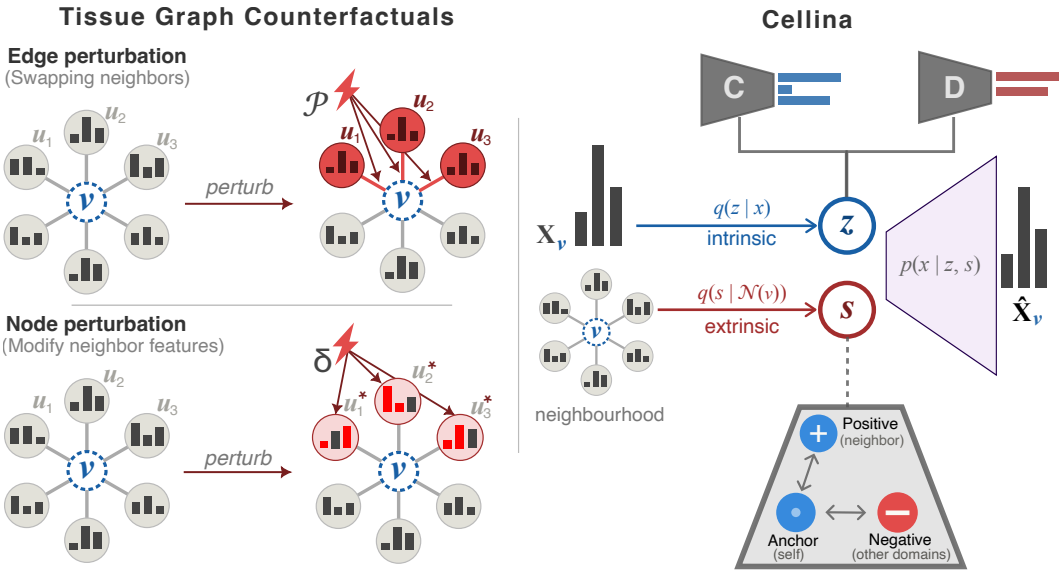

Figure 1: **Tissue graph counterfactuals and *Cellina* overview.** (*Left*) Two interventions on a focal cell v with neighbors u: **edge perturbation** rewires v's neighborhood $\mathcal{N}(v)$ to a counterfactual neighbor pool $\mathcal{P}$, and **node perturbation** alters neighbor expression on a feature (gene) subset S. (*Right*) *Cellina* encodes intrinsic identity $z \sim q(z \mid x_v)$ and spatial representation s from v's local neighborhood, and decodes $p(x \mid z, s)$. Supervision anchors z to cell type and removes spatial-domain information adversarially; *Cellina-GAT* additionally applies a contrastive loss on s.

for explainability [Lucic et al., 2022, Bajaj et al., 2021], and in generative graph VAEs with input-conditioned priors [Ma et al., 2022]. *Cellina* draws on this line of work but reorients it from model explanation to biological intervention, replacing label-swap objectives with simulating counterfactual neighborhood queries over tissue graphs. See Appendix F for full descriptions of competitors.

3 Method

3.1 Notation and Problem Definition

Let $\mathcal{V} = \{v_1, \ldots, v_N\}$ denote N cells on a spatial tissue slide. Each cell $v \in \mathcal{V}$ is associated with:

- $x_v \in \mathbb{Z}_{\geq 0}^G$: gene expression counts across G genes
- $y_v \in \{1, \ldots, C\}$: cell-type label (e.g., T cell, epithelial, fibroblast)
- $d_v \in \{1, \ldots, D\}$: spatial domain label – a discrete partition of a tissue according to pre-defined regions or niches (e.g., tumor vs. healthy regions)

Spatial domain and cell-type labels capture complementary aspects of a cell's identity: cell-type label y_v encodes intrinsic cell identity (*what* kind of cell it is), while spatial domain d_v encodes which tissue region it inhabits (*where* it is located).

The spatial proximity between cells is encoded by a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ whose nodes $\mathcal{V}$ are cells and whose edges $\{u, v\} \in \mathcal{E}$ connect a focal cell v to its neighbors u, with edge weights $W_{uv} \geq 0$. W is computed from the two-dimensional spatial coordinates of each cell v relative to all others using a Gaussian proximity kernel (Appendix B). For each cell v, we denote the spatial neighborhood as:

$$\mathcal{N}(v) = \{u \in \mathcal{V} \mid \{v, u\} \in \mathcal{E}\}$$

On this graph we formalize two intervention operations: *edge* and *node* perturbations (Section 3.3). Because matched pre- and post-intervention measurements of the same cells are generally unavailable *in situ* [Bunne et al., 2023, Dimitrov et al., 2026], we evaluate them as *in-silico perturbations*, mirroring existing single-cell context transfer benchmarks [Roohani et al., 2025, Wu et al., 2024, Szałata et al., 2024]; throughout, we report *in-silico perturbation predictions* for empirical results and reserve *tissue graph counterfactual* for the formal task definition.

3.2 Model Architecture

Generative model. *Cellina* is a (graph) variational autoencoder (VAE) [Kingma and Welling, 2013, Kipf and Welling, 2016] that decomposes each cell's expression into two latent variables: an intrinsic representation z encoding cell identity and a spatial representation s encoding microenvironmental influence. Both have standard normal priors, Normal$(0, I)$. Following standard practice for single-cell counts [Lopez et al., 2018, Gayoso et al., 2022], the likelihood $p(x \mid z, s)$ is a negative binomial distribution parametrized by a decoder $\mathrm{Dec}([z;\, s])$ ($[\cdot\,;\,\cdot]$ denotes concatenation; Appendix D.1).

Inference. We propose two variants of the model. Both use an MLP encoder $\mathrm{Enc}_z(x_v)$ for the intrinsic posterior $q(z \mid x_v) = \mathrm{Normal}(\mu_z, \sigma_z^2)$, and differ in how the spatial posterior is parametrized:

- **Cellina** uses a degree-normalized aggregation of neighbor expression $\varphi(v) = \big(\sum_{u\in\mathcal{N}(v)} W_{uv}\,\tilde{x}_u\big)\big/\big(\sum_{u\in\mathcal{N}(v)} W_{uv}\big)$ where $\tilde{x}_u \in \mathbb{R}^G$ denotes log-normalized x_u; encoded via an MLP encoder $\mathrm{Enc}_s(\varphi)$ that outputs μ_s, σ_s.
- **Cellina-GAT** replaces the fixed aggregator with a multi-layer GATv2 [Brody et al., 2021] operating on v's local subgraph $\mathcal{G}_v = (x_v,\, \{x_u\}_{u\in\mathcal{N}(v)},\, \mathcal{E}_v, W_{uv})$ (where $\mathcal{E}_v \subseteq \mathcal{E}$ is an edge set of $\mathcal{N}(v)$); self-loops are excluded, so s does not directly aggregate v's own expression, which is primarily captured by z; μ_s, σ_s are linear heads on the focal-node representation. For this variant, we add a modified contrastive loss $\mathcal{L}_{\mathrm{spatial}}$ (Appendix D).

These two variants trade off efficiency and expressivity: *Cellina*'s linear aggregator decouples neighborhood construction from training and exhibits training-time scaling similar to non-spatial baselines (Figure A9), whereas *Cellina-GAT* learns attention over each subgraph at additional computational cost per step (Appendix D.2). All approximate posteriors are diagonal Gaussians whose parameters are output by the encoders, and samples are drawn using the reparameterization trick [Kingma and Welling, 2013].

Supervised disentanglement. Because optimizing the ELBO of the VAE alone does not prevent z from absorbing spatially-driven variation, we introduce two auxiliary objectives that route spatial signal to s by removing it from z. Specifically, (i) we anchor z to cell-type identity using a cell type classifier with loss $\mathcal{L}_{\mathrm{clf}}$, and (ii) adversarially strip spatial-domain information from z through a two-part adversarial objective. First, $\mathcal{L}_{\mathrm{disc}}$ is optimized to train the discriminator to predict the domain label from z; second, $\mathcal{L}_{\mathrm{adv}}$ is optimized to encourage the encoder to render z domain-invariant. For *Cellina-GAT*, we further add a modified graph-supervised contrastive loss $\mathcal{L}_{\mathrm{spatial}}$ on s as a biologically grounded inductive bias. We use a cell's spatial neighbors as positives, rather than cells sharing a domain label, to discourage s from collapsing onto domain labels and erasing within-domain neighborhood variation; domain labels enter only through the negatives, which separate distinct domains. Appendix E ablates each auxiliary objective independently, probing the cell-type and domain information retained in the frozen latents; the contrastive term in particular improves latent informativeness and fit quality. The combined training objective (minimized in step 2 over encoder and decoder parameters) follows as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{VAE}} + \alpha_{\mathrm{clf}}\,\mathcal{L}_{\mathrm{clf}} + \alpha_{\mathrm{spatial}}\,\mathcal{L}_{\mathrm{spatial}} - \alpha_{\mathrm{adv}}\,\mathcal{L}_{\mathrm{adv}},$$

where α are data-adaptive normalization scales fixed after the first training epoch (Appendix D.4).

Training procedure. We optimize this objective via alternating updates: (1) a discriminator step that updates the discriminator on detached z (encoder parameters frozen), and (2) a VAE step that updates encoder and decoder with the discriminator frozen, using the combined objective above. Alternating updates ensure the discriminator provides informative gradients to the encoder [Goodfellow et al., 2014]; for details and hyperparameters, see Appendices C and D.

3.3 Tissue Graph Counterfactuals

A tissue graph counterfactual asks: *what would cell v express if its spatial neighborhood were altered while its intrinsic identity remained fixed?* That alteration acts on the graph itself (the cell's microenvironment), not on an intrinsic per-cell covariate. The tissue graph $\mathcal{G}$ has two mutable components: its edges $\mathcal{E}$ with associated weights W, encoding the neighborhood topology, and the neighbor node feature matrix X, encoding cell expression. This naturally gives rise to two counterfactual queries:

Definition 1 (Edge Perturbation). An edge perturbation intervenes on a cell's local neighborhood $\mathcal{N}(v)$, replacing it with an alternative neighborhood $\mathcal{N}'$:

$$\mathcal{N}(v) := \mathcal{N}'.$$

This is an intervention on the topology alone: it admits arbitrary modifications to the neighborhood, including the addition, removal, or substitution of neighbors, with replacements drawn from cells typically observed in the data, without altering the expression of neighboring cells, i.e., the neighbor node features X. Biologically, it asks about composition and location: the infiltration of a cell population into a region, the depletion of one that sustains it, or the arrival of a cell type that was previously absent. In this work, we evaluate **edge perturbation** as an in-silico domain edge rewiring: focal cells of type y in a source domain are assigned new cell-type-matched neighborhoods from the target domain; type y itself is excluded from the donor pool and from W in training and inference, so that none of its target domain expression enters the model (Appendix B.6).

Definition 2 (Node Perturbation). A node perturbation modifies the feature vectors of v's neighbors while preserving the graph topology. Let $\mathcal{S}$ denote the set of intervened genes. For each neighbor $u \in \mathcal{N}(v)$, the perturbed gene feature vector is defined as:

$$x_{u,g}^{\mathrm{cf}} = \begin{cases} T_g(x_{u,g}) & g \in \mathcal{S} \\ x_{u,g} & g \notin \mathcal{S} \end{cases}$$

where $T_g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a gene-specific transformation encoding the counterfactual expression under the intervention. The approach is agnostic to the choice of T_g, which may correspond to additive shifts, knockouts, overexpression, or learned counterfactual values. In our setting, we instantiate T_g as an in-silico perturbation, specifically a multiplicative scaling $T_g(x_{u,g}) = x_{u,g} \cdot e^{\delta_g}$, where $\delta_g \in \mathbb{R}$ is a gene-specific log fold-change applied to each cell x_u (approximately, an additive shift in log-normalized space; Appendix B.7). Biologically, it asks about signaling in a fixed tissue: an agonist, an inhibitor, or a knockdown that shifts a coherent program across a cell's existing neighbors. Operating on a subset $\mathcal{S}$ rather than the full feature vector follows from this: genes act in co-regulated programs, so $\mathcal{S}$ can be chosen to target a specific biological process such as a pathway or regulon.

In sum, an **edge perturbation** alters the topology of the tissue graph (which and how many cells are a focal cell's neighbors), but not neighbor expression. In contrast, a **node perturbation** alters the expression of any number of genes but not the neighbor topology; Appendix O details the operations each perturbation admits, their applications (Table A12), and the narrow case in which they coincide.

4 Experiments

4.1 Evaluation Setting

Interventional spatial data remain scarce, as current spatial perturbation screens are limited by their assays, which often provide limited cellular resolution or incomplete transcript readouts, and by their designs, which generally lack matched controls [Dhainaut et al., 2022, Breinig et al., 2026, Binan et al., 2025]. We therefore evaluate the models under held-out prediction tasks. During training, cells of type y are observed in a source condition d but held out in a target condition d', where they are also excluded from the neighborhood graph W at training and inference. At test time, source cells of type y are assigned the neighborhood they would have in d', either neighbors drawn from d' (edge perturbation) or their own neighbors with expression shifted towards d' (node perturbation; Section 3.3), and the predicted expression is compared with that of the held-out cells of type y in d'. Since assays are destructive, this reference is an observed population rather than the same cells. This mirrors, in a spatial setting, the *context transfer* protocol established for perturbation benchmarks [Szałata et al., 2024, Wu et al., 2024, Roohani et al., 2025], in which a perturbation's effect is observed in some cell types and must be predicted for a cell type in which it was never observed; here, the perturbation is the change of neighborhood from d to d'. In the CRC and mouse brain cohorts, d and d' correspond to spatial domains (e.g., healthy colon and pathologist-annotated cancer areas; Figure 2). In Perturb-FISH interventional data, we predict how control T cells respond to tumor cell neighborhoods d' defined by distinct *in situ* CRISPR knockouts (Table 3).

This approach allows for assessing both types of graph perturbation defined in Section 3.3. In *edge perturbation*, the neighbors of seed cells from the source domain are swapped with cells from the target domain; in *node perturbation*, seed cells retain their original neighbors; their neighbors' feature

vectors are then altered using a cell-type-specific factor $\delta_{y,g}$; for the held-out type this is the global shift $\delta_{\setminus y}$, so that the perturbed input does not provide the held-out population's shift (Appendix B.7).

4.2 Benchmarking Approach

Metrics. Following recent benchmarking practices [Wu et al., 2024, Roohani et al., 2025, Heidari et al., 2026], we evaluate in-silico perturbation predictions along four complementary axes: (i) whether methods recover the correct direction and relative magnitude of gene-level effects (Pearson r on log fold-changes of top differentially expressed genes), (ii) whether they correctly identify the most differentially expressed genes (Signed Precision), (iii) whether predicted and observed expression shifts agree in absolute magnitude (RMSE of log fold-change vectors), and (iv) overall distributional fit (E-distance). All metrics are assessed for each held-out cell type; formal definitions and an expanded metrics set with per-cell-type breakdown appear in Appendices A and G.

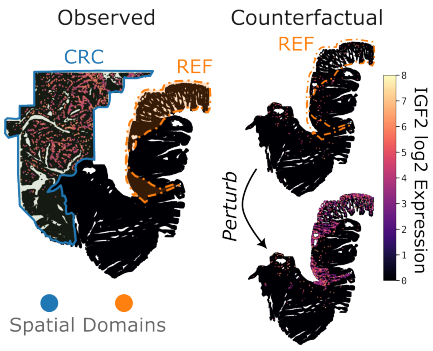

Figure 2: **In-silico expression prediction with *Cellina*.** Observed IGF2 expression in Fibroblasts with pre-annotated spatial domains. Rewiring $\text{Fibroblasts}_{\text{REF}}$ neighborhoods to those of $\text{Fibroblasts}_{\text{CRC}}$ yields predicted IGF2 expression consistent with observed CRC Fibroblasts.

Alternative methods. We consider the following methods, spanning spatially-informed models with a native neighborhood-perturbation interface and popular non-spatial baselines (see Appendix F for full descriptions; all methods, including *Cellina*, are run at their default settings, Appendix C): *Mean shift*, an average expression shift between domains, applied to each cell type, and motivated by similar baselines [Ahlmann-Eltze et al., 2025, Viñas Torné et al., 2026]; *scGen, CPA*, spatially-uninformed perturbation-focused autoencoders, which treat the spatial domain as a label embedding or shift [Lotfollahi et al., 2019, 2023]; *Concert*, a spatially-informed model that propagates perturbations across cells through a Gaussian process over their coordinates, but simulates them by swapping categorical attribute embeddings rather than through graph operations [Lin et al., 2025]; *SpatialProp*, an efficient GNN-based method tailored for in-domain prediction of neighbor perturbations [Sun et al., 2025]; and *MintFlow*, a spatially-informed flow-matching model for in-domain tissue perturbations via neighbor cell-type label swapping, adapted to our edge perturbation task [Akbarnejad et al., 2025]. MintFlow is trained on all cells, including held-out cell types. SIMVI [Dong et al., 2025] was excluded from our in-silico perturbation evaluations due to a lack of native support for such queries and prohibitive GPU memory requirements (Appendix N, Figure A9); a disentanglement comparison with it favors *Cellina* (Appendix M).

We evaluate three variants of our model (Section 3): (1) *Cellina*, the base model with pre-aggregated neighborhoods and both supervision objectives on z; (2) *Cellina (ablated)*, which removes both supervision objectives and retains only the dual-encoder architecture (per-component and architecture ablations in Appendix E); and (3) *Cellina-GAT*, which replaces the pre-aggregated encoder with a graph attention network and a contrastive loss on s.

4.3 Results

Cellina accurately predicts in-silico perturbations in colorectal cancer data.

To evaluate in-silico perturbation predictions in a clinically relevant setting, we consider a spatial transcriptomics dataset of approximately 2.4 million cells across six colorectal cancer tissue slides (an 18,000-gene panel, see Appendix B) [Crowell et al., 2025]. We use pathologist-annotated healthy colonic mucosa (REF) and colorectal cancer (CRC) regions to define the prediction task: each REF cell of type y receives new neighbors sampled from CRC cells that neighbor type y, and the predicted expression is evaluated against held-out CRC cells of the same type.

Both supervised *Cellina* variants outperform every competitor on Pearson, Signed Precision and RMSE_{LFC}. Even *Cellina (ablated)*, trained without cell-type or domain supervision, leads or matches all non-*Cellina* methods on correlation and precision, a result we attribute to encoding continuous, cell-

resolved neighborhoods. Supervised disentanglement adds a further $+0.15$ Pearson in the full model (Table 1; component ablations in Appendix E). Notably, none of the spatially-informed competitors surpasses the non-spatial CPA on Pearson or RMSE. The mean shift baseline, which applies a group-average expression change without knowledge of individual neighborhoods, is competitive with scGen and spatially-informed Concert on Pearson r. scGen, nevertheless, attains the lowest E-distance (4.65), followed by *Cellina (ablated)*, but this metric is computed in the log-normalized PCA space that scGen is trained to reconstruct; when scored in count space by negative-binomial deviance, scGen falls behind *Cellina* and CPA (Table A5). We therefore prioritize Pearson r and Signed Precision, as they capture which genes respond to a perturbation, i.e. the biologically relevant signal that would guide follow-up experiments.

We also confirm that these results are robust to random model initialization across seeds (Appendix I). Moreover, when an entire patient is held out from training together with the target cell type, *Cellina* widens its gain over CPA from $+0.18$ to $+0.27$ Pearson, leading on all four metrics (Appendix J), indicating that it generalizes to unseen donors.

Table 1: Leave-one-cell-type-out performance (top 50 DEGs) for predicting the in-silico perturbed state of healthy colon cells placed in tumor region (REF $\rightarrow$ CRC). Mean $\pm$ std across cell types and patient samples. **node-pert** refers to node perturbation task. Best per metric within each block (edge- vs. node-perturbation) in **bold**; *Cellina* variants in gray, spatially-informed competitors in orange.

Method	Pearson ↑	$\text{Precision}_{\text{signed}}$ ↑	E-distance ↓	RMSE_{LFC} ↓
Mean shift	0.51 ± 0.26	0.18 ± 0.15	29.89 ± 10.49	5.08 ± 2.51
CPA	0.68 ± 0.19	0.22 ± 0.19	6.44 ± 2.27	1.50 ± 0.56
scGen	0.50 ± 0.37	0.19 ± 0.20	$\mathbf{4.65 \pm 3.49}$	1.99 ± 0.84
Concert	0.49 ± 0.43	0.21 ± 0.22	29.98 ± 5.22	1.81 ± 0.67
MintFlow	0.65 ± 0.31	0.23 ± 0.21	13.01 ± 1.76	1.51 ± 0.43
Cellina (ablated)	0.71 ± 0.24	0.31 ± 0.22	4.93 ± 1.66	1.51 ± 0.88
Cellina	0.86 ± 0.17	$\mathbf{0.45 \pm 0.19}$	7.00 ± 1.05	1.14 ± 0.56
Cellina-GAT	$\mathbf{0.89 \pm 0.09}$	0.42 ± 0.20	8.59 ± 1.90	$\mathbf{1.09 \pm 0.61}$
$\text{Cellina}_{\text{node-pert}}$	0.80 ± 0.18	0.36 ± 0.19	$\mathbf{7.20 \pm 1.45}$	1.39 ± 0.76
$\text{Cellina-GAT}_{\text{node-pert}}$	$\mathbf{0.88 \pm 0.09}$	$\mathbf{0.40 \pm 0.18}$	11.35 ± 3.26	$\mathbf{1.08 \pm 0.49}$
$\text{SpatialProp}_{\text{node-pert}}$	0.46 ± 0.29	0.06 ± 0.11	27.12 ± 8.51	7.39 ± 1.23

Node perturbation evaluation. Next, we evaluate *Cellina* in the same (REF $\rightarrow$ CRC) regime, but under a node-perturbation setting: rather than changing the neighborhood context entirely, we retain each focal cell's original neighbors and shift their expression by cell-type-specific log-fold vectors $\delta_{y,g}$ ($\delta_{\setminus y}$ for the held-out type; top 200 genes) that reflect the expression difference between REF and CRC (Appendix B.7). By design, SpatialProp is the direct competitor in this evaluation; *Cellina* leads it on Pearson, Signed Precision, E-distance, RMSE and NB deviance (Table A5). For *Cellina-GAT*, **edge** and **node perturbation** are on par on correlation and precision, while base *Cellina*'s node perturbation trails its edge counterpart (CRC 0.80 vs. 0.86 Pearson). We also see that node perturbation peaks at $\approx$100–200 perturbed genes and degrades beyond (Appendix B.7). Node perturbations are, however, more targeted, enabling scenarios where only specific gene programs in the microenvironment are modified, which we explore in Section 4.4.

Cellina performs competitively across domains in the mouse brain.

To further assess multi-domain generalization, we repeat the same evaluation on a whole-adult-mouse MERFISH cohort [Zhang et al., 2023] (1,122-gene panel, ~146K cells, three slides), holding out two independently annotated spatial domains (Fiber-tracts and Isocortex). Averaged over held-out domains, *Cellina-GAT* leads all baselines on Pearson (0.84, tied with *Cellina*), Signed Precision (0.53) and RMSE_{LFC} (5.97), with CPA as its closest competitor (0.82, 0.44, 6.36) and scGen with the lowest E-distance (5.73; Table 2). Concert and MintFlow are below CPA on all four metrics, and the node-perturbation ranking likewise holds, with *Cellina-GAT* leading SpatialProp by a safe margin.

Table 2: Leave-one-cell-type-out performance (top 50 DEGs) for the in-silico perturbations Thalamus $\rightarrow$ Isocortex, Fiber-tracts. Mean $\pm$ std across cell types and three slides, averaged over both held-out domains; per-domain results in Appendix Table A4. Best per metric within each block (edge- vs. node-perturbation) in **bold**; *Cellina* variants in gray, spatially-informed competitors in orange.

Method	Pearson $\uparrow$	$\text{Precision}_{\text{signed}}$ $\uparrow$	E-distance $\downarrow$	RMSE_{LFC} $\downarrow$
Mean shift	0.43 ± 0.23	0.12 ± 0.09	25.18 ± 4.75	10.98 ± 2.78
CPA	0.82 ± 0.16	0.44 ± 0.13	6.53 ± 1.99	6.36 ± 4.71
scGen	0.77 ± 0.16	0.20 ± 0.14	$\mathbf{5.73 \pm 4.84}$	6.48 ± 3.97
Concert	0.77 ± 0.16	0.18 ± 0.15	24.91 ± 2.08	7.45 ± 4.99
MintFlow	0.81 ± 0.17	0.29 ± 0.16	19.52 ± 1.81	7.27 ± 5.43
Cellina (ablated)	0.81 ± 0.15	0.37 ± 0.15	8.33 ± 4.76	6.78 ± 4.54
Cellina	$\mathbf{0.84 \pm 0.16}$	0.51 ± 0.17	6.93 ± 0.83	6.08 ± 5.06
Cellina-GAT	$\mathbf{0.84 \pm 0.16}$	$\mathbf{0.53 \pm 0.17}$	7.67 ± 0.83	$\mathbf{5.97 \pm 5.07}$
$\text{Cellina}_{\text{node-pert}}$	0.82 ± 0.15	0.42 ± 0.15	8.96 ± 2.92	6.65 ± 4.63
$\text{Cellina-GAT}_{\text{node-pert}}$	$\mathbf{0.83 \pm 0.16}$	$\mathbf{0.51 \pm 0.16}$	$\mathbf{8.61 \pm 1.15}$	$\mathbf{5.98 \pm 4.98}$
$\text{SpatialProp}_{\text{node-pert}}$	0.66 ± 0.14	0.04 ± 0.04	30.79 ± 5.18	8.18 ± 2.05

Cellina predicts *in vivo* spatial neighbor perturbation effects.

Next, we evaluate *Cellina* using *in vivo* interventional Perturb-FISH data from a human melanoma xenograft [Binan et al., 2025]. The data measure CRISPR knockouts (KOs) of NFκB-pathway genes in cancer cells and infiltrating T cells. The task is to predict how control T cells (far away from perturbed cancer cells) respond if their neighbors expressed the profile of perturbed cancer cells. During training we hold out T cells that have perturbed cancer cells as neighbors. At inference, we set the expression of cells neighboring the control T cells to the per-gene mean of the KO-specific cancer population and compare the predicted expression against held-out T cells (Appendix B).

Predicting the full T cell response is dominated by a shared transcriptional shift that varies little across perturbations, which even the mean shift captures (0.51 vs. 0.55 Pearson for *Cellina-GAT*; Appendix Table A6); we, therefore, isolate the KO-specific component by removing the average perturbation shift from the predicted and observed logFC [Viñas Torné et al., 2026]. On this task, *Cellina* variants lead SpatialProp, and a *Cellina* variant is best on every metric except E-distance (*Cellina-GAT* on Pearson, 0.40, and RMSE_{LFC}; base *Cellina* on Signed Precision, 0.68; Table 3). The mean shift has the lowest E-distance yet no KO-specific signal (Pearson 0.06). To further verify that the perturbation response is due to the knockout effects, rather than a shared effect of having cancer neighbors, we perform an additional experiment in which the neighbors instead express the profile of randomly drawn cancer cells. All methods drop in performance under this random control, and *Cellina*'s KO-specific Pearson falls to zero ($0.33 \rightarrow 0.00$; Appendix Table A6), confirming that its predictions are knockout-specific.

Table 3: Node perturbation performance for the Perturb-FISH *in vivo* CRISPR KO dataset. KO-specific effect; mean over 9 KOs, $\pm$ std over 5 seeds. A *Cellina* variant (gray) is best on Pearson, Signed Precision and RMSE_{LFC}. Spatial competitors in orange.

Method	Pearson $\uparrow$	$\text{Precision}_{\text{signed}}$ $\uparrow$	E-distance $\downarrow$	RMSE_{LFC} $\downarrow$
Mean shift	0.06 ± 0.06	0.51 ± 0.03	$\mathbf{0.13 \pm 0.01}$	0.46 ± 0.02
Cellina	0.33 ± 0.08	$\mathbf{0.68 \pm 0.03}$	4.13 ± 0.15	0.43 ± 0.01
Cellina-GAT	$\mathbf{0.40 \pm 0.07}$	0.66 ± 0.03	4.89 ± 0.19	$\mathbf{0.41 \pm 0.01}$
SpatialProp	0.16 ± 0.06	0.55 ± 0.06	7.44 ± 0.04	0.64 ± 0.04

4.4 *Cellina* Captures Biologically Meaningful Within-Domain Heterogeneity

To test whether *Cellina*'s spatial latent s captures microenvironmental variation finer than the domain labels used during training, we ask whether it separates cells within a single annotated domain, i.e., the tumor region of the colorectal cancer cohort [Crowell et al., 2025]. We cluster the latent s over CRC cells to discover spatially autocorrelated subdomains [DeTomaso and Yosef, 2021], which recovers three modules, of which we analyze the two with the most distinct pathway profiles

(denoted CRC1 and CRC2; Appendix H). In UMAP space, reconstructed Fibroblast counts from CRC1 and CRC2 form clearly separated clusters, and in-silico perturbed Fibroblasts (REF→CRC1/2) integrate well with each respective subdomain (Figure 3(a)), suggesting that s effectively captures local variation.

***Cellina*'s predictions are specific to the local neighborhood.** To support that s captures context-specific variation, first, we compare predictions conditioned on neighborhoods sampled from the matched subdomain against those conditioned on the global CRC domain, scored against the same observed reference, and find the former better on average across the five cell types (Figure 3(b)). Further, when we rewire every neighbor to a single cell type (Table A9), we see that identical neighbors yield different responses across focal cell types, and that predictions do not merely propagate neighbor expression, since a neighborhood of the held-out type itself is rarely the best case. We also observe that edge-perturbation predictions depend on the neighborhood size in a cell-type-specific manner and degrade once the neighborhoods span a large part of the slide (Table A10). Similarly, under node perturbation, the agreement with the observed response increases with the fraction of perturbed neighbors (Table A11).

Subdomains recover interpretable signaling programs. To interpret CRC1 and CRC2 biologically, we score their gene modules against PROGENy pathway gene sets [Schubert et al., 2018, Badia-i Mompel et al., 2022], recovering distinct signaling profiles (Figure 3(c)): TGFβ-dominant for CRC1 and NFκB/MAPK-dominant for CRC2. These signatures arise without direct supervision and are consistent with the signaling heterogeneity reported by Crowell et al. [2025].

Pathway-specific neighbor perturbations are consistent with biologically grounded responses. We next ask whether prior pathway knowledge alone suffices to reproduce these subdomain responses (Figure 3(d)). Encouragingly, performing node perturbations with PROGENy pathway weights as alteration vectors δ_g recovers the subdomain effects to a large extent (Appendix H; Pearson $r = 0.77$ (TGFβ→CRC1) and $r = 0.76$ (NFκB→CRC2)). Notably, the most up-regulated gene predicted under each pathway (FN1 for TGFβ, MMP3 for NFκB) is consistent with established fibroblast programs: TGFβ signaling is a canonical driver of extracellular-matrix production (FN1), while NFκB activation induces matrix-remodeling metalloproteinases (MMP3)—observed hallmarks of cancer-associated fibroblast and immune microenvironments [Crowell et al., 2025].

5 Discussion and Conclusion

Foundation models of single cells are scaling rapidly [Bunne et al., 2024], yet most still model cells as independent samples [Rood et al., 2024]. For such models to serve as virtual cells in tissue context, they must also predict how a cell's state changes with its neighborhood. Recent work has demonstrated compelling instances of modeling tissue perturbations: in-silico cell-type depletion and swapping via label manipulation, including regulatory T cell modulation relevant to cellular therapy [Akbarnejad et al., 2025]; combinatorial queries on perturbations and covariates in interventional spatial data [Lin et al., 2025]; and gradual microenvironment steering, paired with causal analyses linking perturbation predictions to downstream response [Sun et al., 2025]. We hope that our formalization of tissue graph counterfactuals can serve as a common language for such instances, and for building and evaluating models that predict how cells respond to altered neighborhoods. Although such perturbations remain in-silico hypotheses that require wet-lab validation, they offer a way to query virtual cell responses in spatial context and could help guide future experimental design.

Spatial neighborhoods are continuous, compositionally heterogeneous contexts, and averaging over coarse labels or shifts ignores the variation within them. Two design choices follow directly from this observation and underpin *Cellina*'s performance. First, encoding neighborhoods continuously yields a strong prior even without any supervision: *Cellina (ablated)*, trained without cell-type or domain labels, already matches all non-*Cellina* baselines on correlation and precision in colorectal cancer. Second, supervised disentanglement adds substantial further gains. *Cellina*'s explicit disentanglement additionally enables the discovery of the subdomain structure that can be used to refine pathology annotations, and simulate pathway-specific neighbor perturbations. Together, these components allow a *Cellina* variant to outperform non-spatial baselines on the logFC-based metrics in every dataset we evaluated, whereas none of the other spatially-informed methods consistently outperforms the strongest non-spatial baseline. *Cellina* itself is a relatively simple model, and the more expressive but

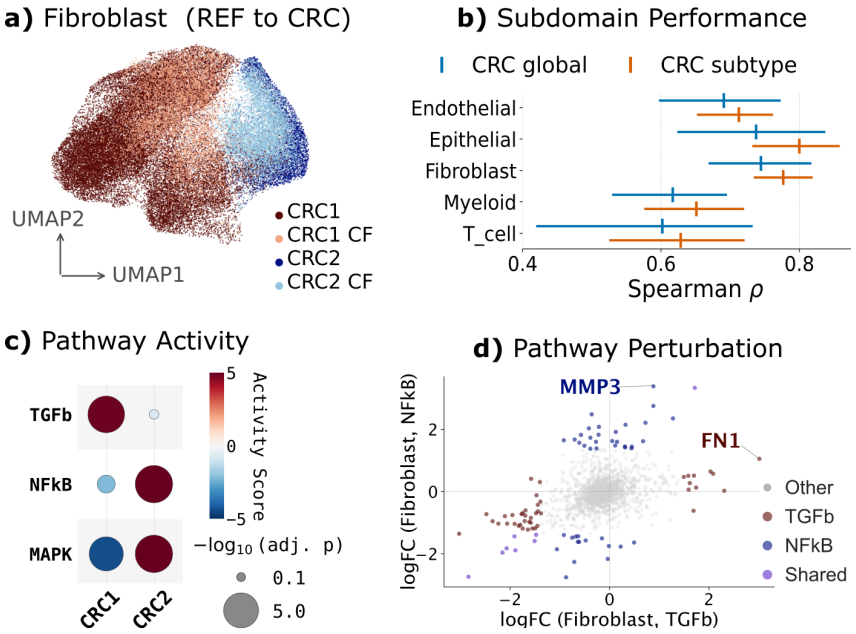

Figure 3: **Cellina identifies spatial subdomains and enables pathway-specific perturbations in CRC.** (**a**) UMAP of *Cellina*-generated Fibroblast counts; in-silico perturbed cells (CRC1 CF, CRC2 CF) integrate with their respective observed CRC subdomain populations. (**b**) Spearman ρ across cell types (mean, 95% CI), comparing global (pre-annotated) CRC neighborhoods versus *Cellina*-identified subdomains. (**c**) Inferred pathway activity per CRC subdomain. (**d**) Top-50 most affected genes under domain-matched pathway perturbations (x-axis: CRC1; y-axis: CRC2).

costlier *Cellina-GAT* leads on node perturbation, but performs largely on par in edge perturbations, indicating that the pre-aggregated neighborhoods already capture the relevant microenvironmental structure; this is consistent with a broader pattern in single-cell perturbation modeling, where simpler approaches often match more complex ones [Ahlmann-Eltze et al., 2025, Viñas Torné et al., 2026].

Limitations and future directions.

Cellina's supervised disentanglement objectives rely on cell-type and spatial-domain annotations, themselves simplifications of continuous biology. While the annotations are required only during training, and not used during inference, they can impact how well the spatial latent s reflects neighborhood variation. Furthermore, the underlying imaging-based spatial transcriptomics data also depend on cell segmentation, which remains error-prone: transcripts are frequently misassigned across cell boundaries and can dominate downstream niche and neighbor-influence analysis [Mitchel et al., 2026]. A natural extension is to account for segmentation uncertainty within *Cellina*, for instance by jointly modeling raw imaging and transcript-level data.

Beyond these data-related limitations, our predictions are not causal effect estimates, even where they are evaluated against interventional data. Instead, we use *counterfactual* operationally to denote simulated outcomes under altered neighborhoods, rather than in the strict Pearlian sense. Similarly, our evaluation is a context transfer: because spatial assays are destructive, predictions are scored against the observed target-domain population rather than against the same cells after an intervention. Further, *Cellina* does not impose the structural assumptions required for identifiability [Khemakhem et al., 2020, Lachapelle et al., 2022], though it could be extended with, e.g., sparse mechanism priors [Lachapelle et al., 2022, Lopez et al., 2023], as a path toward causal representation learning with formal guarantees [Schölkopf et al., 2021]. Moreover, although *Cellina* is a probabilistic model, we report only mean predictions and leave calibration of its predictive uncertainty to future work. Finally, emerging spatial perturbation screens [Dhainaut et al., 2022, Breinig et al., 2026, Binan et al., 2025], albeit limited in scale and resolution, increasingly provide spatially resolved interventional readouts that could ground these predictions empirically. Our task on spatially-resolved knockouts profiled *in vivo* is an early attempt at this, and cell-depletion experiments with single-cell spatial readouts would provide a natural interventional test for future depletion simulations.

Acknowledgments and Disclosure of Funding

The authors' work is supported through state funds approved by the State Parliament of Baden-Württemberg for the Innovation Campus Health + Life Science alliance Heidelberg Mannheim, the Data Science Collaborative Research Programme 2022 by the Novo Nordisk Foundation (grant NNF22OC0076414), the Priority Program Translational Oncology of the Deutsche Krebshilfe (grant 70115167), the Helmholtz Association under the joint research school "HIDSS4Health – Helmholtz Information and Data Science School for Health", the European Research Council (Synergy Grant DECODE 810296), and KU Leuven (BOFZAP starting grant STG/22/024 and C1 grant C14/24/151). We also thank Philipp Sven Lars Schaeffer, Zeynep Edizcan, Ahmet Rifaioglu, Elyas Heidari, Rama Abdulhamid, Ricardo Ramirez Flores, and Julio Saez-Rodriguez for their feedback. The authors declare that they have no competing interests or financial relationships outside the submitted work.

References

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22(8): 1657–1661, 2025.

Amir Akbarnejad, Lloyd Steele, Daniyal J Jafree, Sebastian Birk, Marta Rosa Sallese, Koen Rademaker, Adam Boxall, Benjamin Rumney, Catherine Tudor, Minal Patel, et al. Mapping and reprogramming human tissue microenvironments with mintflow. *bioRxiv*, 2025. doi: 10.1101/2025.06.24.661094.

Erick Armingol, Adam Officer, Olivier Harismendy, and Nathan E Lewis. Deciphering cell–cell interactions and communication from gene expression. *Nature Reviews Genetics*, 22(2):71–88, 2021.

Pau Badia-i Mompel, Jesús Vélez Santiago, Jana Braunger, Celina Geiss, Daniel Dimitrov, Sophia Müller-Dott, Petr Taus, Aurelien Dugourd, Christian H Holland, Ricardo O Ramirez Flores, et al. decoupler: ensemble of computational methods to infer biological activities from omics data. *Bioinformatics Advances*, 2(1):vbac016, 2022.

Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust counterfactual explanations on graph neural networks. *Advances in Neural Information Processing Systems*, 34:5644–5655, 2021.

Loïc Binan, Aiping Jiang, Serwah A. Danquah, Vera Valakh, Brooke Simonton, Jon Bezney, Robert T. Manguso, Kathleen B. Yates, Ralda Nehme, Brian Cleary, and Samouil L. Farhi. Simultaneous CRISPR screening and spatial transcriptomics reveal intracellular, intercellular, and functional transcriptional circuits. *Cell*, 188(8):2141–2158.e18, April 2025. ISSN 1097-4172. doi: 10.1016/j.cell.2025.02.012.

Marco Breinig, Artem Lomakin, Elyas Heidari, Michael Ritter, Gleb Rukhovich, Lio Böse, Luise Butthof, Lena Wendler-Link, Hendrik Wiethoff, Tanja Poth, et al. Integrated in vivo combinatorial functional genomics and spatial transcriptomics of tumours to decode genotype-to-phenotype relationships. *Nature Biomedical Engineering*, 10(1):125–143, 2026.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768, 2023.

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B. Burkhardt, Andrea Califano, Jonah Cool, Abby F. Dernburg, Kirsty Ewing, Emily B. Fox, Matthias Haury, Amy E. Herr, Eric Horvitz, Patrick D. Hsu, Viren Jain, Gregory R. Johnson, Thomas Kalil, David R. Kelley, Shana O. Kelley, Anna Kreshuk, Tim Mitchison, Stephani Otte, Jay Shendure, Nicholas J. Sofroniew, Fabian Theis, Christina V. Theodoris, Srigokul Upadhyayula, Marc Valer, Bo Wang, Eric Xing, Serena

Yeung-Levy, Marinka Zitnik, Theofanis Karaletsos, Aviv Regev, Emma Lundberg, Jure Leskovec, and Stephen R. Quake. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024. ISSN 0092-8674. doi: 10.1016/j.cell.2024.11.015. URL https://doi.org/10.1016/j.cell.2024.11.015.

Helena L Crowell, Irene Ruano, Zedong Hu, Yourae Hong, Gin Caratù, Hubert Piessevaux, Ashley Heck, Rachel Liu, Max Walter, Megan Vandenberg, et al. Tracing colorectal malignancy transformation from cell to tissue scale. *bioRxiv*, 2025.

David DeTomaso and Nir Yosef. Hotspot identifies informative gene modules across modalities of single-cell genomics. *Cell Systems*, 12(5):446–456, 2021.

Maxime Dhainaut, Samuel A Rose, Guray Akturk, Aleksandra Wroblewska, Sebastian R Nielsen, Eun Sook Park, Mark Buckup, Vladimir Roudko, Luisanna Pia, Robert Sweeney, et al. Spatial crispr genomics identifies regulators of the tumor microenvironment. *Cell*, 185(7):1223–1239, 2022.

Daniel Dimitrov, Stefan Schrod, Martin Rohbeck, and Oliver Stegle. Interpretation, extrapolation and perturbation of single cells. *Nature Reviews Genetics*, 27(5):349–370, 2026. doi: 10.1038/s41576-025-00920-4.

Mingze Dong, David G Su, Harriet Kluger, Rong Fan, and Yuval Kluger. Simvi disentangles intrinsic and spatial-induced cellular states in spatial omics data. *Nature Communications*, 16(1):2990, 2025.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

David S Fischer, Anna C Schaar, and Fabian J Theis. Modeling intercellular communication in tissues using spatial graphs of cells. *Nature Biotechnology*, 41(3):332–336, 2023. doi: 10.1038/s41587-022-01467-z.

Boyang Fu, Sameer Gabbita, George Dasoulas, Xiang Lin, Shanghua Gao, Xiaorui Su, Soumya Ghosh, and Marinka Zitnik. Strand: Sequence-conditioned transport for single-cell perturbations. *arXiv preprint arXiv:2602.10156*, 2026.

Aviv Gabbay and Yedid Hoshen. Demystifying inter-class disentanglement. *arXiv preprint arXiv:1906.11796*, 2019.

Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, et al. A python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, 40(2):163–166, 2022.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

Mahshid Heidari, Mina Karimpour, Sumana Srivatsa, and Hesam Montazeri. Evaluating single-cell perturbation response models is far from straightforward. *bioRxiv*, 2026.

Leon Hetzel, Simon Boehm, Niki Kilbertus, Stephan Günnemann, Mohammad Lotfollahi, and Fabian Theis. Predicting cellular responses to novel drug perturbations at a single-cell resolution. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 26711–26722. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/aa933b5abc1be30baece1d230ec575a7-Paper-Conference.pdf.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *International conference on artificial intelligence and statistics*, pages 2207–2217. PMLR, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *CoRR*, abs/2004.11362, 2020. URL https://arxiv.org/abs/2004.11362.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, et al. Cellflow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, 2025.

Sébastien Lachapelle, Pau Rodriguez, Yash Sharma, Katie E Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In *Conference on Causal Learning and Reasoning*, pages 428–484. PMLR, 2022.

Nathan Levy, Florian Ingelfinger, Artemii Bakulin, Giacomo Cinnirella, Pierre Boyeau, Boaz Nadler, Can Ergen, and Nir Yosef. scviva: a probabilistic framework for representation of cells and their environments in spatial transcriptomics. *bioRxiv*, 2025.

Xiang Lin, Zhenglun Kong, Soumya Ghosh, Manolis Kellis, and Marinka Zitnik. Concert predicts niche-aware perturbation responses in spatial transcriptomics. *bioRxiv*, 2025.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018.

Romain Lopez, Natasa Tagasovska, Stephen Ra, Kyunghyun Cho, Jonathan Pritchard, and Aviv Regev. Learning causal representations of single cells via sparse mechanism shift modeling. In *Conference on Causal Learning and Reasoning*, pages 662–691. PMLR, 2023.

Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):MSB202211517, 2023.

Ana Lucic, Maartje A Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. CF-GNNExplainer: Counterfactual explanations for graph neural networks. In *International conference on artificial intelligence and statistics*, pages 4499–4511. PMLR, 2022.

Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and Fabian J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, 2022. ISSN 1548-7105. doi: 10.1038/s41592-021-01336-8. URL `https://doi.org/10.1038/s41592-021-01336-8`.

Jing Ma, Ruocheng Guo, Saumitra Mishra, Aidong Zhang, and Jundong Li. CLEAR: Generative counterfactual explanations on graphs. *Advances in Neural Information Processing Systems*, 35: 25895–25907, 2022.

Stathis Megas, Daniel G. Chen, Krzysztof Polanski, Moshe Eliasof, Carola-Bibiane Schönlieb, and Sarah A Teichmann. Estimation of single-cell and tissue perturbation effect in spatial transcriptomics via spatial causal disentanglement. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=Tqdsruwyac`.

Jonathan Mitchel, Teng Gao, Viktor Petukhov, Eli Cole, and Peter V Kharchenko. Impact and correction of segmentation errors in spatial transcriptomics. *Nature Genetics*, 58(2):434–444, 2026. doi: 10.1038/s41588-025-02497-4.

Stefan Peidli, Tessa D Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J Schumacher, Jake P Taylor-King, Debora S Marks, et al. scperturb: harmonized single-cell perturbation data. *Nature Methods*, 21(3):531–540, 2024.

Zoe Piran, Niv Cohen, Yedid Hoshen, and Mor Nitzan. Disentanglement of single-cell data with biolord. *Nature Biotechnology*, 42(11):1678–1683, 2024.

Jennifer E Rood, Anna Hupalowska, and Aviv Regev. Toward a foundation model of causal cell and tissue biology with a perturbation cell and tissue atlas. *Cell*, 187(17):4520–4545, 2024.

Yusuf H Roohani, Tony J Hua, Po-Yuan Tung, Lexi R Bounds, Feiqiao B Yu, Alexander Dobin, Noam Teyssier, Abhinav Adduri, Alden Woodrow, Brian S Plosky, et al. Virtual cell challenge: Toward a turing test for the virtual cell. *Cell*, 188(13):3370–3374, 2025.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Michael Schubert, Bertram Klinger, Martina Klünemann, Anja Sieber, Florian Uhlitz, Sascha Sauer, Mathew J Garnett, Nils Blüthgen, and Julio Saez-Rodriguez. Perturbation-response genes reveal signaling footprints in cancer gene expression. *Nature Communications*, 9(1):20, 2018.

Eric D Sun, Alejandro Buendia, Anne Brunet, and James Zou. Spatialprop: tissue perturbation modeling with spatially resolved single-cell transcriptomics. *bioRxiv*, 2025. doi: 10.64898/2025.11.30.691355.

Artur Szałata, Andrew Benz, Robrecht Cannoodt, Mauricio Cortes, Jason Fong, Sunil Kuppasani, Richard Lieberman, Tianyu Liu, Javier A. Mas-Rosario, Rico Meinl, Jalil Nourisa, Jared Tumiel, Tin M. Tunjic, Mengbo Wang, Noah Weber, Hongyu Zhao, Benedict Anchang, Fabian J. Theis, Malte D. Luecken, and Daniel B. Burkhardt. A benchmark for prediction of transcriptomic responses to chemical perturbations across cell types. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 20566–20616. Curran Associates, Inc., 2024. doi: 10.52202/079017-0650. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/24c4d51f3ef48dd2dbab78243ecb26a1-Paper-Datasets_and_Benchmarks_Track.pdf.

Jovan Tanevski, Ricardo Omar Ramirez Flores, Attila Gabor, Denis Schapiro, and Julio Saez-Rodriguez. Explainable multiview framework for dissecting spatial relationships from highly multiplexed data. *Genome Biology*, 23(1):97, 2022.

Philippe Vanhems, Alain Barrat, Ciro Cattuto, Jean-François Pinton, Nagham Khanafer, Corinne Régis, Byeul-a Kim, Brigitte Comte, and Nicolas Voirin. Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PLoS ONE*, 8(9):e73970, 2013.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A Teichmann, Mor Nitzan, and Maria Brbić. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, 44(6):1050–1059, 2026. doi: 10.1038/s41587-025-02777-8.

Isaac Virshup, Danila Bredikhin, Lukas Heumos, Giovanni Palla, Gregor Sturm, Adam Gayoso, Ilia Kats, Mikaela Koutrouli, Bonnie Berger, et al. The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nature Biotechnology*, 41(5):604–606, 2023.

Hanchen Wang, Jure Leskovec, and Aviv Regev. Limitations of cell embedding metrics assessed using drifting islands. *Nature Biotechnology*, 44(4):574–577, 2026.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018.

Yan Wu, Esther Wershof, Sebastian M Schmon, Marcel Nassar, Błażej Osiński, Ridvan Eksi, Zichao Yan, Rory Stark, Kun Zhang, and Thore Graepel. PerturBench: Benchmarking machine learning models for cellular perturbation analysis. *arXiv preprint arXiv:2408.10609*, 2024.

Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular Systems Biology*, 17(1):MSB20209620, 2021.

Meng Zhang, Xingjie Pan, Won Jung, Aaron R Halpern, Stephen W Eichhorn, Zhiyun Lei, Limor Cohen, Kimberly A Smith, Bosiljka Tasic, Zizhen Yao, et al. Molecularly defined and spatially resolved cell atlas of the whole mouse brain. *Nature*, 624(7991):343–354, 2023.

A Evaluation Metrics

Unless stated otherwise, all metrics are computed using library-size–normalized gene expression, with fixed library size $\ell_0 = 10^4$. Let $c_{g,v}^{(\mathrm{obs})}$ denote the observed raw count of gene g in cell v, and $c_{g,v}^{(\mathrm{pred})}$ the corresponding model-predicted raw count. We convert both to normalized expression as

$$p_{g,v} = \frac{c_{g,v}}{\sum_{g'} c_{g',v}} \cdot \ell_0$$

We consider perturbation settings where each gene is evaluated between a control and a perturbed condition. The log-fold change (logFC) for gene g is defined as

$$\mathrm{logFC}_g = \log(p_g^{(\mathrm{pert})} + 1) - \log(p_g^{(\mathrm{ctrl})} + 1),$$

computed either from observed data or from model predictions.

For observed data, $p_g^{(\cdot)}$ denotes the empirical mean of normalized expression across cells in the corresponding condition. For model predictions, the model outputs raw counts $\hat{c}_{g,v}^{(\mathrm{pert})}$, which are first normalized and then averaged:

$$p_g^{(\mathrm{pert})} = \mathbb{E}_{v \in \mathrm{pert}} \left[\frac{\hat{c}_{g,v}^{(\mathrm{pert})}}{\sum_{g'} \hat{c}_{g',v}^{(\mathrm{pert})}} \cdot \ell_0 \right],$$

and analogously for the control condition.

In all cases, p denotes library-size–normalized gene expression, either computed from observed counts, predicted counts, or model-specific mean parameters (e.g., NB mean for generative models or inverse-transformed outputs for log-normalized models such as scGen). The observed and predicted log-fold change vectors are thus defined as:

$$\mathbf{real} = \left(\mathrm{logFC}_g^{(\mathrm{obs})}\right)_{g \in \mathcal{T}}, \quad \mathbf{pred} = \left(\mathrm{logFC}_g^{(\mathrm{pred})}\right)_{g \in \mathcal{T}}$$

We restrict evaluation to the top differentially expressed genes defined as

$$\mathcal{T} = \arg\operatorname*{top}_n \left(\left| \mathrm{logFC}_g^{(\mathrm{obs})} \right| \right)$$

Pearson r: Pearson correlation between observed and predicted log-fold changes across genes, i.e., between $\mathbf{real}$ and $\mathbf{pred}$.

Spearman ρ: Spearman rank correlation between $\mathbf{real}$ and $\mathbf{pred}$.

RMSE (counts): Root mean squared error between predicted and observed counts on ground-truth differentially expressed (DE) genes

$$\mathrm{RMSE}(\mathbf{p}, \hat{\mathbf{p}}; \mathcal{T}) = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \left(\hat{p}_i - p_i\right)^2}$$

For better readability, we report $\log_{10}(\mathrm{RMSE}_{\mathrm{counts}})$.

Signed Precision: the fraction of the top-n observed DE genes $\mathcal{T}$ that are also among the top-n genes by predicted $|\mathrm{logFC}|$ and have the same sign:

$$\mathrm{Precision}_n^{\mathrm{signed}}(\mathbf{real}, \mathbf{pred}) = \frac{1}{n} \sum_{i \in \mathcal{T} \cap \mathcal{T}^{\mathrm{pred}}} \mathbf{1}\big(\mathrm{sign}(\mathbf{real}_i) = \mathrm{sign}(\mathbf{pred}_i)\big)$$

RMSE (log-fold change): This metric evaluates magnitude differences between predicted and ground truth logFC on a selected set of differentially expressed genes. It computes the root mean squared error (RMSE) between the ground-truth and predicted values restricted to that gene set.

$$\mathrm{RMSE}_{\mathrm{LFC}}(\mathbf{real}, \mathbf{pred}; \mathcal{T}) = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \left(\mathbf{real}_i - \mathbf{pred}_i\right)^2}$$

E-distance (local): Energy distance is a widely used distribution-level metric for single-cell data [Peidli et al., 2024], measuring the overall difference between two cell populations; here, the observed (X) and predicted (Y) populations. We use the local variant proposed by Heidari et al. [2026], which restricts pairwise comparisons to k-nearest neighborhoods, making it more sensitive to disruptions of gene-gene co-expression that the global E-distance may miss. Here, $\mathcal{N}_k^X(\cdot)$ and $\mathcal{N}_k^Y(\cdot)$ denote the $k = 10$ nearest cells in X and Y by Euclidean distance, excluding the query cell itself in the within-population terms:

$$E_{\text{local}}(X,Y) = \frac{1}{nk}\sum_{i=1}^{n}\Bigg[\sum_{y'\in\mathcal{N}_k^Y(x_i)}\|x_i - y'\| + \sum_{x'\in\mathcal{N}_k^X(y_i)}\|y_i - x'\| - \sum_{x'\in\mathcal{N}_k^X(x_i)}\|x_i - x'\| - \sum_{y'\in\mathcal{N}_k^Y(y_i)}\|y_i - y'\|\Bigg]$$

Unlike the standard E-distance, which captures global distributional discrepancy, this variant measures local manifold differences. It approaches zero whenever the predicted cells lie on the local manifold of the observed ones. Following Fu et al. [2026], we fit a 50-component PCA map $T(\cdot)$ on the training split and evaluate $E_{\text{local}}\big(T(X), T(Y)\big)$ on log-normalized expression. For robustness and speed, we subsample $n = 200$ cells from each population and average over 10 subsamples.

NB deviance: E-distance above is computed on log-normalized expression, which is the space log-normalized models are trained to reconstruct, whereas *Cellina* and CPA are fit to a negative-binomial likelihood on raw counts. To assess distributional fit in count space as well, we also report the negative-binomial unit deviance of the predicted mean against the observed mean counts. To keep the comparison independent of any single model's fitted dispersion, we estimate one dispersion θ_g per gene from the observed counts of the target population and score every method under the same θ_g:

$$D = \frac{1}{G}\sum_{g=1}^{G}\max\left(0,\ 2\left[\bar{c}_g^{(\text{obs})}\log\frac{\bar{c}_g^{(\text{obs})}}{\bar{\mu}_g} - \left(\bar{c}_g^{(\text{obs})} + \theta_g\right)\log\frac{\bar{c}_g^{(\text{obs})} + \theta_g}{\bar{\mu}_g + \theta_g}\right]\right)$$

where $\bar{c}_g^{(\text{obs})} = \frac{1}{|\mathcal{V}_{\text{obs}}|}\sum_{v\in\mathcal{V}_{\text{obs}}} c_{g,v}^{(\text{obs})}$ is the observed mean, $\bar{\mu}_g = \frac{1}{|\mathcal{V}_{\text{pred}}|}\sum_{v\in\mathcal{V}_{\text{pred}}}\hat{\mu}_{g,v}$ is the method's predicted mean, $\mathcal{V}$ the evaluated cells, and the first term is taken as 0 where $\bar{c}_g^{(\text{obs})} = 0$. Because scGen and the mean shift define no dispersion of their own, and their output is inverse-transformed log-normalized expression, this is the only variant of the metric under which they are not scored against a dispersion borrowed from a competing model.

B Code and Data Availability, and Pre-Processing

B.1 Code Availability

Cellina is available at `https://github.com/PMBio/cellina` under the BSD 3-Clause License. Scripts and configurations required to download all data and reproduce the experiments presented in this work are available at `https://github.com/PMBio/cellina-reproducibility`. Tutorials for *Cellina* and *Cellina-GAT* are at `https://cellina.readthedocs.io/`. All experiments were re-run for this version of the manuscript with *Cellina* v1.2.0, which fixes an issue with log-transformation of spatial features and improves graph sampling; the shared evaluation pipeline was updated accordingly, so some baseline values also differ slightly from the submitted version. Rankings against competitors and main conclusions are unchanged.

B.2 Colorectal Cancer Patient Cohort

We downloaded the processed CRC data [Crowell et al., 2025] as AnnData files from Zenodo (https://zenodo.org/records/15574384). On each slide, we apply a standard feature selection procedure (Seurat-flavor) to subset each slide to the 2000 highly variable genes implemented in `scanpy` [Wolf et al., 2018]. The original CRC dataset contains eight slides, two of which (slide IDs 110, 222) were not used in our analyses for the following reasons: slide 110 contained sequencing artifacts in the form of major empty patches disrupting neighborhood computations, while slide 222 did not contain

any REF cells. Cell-type labels y are taken directly from the source study's annotation, and the spatial domain label d is that study's independent histopathological region annotation. For evaluations, we merged the study's fine-grained subtypes (e.g., epithelial subpopulations annotated as Epi1-Epi4) into broad categories, so that y denotes intrinsic cell lineage, independent of spatial domain. This was important since some of the subtypes, specifically epithelial subtypes, are domain-specific, so retaining them would conflate cell type identity with domain (e.g., cancer vs. healthy Epithelial cells) and make the REF $\rightarrow$ CRC prediction task circular.

For the disentanglement benchmark, all sections were merged into a single dataset prior to model fitting. For all other evaluations and downstream applications presented in this work, models were instead fit on one section at a time.

B.3 Whole-Brain Mouse Cohort

We downloaded the MERFISH whole-brain mouse cohort [Zhang et al., 2023] from the CZI CELLxGENE portal (https://datasets.cellxgene.cziscience.com/93c3bb97-ea05-4ee0-a760-a1508cd04612.h5ad), from which we selected three adjacent coronal slides from the mid-brain (C57BL6J-2.036, C57BL6J-2.039, C57BL6J-2.041). We retained all features without further selection, and the published cell-type and anatomical-region annotations are used unchanged. We further restricted the analysis to three anatomical domains, selected based on (i) their overall abundance, (ii) their consistent representation across the three selected sections, and (iii) the diversity of cell types they contained. Of these, the Thalamus was used as the source domain, while Fiber-tracts and Isocortex served as target domains.

B.4 Perturb-FISH Tumor Xenograft Dataset

We downloaded the Perturb-FISH tumor xenograft dataset of Binan et al. [2025], a human melanoma xenograft profiled with a 500-gene panel, from `https://api.brainimagelibrary.org/web/view?bildid=ace-gem-get`, and use it for the *in vivo* intervention simulation and evaluation. Cells are annotated into three categories (cancer, T cell, and other), and each cancer cell additionally carries a perturbation label indicating which NFκB-pathway gene was knocked out in it, or *Control* if unperturbed. We filter cells by a minimum count per cell and remove cells with more than one simultaneous knockout to isolate single-gene effects, yielding 154,111 annotated cells. We then subset the 500-gene panel to the 154 genes whose average expression is at least one-fifth of a count per cell. To construct the KO set used during evaluation, we note that the effects of a knockout can only propagate to a neighbor if the KO affects the cancer cell itself. We therefore keep KOs with a noticeable effect in the perturbed cells themselves, i.e. those that shift at least 3 genes in the perturbed cancer cells (Mann–Whitney U test with Benjamini–Hochberg FDR correction, $q < 0.05$, for significance and Cohen's $d > 0.2$ for effect size), and that are surrounded by at least 60 T cells. This leaves 9 KOs: MAP2K2, IRAK1, IRF7, PELI1, IRF3, IRF5, NFKBIA, LBP, and MAP2K3. For each knockout, we score predictions against unperturbed T cells with at least one neighboring cancer cell carrying that knockout and none carrying a different one. During evaluation, we sample 6,000 control T cells and simulate a perturbation in their neighborhood by setting neighbors to the mean of the KO-specific cancer population, a choice motivated by the quality of the data. The processed data is available at `https://doi.org/10.6084/m9.figshare.34790289`.

B.5 Graph Pre-Processing (Shared)

We process each sample independently to compute a spatial neighbor graph using a Gaussian proximity kernel with bandwidth $\sigma = 100\,\mu\text{m}$, consistent with the physical length scales at which neighboring cells exchange molecular signals [Armingol et al., 2021]:

$$W_{uv} = \exp\left(-\frac{d(u,v)^2}{2\sigma^2}\right) \tag{1}$$

where $d(u, v)$ is the Euclidean distance between cell centroids. W_{uv} is set to zero for pairs outside the (k) nearest neighbors of either cell or when $W_{uv} < \tau = 0.1$; self-loops are excluded ($W_{vv} = 0$). Nearest neighbors parameter k varies per experiment and model: for *Cellina* we use $k = 200$ for CRC and MERFISH experiments while for *Cellina-GAT*, we use $k = 50$ for both datasets due to the high training cost of *Cellina-GAT*. Both use $k = 30$ for the *in vivo* experiment. Furthermore, for

Cellina-GAT we use binary edge weights ($W_{uv} \in \{0, 1\}$), letting attention learn edge importance during message passing.

B.6 Edge-Perturbation Donor Sampling

In this work, we simulate **edge perturbation** as an in-silico domain edge rewiring. Let $\mathcal{I}_y \subset \mathcal{V}$ denote the focal cells of type y in the source domain d, whose counterfactual expression we wish to predict, and let the *anchors* $\mathcal{A}_y$ be the cells of the same type y in the target domain d'. For each focal cell $v \in \mathcal{I}_y$, we drop its original edges and assign it new neighbors $\mathcal{N}'(v)$ taken from the neighborhoods that the anchors have in d', within the same slide. In all leave-one-out benchmarks, cells of type y are removed from these neighborhoods, and the anchors are excluded from the neighborhood graph W during both training and inference, i.e., their expression never enters the model and serves solely as evaluation ground truth; the anchors' neighborhoods $\mathcal{N}(c)$ are read from the unmasked spatial graph.

We implement two draws for $\mathcal{N}'(v)$ (Algorithms 1 and 2). In the *anchored* draw, each focal cell inherits the observed neighborhood of one anchor $c \sim \mathrm{Uniform}(\mathcal{A}_y)$, sampled with replacement across focal cells, together with its Gaussian-kernel weights (Equation 1; binary edges for *Cellina-GAT*); i.e., the focal cell is placed into a real neighborhood from d'. In the *rewired* draw, we instead pool the neighbors of all anchors into $\mathcal{P}_{\setminus y}$ and sample $n_d = 50$ donors per focal cell from this pool without replacement, attached with unit weights; the resulting neighborhood consists of cells observed around type-y cells in d', but does not correspond to any single observed neighborhood. In both draws, $\varphi(v)$ (*Cellina*) or the local subgraph $\mathcal{G}_v$ (*Cellina-GAT*) is then rebuilt from the new neighbors.

All leave-one-out edge perturbation benchmarks use the anchored draw for both *Cellina* and *Cellina-GAT*. The rewired draw is used where the target composition does not occur in the tissue; for example, for the homotypic neighborhoods of Appendix K, where the donor pool is instead all cells of a single type in d', including the held-out type y itself. The draws are made once per run with a fixed seed and shared between expression and latent predictions.

Algorithm 1 Edge perturbation, *anchored* draw. Focal cells $\mathcal{I}_y$: type y in source domain d. Anchors $\mathcal{A}_y$: type-y cells in target domain d', held out from W at training and inference; $\mathcal{N}(c)$ is read from the unmasked spatial graph, so anchors contribute only their neighborhoods, never their expression.

Cellina
drop every edge incident on $\mathcal{I}_y$
for each focal cell $v \in \mathcal{I}_y$ **do**
  $c \sim \mathrm{Uniform}(\mathcal{A}_y)$ ▷ with replacement across focal cells
  $\mathcal{N}'(v) \leftarrow \mathcal{N}(c) \setminus \{u : y_u = y\}$ with the kernel weights W_{uc} of Equation 1
  $\varphi(v) \leftarrow \sum_{u \in \mathcal{N}'(v)} W_{uc}\, \tilde{x}_u \,/\, \sum_{u \in \mathcal{N}'(v)} W_{uc}$ ▷ recomputed weighted mean
end for
Cellina-GAT
drop every edge incident on $\mathcal{I}_y$
for each focal cell $v \in \mathcal{I}_y$ **do**
  $c \sim \mathrm{Uniform}(\mathcal{A}_y)$ ▷ with replacement across focal cells
  $\mathcal{N}'(v) \leftarrow \mathcal{N}(c) \setminus \{u : y_u = y\}$
  add edges $u \to v$ for all $u \in \mathcal{N}'(v)$
end for
rebuild the local subgraph $\mathcal{G}_v$ for all $v \in \mathcal{I}_y$ and encode $s \sim q(s \mid \mathcal{G}_v)$ by message passing

B.7 Node-Perturbation Shift and Gene-Set Sweep

Recall that node perturbation modifies a subset $\mathcal{S}$ of $m = |\mathcal{S}|$ genes per neighbor cell. In our experiments, we alter the genes of neighbors in source domains by the log fold-change (logFC) shift δ, representing the average difference between pseudobulked cell populations. Concretely, δ is added to the log-normalized count $\tilde{x}_{u,g}$ of each neighbor u, i.e. $\tilde{x}^{\mathrm{cf}}_{u,g} = \tilde{x}_{u,g} + \delta_g$ for $g \in \mathcal{S}$ (with δ_g replaced by $\delta_{y(u),g}$ in the cell-type-specific variant below). Specifically, we calculate a **global** and a **cell-type-specific** δ. Let $\ell_0 = 10^4$ denote the target library size, $\mathcal{C}^d$ the set of cells in domain $d \in \{\mathrm{REF}, \mathrm{CRC}\}$, $\mathcal{C}^d_y \subseteq \mathcal{C}^d$ its restriction to cell type y, and $x_{i,g}$ the raw count of gene g in cell i.

Algorithm 2 Edge perturbation, *rewired* draw. Focal cells $\mathcal{I}_y$ and anchors $\mathcal{A}_y$ as in Algorithm 1. Donor pool $\mathcal{P}_{\setminus y} = \bigcup_{c \in \mathcal{A}_y} \mathcal{N}(c) \setminus \{u : y_u = y\}$.

Cellina
drop every edge incident on $\mathcal{I}_y$
for each focal cell $v \in \mathcal{I}_y$ **do**
  $\mathcal{N}'(v) \sim \mathrm{Sample}(\mathcal{P}_{\setminus y}, n_d,$ without replacement$)$
  add unit-weight edges $v \leftrightarrow \mathcal{N}'(v)$
  $\varphi(v) \leftarrow \frac{1}{n_d} \sum_{u \in \mathcal{N}'(v)} \tilde{x}_u$ ▷ recomputed mean aggregate
end for
Cellina-GAT
drop every edge incident on $\mathcal{I}_y$
for each focal cell $v \in \mathcal{I}_y$ **do**
  $\mathcal{N}'(v) \sim \mathrm{Sample}(\mathcal{P}_{\setminus y}, n_d,$ without replacement$)$
  add edges $u \to v$ for all $u \in \mathcal{N}'(v)$
end for
rebuild the local subgraph $\mathcal{G}_v$ for all $v \in \mathcal{I}_y$ and encode $s \sim q(s \mid \mathcal{G}_v)$ by message passing

- **Global.** Pseudobulk across all cells per domain, $b_g^d = \sum_{i \in \mathcal{C}^d} x_{i,g}$, library-normalize, and log-transform: $\tilde{b}_g^d = \log\Big(1 + \ell_0\, b_g^d / \sum_{g'} b_{g'}^d\Big)$. The global shift is $\delta_g = \tilde{b}_g^{\mathrm{CRC}} - \tilde{b}_g^{\mathrm{REF}}$.
- **Cell-type-specific.** Pseudobulk within each cell type y and domain, $b_{y,g}^d = \sum_{i \in \mathcal{C}_y^d} x_{i,g}$, library-normalize, and log-transform: $\tilde{b}_{y,g}^d = \log\Big(1 + \ell_0\, b_{y,g}^d / \sum_{g'} b_{y,g'}^d\Big)$. The cell-type-specific shift is $\delta_{y,g} = \tilde{b}_{y,g}^{\mathrm{CRC}} - \tilde{b}_{y,g}^{\mathrm{REF}}$.

Note that *Cellina*'s neighborhoods aggregate log1p-normalized expression, while *Cellina-GAT* operates on raw count data. Accordingly, for *Cellina*, the perturbation is an additive shift in log-normalized space ($\tilde{x}^{\mathrm{cf}} = \tilde{x} + \delta$), whereas for *Cellina-GAT* the corresponding operation is a multiplicative scaling on counts ($x^{\mathrm{cf}} = x \cdot e^{\delta}$), obtained by exponentiating the logFC shift. As such, *Cellina-GAT* applies T_g directly in count space, $T_g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Also, to preserve a strict counterfactual setting, for the held-out cell type y we assign the global shift $\delta_{\setminus y}$, pseudobulked with type y removed, so the perturbed input carries the domain-level shift only and never the held-out population's own signature.

In this sweep (Figure A1), we tested the performance of *Cellina* as m approaches the total number of genes G, i.e., every gene of every neighbor is altered by δ_g or $\delta_{y,g}$, producing a transcriptome-wide shift for all neighborhoods toward the target domain profile. Here, we see that all four metrics improve with m for both logFC variants and saturate by $m \approx 100$–200, collapsing shortly after (Figure A1). At $m = 200$ (in-distribution), the cell-type-specific variant consistently outperforms the global one and reaches Pearson $r \approx 0.89$, Signed Precision ≈ 0.56, E-distance ≈ 1.05, and $\mathrm{RMSE}_{\mathrm{LFC}}$ of 0.66, compared with the edge perturbation ceiling of 0.95, 0.69, 0.50, and 0.49 respectively. Node perturbation therefore saturates *below* the edge-perturbation reference for every gene set, because δ is a population-level logFC: neighbors are shifted toward an average target profile rather than onto the profiles of real observed cells, so no choice of $\mathcal{S}$ recovers a real neighborhood's composition (Appendix O). We assume that this is a biologically meaningful result: biological perturbations are often assumed to elicit relatively sparse shifts on a few genes, rather than in full expression space [Lopez et al., 2023].

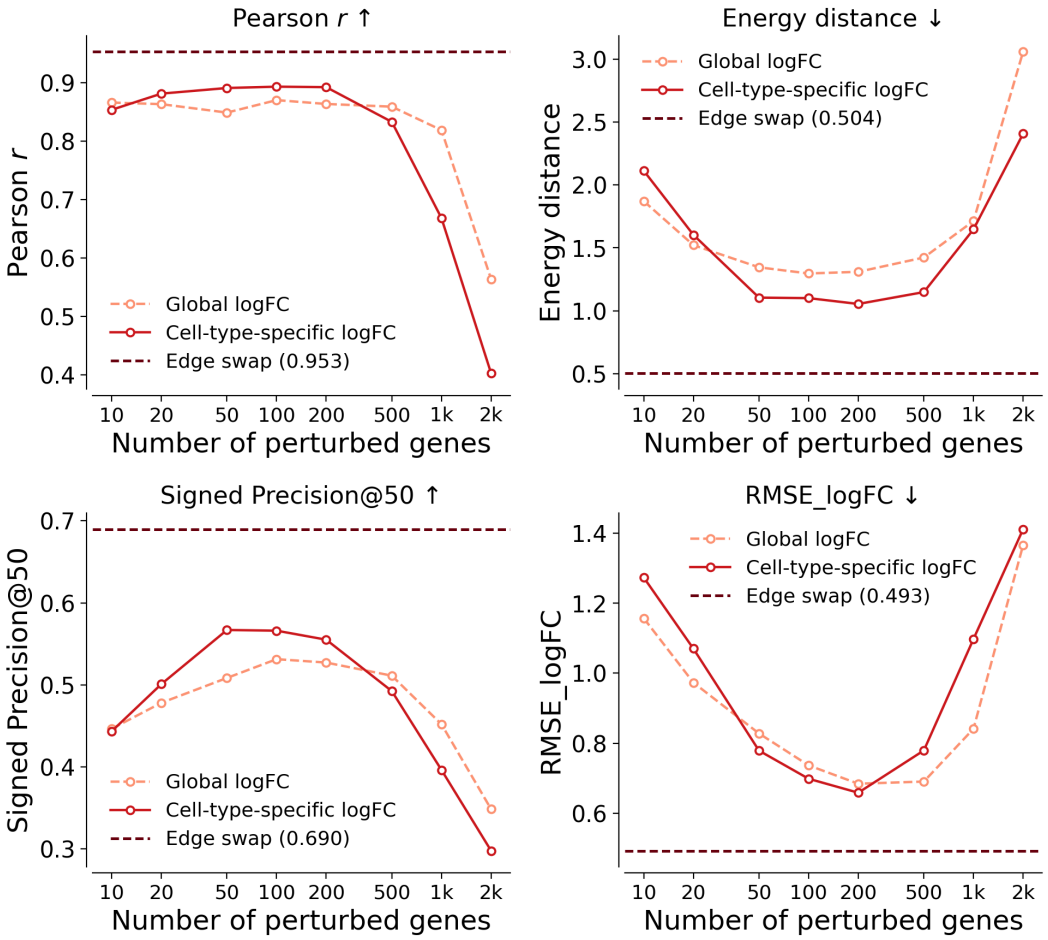

Figure A1: **Node perturbation as a function of the perturbed gene set.** We sweep over a range of gene sets and evaluate in-distribution node perturbations against the target domain under two choices of the logFC shift δ_g: a *global* logFC pooled across all cells (dashed, light), and a *cell-type-specific* $\delta_{y,g}$ logFC computed within cell-type groups (solid, dark). The maroon dashed line marks the edge perturbation reference, i.e. a complete neighborhood swap, plus sampling from the domain target neighbors.

C Hyperparameters, Architecture, and Running Time Details

We implement *Cellina* using the `scvi-tools` API [Gayoso et al., 2022] – a standard framework in single-cell genomics. Hence, we adopt most default hyperparameters and best practices recommended by `scvi-tools`. To ensure a consistent comparison, all methods are evaluated using their default settings across experiments and whenever possible using count likelihoods.

C.1 Compute Resources

All experiments were conducted on GPU machines with four NVIDIA GeForce RTX 4090, AMD Ryzen Threadripper PRO 7975WX 32-Cores, and 500GB of RAM.

We provide training times for competitors and *Cellina* variants in Figure A9. A single training and inference run of *Cellina* takes approximately one hour on the CRC slides from Crowell et al. [2025] and approximately 20 minutes on the mouse brain data from Zhang et al. [2023]. The in-silico perturbation evaluations reported in Tables 1 and 2 are the most expensive experiments, requiring roughly ten and two days of compute, respectively. The ablation sweeps in the Appendix require three to five days, while the remaining appendix experiments each complete within one day. In total,

all reported experiments required approximately two to three weeks of GPU time on the hardware described above.

Table A1: Architecture and training hyperparameters. *Cellina* uses a VAE architecture, with standard scvi-tools defaults for single-cell data, trained with unit weights on all active regularization terms, each normalized to the VAE loss scale (the graph contrastive term applies to *Cellina-GAT* only).

Component	Parameter	Cellina	Cellina-GAT
z-encoder	Hidden dim	128	128
z-encoder	Layers	2	3
z-encoder	Latent dim m	64	64
s-encoder	Hidden dim	128	128
s-encoder	Layers	2	3
s-encoder	Latent dim m	64	64
Decoder	Hidden dim	128	128
Decoder	Layers	2	3
Discriminator	Hidden dim	32	32
Discriminator	Layers	2	2
GNN	Convolution	—	GATv2
Training	Batch size	2048	256
Training	Max epochs	100	100
Training	Optimizer	Adam	Adam
Training	Learning rate	10^{-3}	10^{-3}
Training	Weight decay	10^{-6}	10^{-4}
Training	KL warmup	linear, $0 \to 1$	linear, $0 \to 1$
Training	λ_{clf}	1	1
Training	$\lambda_{\mathrm{disc}} = \lambda_{\mathrm{adv}}$ (tied)	1	1
Training	$\lambda_{\mathrm{spatial}}$	0	1
Spatial graph	Bandwidth	100 μm	100 μm
Spatial graph	Kernel	Gaussian	—
Spatial graph	Cutoff	0.1	—
Count Decoding	Distribution	Negative Binomial	Negative Binomial

D Model Details and Training Objective

D.1 Generative Model

Cellina is a (graph) variational autoencoder [Kingma and Welling, 2013, Kipf and Welling, 2016] with two latent variables:

- $z \in \mathbb{R}^m$: intrinsic cell identity, capturing variation independent of spatial context
- $s \in \mathbb{R}^m$: spatial niche representation, capturing microenvironmental variation

The joint generative model is:

$$p(x, z, s) = p(x \mid z, s)\, p(z)\, p(s)$$

with standard normal priors $p(z) = p(s) = \mathrm{Normal}(0, I_m)$.

The likelihood used in the evaluations throughout this study is a Negative Binomial (NB) distribution over counts, a standard choice in modeling single-cell data [Lopez et al., 2018, Gayoso et al., 2022], with parameters produced by a decoder operating on $[z;\, s] \in \mathbb{R}^{2m}$:

$$x \sim \mathrm{NB}(\mu, r), \quad \text{with } \mu = \exp(\ell)\, \mu_\theta([z;\, s], b),\ r = r_\theta([z;\, s], b)$$

where b is a one-hot-encoded sequencing batch covariate injected into the decoder, $\mu_\theta \in \mathbb{R}^G_{>0}$, $r_\theta \in \mathbb{R}^G_{>0}$ are learnable per-gene mean and inverse dispersion parameters, and $\ell = \log \sum_g x_g$ is the observed log-library size used to scale the NB rate. The decoder input dimensionality is $2m$, reflecting the concatenation of both latent variables. The model supports conditioning on b in both the encoders and the decoder, but all experiments reported in this work use a single batch, so b is fixed and omitted from the main-text notation.

D.2 Inference Model

Both *Cellina* variants share the z encoder (an MLP with counts as input); they differ only in how the niche input is constructed and how s is encoded, as detailed below.

The approximate posterior factorizes differently per variant:

$$\begin{aligned} \text{Cellina:} \quad & q(z, s \mid x, \varphi_v) = q(z \mid x)\, q(s \mid \varphi_v) \\ \text{Cellina-GAT:} \quad & q(z, s \mid x, \mathcal{G}_v) = q(z \mid x)\, q(s \mid \mathcal{G}_v) \end{aligned}$$

where $\mathcal{G}_v = (x_v,\, \{x_u\}_{u \in \mathcal{N}(v)},\, \mathcal{E}_v)$ denotes v's local subgraph. Both variants share how z is encoded: an MLP parameterizing a diagonal Gaussian over counts,

$$q(z \mid x) = \text{Normal}(\mu_z(x,\, b),\ \sigma_z^2(x,\, b))$$

Cellina (base). The spatial input is the degree-normalized aggregation of log-normalized neighbor expression. Let $\mathcal{N}(v) = \{u : W_{uv} > 0\}$ and $\tilde{X} \in \mathbb{R}^{N \times G}$ the matrix of log-normalized counts, where

$$\tilde{x}_{u,g} = \log\!\left(1 + \frac{x_{u,g}}{\ell_u} \cdot \ell_0\right), \qquad \ell_u = \sum_g x_{u,g}, \quad \ell_0 = 10^4,$$

with ℓ_u denoting the library size (total counts per cell). Spatial weights are given by a Gaussian proximity kernel as described in equation (1). The niche feature vector is then:

$$\varphi(v) = \frac{\sum_{u \in \mathcal{N}(v)} W_{uv}\, \tilde{x}_u}{\sum_{u \in \mathcal{N}(v)} W_{uv}}$$

a simple degree-normalized aggregation of neighbor expression. The s encoder is an MLP:

$$q(s \mid \varphi_v) = \text{Normal}(\mu_s(\varphi_v,\, b),\ \sigma_s^2(\varphi_v,\, b))$$

Cellina-GAT. The s encoder is replaced by a graph neural network f_s that processes v's local subgraph $(x_v,\, \{x_u\}_{u \in \mathcal{N}(v)},\, \mathcal{E}_v)$ directly (log1p-transformed inside f_s), where $\mathcal{E}_v$ is the local edge set derived from the same proximity graph W (i.e. $\mathcal{E}_v = \{\{u, v\} : W_{uv} > 0\}$); edges are binarized ($W_{uv} \in \{0, 1\}$), so GATv2 receives only the graph topology — attention weights are learned entirely from gene expression. f_s is a multi-layer GATv2 [Brody et al., 2021], implemented via `pytorch-geometric` [Fey and Lenssen, 2019]. Self-loops are excluded, so f_s does not directly aggregate v's own expression, which is captured by z. The posterior conditions on the local subgraph:

$$q(s \mid \mathcal{G}_v) = \text{Normal}\big(\mu_s^{\mathcal{G}}(\mathcal{G}_v),\ \sigma_s^{2\mathcal{G}}(\mathcal{G}_v)\big)$$

where $\mu_s^{\mathcal{G}}$ and $\sigma_s^{2\mathcal{G}}$ are linear projections applied to the seed-node representation after L rounds of message passing. The learned attention weights can be extracted per edge via *Cellina-GAT*'s API, providing a neighbor-level interpretability readout unavailable to the pre-aggregated encoder.

Samples are drawn via the reparameterization trick [Kingma and Welling, 2013].

D.3 Training Objective

ELBO. The variational lower bound is:

$$\text{ELBO} = \mathbb{E}_q\big[\log p(x \mid z, s)\big] - \beta_t \Big[\text{KL}\big(q(z \mid x) \| p(z)\big) + \text{KL}\big(q(s \mid \varphi_v) \| p(s)\big)\Big]$$

where β_t is a KL warmup schedule increasing linearly from 0 to 1 [Gayoso et al., 2022], and for *Cellina-GAT*, we replace φ_v with $\mathcal{G}_v$. The size of the library is treated as observed ($\ell = \log \sum_g x_g$), so no KL term appears in the library. We minimize the negative ELBO:

$$\mathcal{L}_{\text{VAE}} = -\text{ELBO}$$

Dual supervised disentanglement on z. Optimizing $\mathcal{L}_{\mathrm{VAE}}$ alone does not prevent z from absorbing spatial variation. To enforce a meaningful partition, we apply two additional objectives exclusively to z.

We write $\Delta^K = \{p \in \mathbb{R}^K_{\geq 0} : \sum_i p_i = 1\}$ for the $(K-1)$-dimensional probability simplex and $\mathrm{sg}(\cdot)$ for the stop-gradient operator sg.

A cell-type classifier $f_{\mathrm{clf}} : \mathbb{R}^m \to \Delta^C$ is trained jointly to predict the cell-type label y from z:

$$\mathcal{L}_{\mathrm{clf}} = \mathbb{E}\big[-\log f_{\mathrm{clf}}(y \mid z)\big]$$

An adversarial domain discriminator $f_{\mathrm{disc}} : \mathbb{R}^m \to \Delta^D$ is trained in a two-step alternating procedure. In step 1 (VAE frozen), the discriminator is trained to predict the spatial domain label d from a detached z:

$$\mathcal{L}_{\mathrm{disc}} = \mathbb{E}\big[-\log f_{\mathrm{disc}}(d \mid \mathrm{sg}(z))\big]$$

In step 2 (discriminator frozen), the VAE is trained to fool the discriminator by maximizing its cross-entropy, that is, minimizing the negated cross-entropy with weight -1:

$$\mathcal{L}_{\mathrm{adv}} = \mathbb{E}\big[-\log f_{\mathrm{disc}}(d \mid z)\big]$$

In the base *Cellina* variant, s is unsupervised; *Cellina-GAT* additionally applies a graph contrastive loss on s.

Spatial contrastive loss for *Cellina-GAT*. For the *Cellina-GAT* variant we additionally consider a modified graph-supervised contrastive loss on s. The loss operates on the ℓ_2-normalized spatial-latent posterior mean

$$\hat{s}_v \;=\; \mu_s(v) \,/\, \|\mu_s(v)\|_2,$$

and uses scaled cosine similarity $\mathrm{sim}_\tau(v,u) = \hat{s}_v^\top \hat{s}_u / \tau$ with temperature $\tau = 0.25$. For a mini-batch of anchor cells $\mathcal{B} \subseteq \mathcal{V}$ and an anchor $v \in \mathcal{B}$, define

$$\begin{aligned}
\mathcal{P}(v) &= \mathcal{N}(v) && \text{(positives: spatial neighbors of } v\text{)} \\
\mathcal{Q}(v) &= \{\, u \in \mathcal{V} : d_u \neq d_v,\ u \notin \mathcal{N}(v) \cup \{v\} \,\} && \text{(negatives: different-domain non-neighbors)} \\
\mathcal{A}(v) &= \mathcal{P}(v) \cup \mathcal{Q}(v).
\end{aligned}$$

The per-anchor loss is

$$\ell_{\mathrm{SupCon}}(v) \;=\; -\frac{1}{|\mathcal{P}(v)|} \sum_{p \,\in\, \mathcal{P}(v)} \log \frac{\exp\big(\mathrm{sim}_\tau(v,p)\big)}{\displaystyle\sum_{a \,\in\, \mathcal{A}(v)} \exp\big(\mathrm{sim}_\tau(v,a)\big)},$$

and the batch-level loss averages over anchors with at least one valid positive and negative, both subsampled from $\mathcal{B}$:

$$\mathcal{L}_{\mathrm{spatial}} \;=\; \frac{1}{|\mathcal{B}^\star|} \sum_{v \in \mathcal{B}^\star} \ell_{\mathrm{SupCon}}(v), \qquad \mathcal{B}^\star \;=\; \{\, v \in \mathcal{B} : \mathcal{P}(v) \neq \emptyset,\ \mathcal{Q}(v) \neq \emptyset \,\}.$$

This is a variant of SupCon [Khosla et al., 2020], in which positives are induced by the spatial graph $\mathcal{E}$ rather than by class label, while supervision enters through the negative set via domain labels d_v. We motivate grounding positives in the spatial graph rather than in labels to avoid the collapse behavior of label-supervised contrastive objectives: with domain labels as positives, s could degenerate toward one embedding per domain, erasing exactly the within-domain neighborhood variation it is meant to capture. Treating local neighborhood membership as the positive criterion encodes a biologically motivated inductive bias — a cell's relevant microenvironment is its immediate spatial context, not the coarse domain partition. Same-domain non-neighbors are excluded from both sets, avoiding ambiguous supervision from cells whose spatial relationship to the anchor is uninformative. For example, cells at the extremes of the same domain (or tissue region) may be completely unrelated.

D.4 Loss Normalization

The λ_{clf}, λ_{disc}, λ_{adv}, and λ_{spatial} weights (Table A1) control the relative importance of each auxiliary objective but do not account for the inherent scale differences between the reconstruction loss and the auxiliary terms. To prevent any auxiliary objective from dominating training and to simplify the choice of weights, we additionally compute fixed normalization scales α from the raw loss values observed during the first training epoch:

$$\alpha_{\text{clf}} = \frac{\overline{|\mathcal{L}_{\text{VAE}}|}}{\overline{|\mathcal{L}_{\text{clf}}|} + \epsilon}, \quad \alpha_{\text{adv}} = \frac{\overline{|\mathcal{L}_{\text{VAE}}|}}{\overline{|\mathcal{L}_{\text{adv}}|} + \epsilon}, \quad \alpha_{\text{disc}} = \frac{\overline{|\mathcal{L}_{\text{VAE}}|}}{\overline{|\mathcal{L}_{\text{disc}}|} + \epsilon}, \quad \alpha_{\text{spatial}} = \frac{\overline{|\mathcal{L}_{\text{VAE}}|}}{\overline{|\mathcal{L}_{\text{spatial}}|} + \epsilon}$$

where overbars denote epoch-0 means and $\epsilon = 10^{-8}$. These scales are fixed after the first epoch. The full training objective in step 2 is then:

$$\mathcal{L} = \mathcal{L}_{\text{VAE}} + \lambda_{\text{clf}}\,\alpha_{\text{clf}}\,\mathcal{L}_{\text{clf}} + \lambda_{\text{spatial}}\,\alpha_{\text{spatial}}\,\mathcal{L}_{\text{spatial}} - \lambda_{\text{adv}}\,\alpha_{\text{adv}}\,\mathcal{L}_{\text{adv}}$$

minimized over encoder and decoder parameters, with the discriminator frozen. The ablation studies and sweeps in Appendix E show that, with this normalization in place, unit weights, used throughout our work, are a robust default for λ_{clf}, λ_{disc}, and λ_{spatial}.

D.5 Adversarial Training Procedure

The two-step alternating training is implemented via PyTorch Lightning's manual optimization (`https://github.com/Lightning-AI/pytorch-lightning`). Both optimizers are Adam. The VAE optimizer covers all parameters except the discriminator head; the discriminator optimizer covers the discriminator head only. In each training step:

Step 1. Freeze VAE. Sample z without gradients. Minimize discriminator cross-entropy:

$$\theta_{\text{disc}} \leftarrow \theta_{\text{disc}} - \eta \nabla_{\theta_{\text{disc}}} \left[\lambda_{\text{disc}} \cdot \mathbb{E}[-\log f_{\text{disc}}(d \mid \text{sg}(z))]\right]$$

Step 2. Freeze the discriminator. Minimize VAE + classifier + adversarial loss:

$$\theta_{\text{VAE}} \leftarrow \theta_{\text{VAE}} - \eta \nabla_{\theta_{\text{VAE}}} \left[\mathcal{L}_{\text{VAE}} + \lambda_{\text{clf}}\mathcal{L}_{\text{clf}} + \lambda_{\text{spatial}}\mathcal{L}_{\text{spatial}} - \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}\right]$$

In step 2, the adversary λ is negated: minimizing $-\lambda_{\text{adv}} \cdot \mathbb{E}[-\log f_{\text{disc}}(d \mid z)]$ is equivalent to maximizing the adversary's cross-entropy — i.e., encoding z such that the adversary cannot recover d.

E Component Ablations and Loss-Weight Sensitivity

We ablate each model component independently by sweeping its loss weight, one λ at a time, over $\{0, 10^{-9}, 10^{-7}, 10^{-5}, 10^{-3}, 0.1, 1, 10, 100, 10^{3}\}$ on a single CRC slide, while holding all other λ values at 10^{-7}; the $\lambda = 0$ endpoint removes the component entirely. We train five random seeds per setting and evaluate on a 10% holdout via macro-F1 from a logistic-regression probe trained on the frozen latents (measuring the cell-type and spatial-domain information each latent carries) and marginal log-likelihood (MLL, $N = 500$ samples). Results are shown in Figure A2.

Cell-type classifier (λ_{clf}). As we increase λ_{clf}, cell-type F1 on z improves, indicating that the classifier successfully anchors z to cell identity. We also observe a drop in spatial-domain information carried by z, which we take as evidence that the disentanglement is working as intended. Both effects level off around $\lambda_{\text{clf}} = 1$, and there is only a modest MLL cost for them.

Domain adversary ($\lambda_{\text{disc}} = \lambda_{\text{adv}}$). Larger λ_{disc} values push spatial-domain accuracy on z down, showing that the adversary is fulfilling its purpose and stripping domain-level signal from z: domain macro-F1 decodable from frozen z falls from ≈ 0.57 with the adversary off to ≈ 0.40 at the deployed unit weight. Cell-type F1 is largely retained ($\approx 0.66 \rightarrow 0.60$); we attribute this small dip to the entanglement between cell-type composition and spatial domain (i.e., domains are to a certain extent defined by their cell-type composition). What the adversary removes at unit weight is thus largely domain-predictive, not lineage-predictive, information. Beyond unit weight this trade-off degrades, with cell-type F1 dropping to ≈ 0.24 at a high cost in MLL.

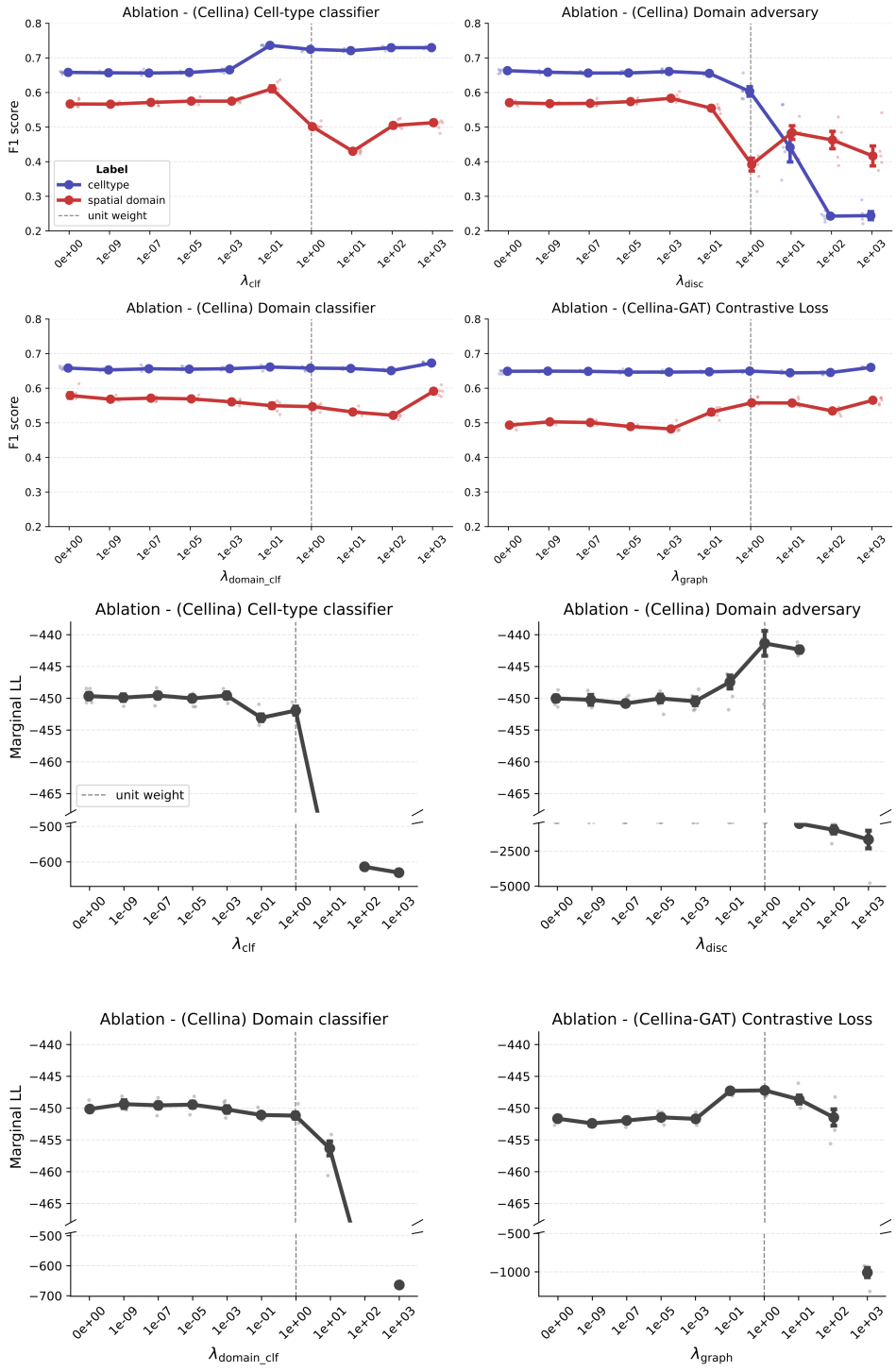

Figure A2: Lambda sweep ablations. Each panel varies one λ (others set to 10^{-7}). *Top:* macro-F1 from logistic-regression probes on the frozen latents, evaluated against cell-type y and spatial-domain d labels; the top row (cell-type classifier, domain adversary) probes z, and the bottom row (domain classifier on *Cellina*'s s, *Cellina-GAT* contrastive loss) probes s. *Bottom:* marginal log-likelihood. Lines connect median values; error bars show standard error over five seeds; dots show individual seeds. Overall, we see that unit weights provide a good balance between disentanglement and fit quality. The axis label λ_{graph} denotes λ_{spatial}.

Domain classifier on s ($\lambda_{\text{domain_clf}}$). We also experimented with a supervised domain-classification loss on s, but it did not improve spatial-domain F1 from s and introduced a small MLL penalty at $\lambda = 1$. We therefore drop this term from *Cellina* and leave s entirely unsupervised in the base variant.

Graph contrastive loss (λ_{spatial}, *Cellina-GAT* only). The modified contrastive loss described above raises spatial-domain F1 from s ($\approx 0.49 \rightarrow 0.56$ at unit weight) and gives a slight MLL improvement, which we view as evidence that it provides a useful inductive bias for *Cellina*'s GAT variant.

Taken together, MLL stays largely flat around $\lambda = 1$ and only degrades for substantially larger values, which suggests that our normalization scheme (Appendix D.4) gives a robust default at unit weights and removes the need for per-dataset tuning. As such, for all experiments reported in this paper, we set $\lambda = 1$, and thus omit it from the loss definition in the main text.

Architecture ablations. We also check whether the encoders need to be variational, and whether attention contributes to performance over plain graph convolution. We test both on the CRC edge perturbation task, with the same protocol as Table 1 (Table A2). *Cellina-AE* replaces the variational encoders with a deterministic autoencoder, i.e. it outputs point embeddings and drops the KL terms from the objective (Appendix D.3). *Cellina-GCN* swaps the GATv2 layers of *Cellina-GAT* for graph-convolution layers of the same depth and width. The deterministic encoder *Cellina-AE* does slightly better on three of the four metrics and ties on Signed Precision. If anything, this variant would widen the margins over the baselines in Table 1; we keep the variational formulation, which all reported experiments used. GAT and GCN are indistinguishable, which is in line with the small gap between *Cellina-GAT* and *Cellina* in Table 1. On these data, the gains largely come from conditioning on the neighborhood and from the supervised disentanglement (*Cellina (ablated)* in Table 1), not from how the neighbors are aggregated. We, nevertheless, keep the attention variant as default, as its per-edge weights give a neighbor-level readout of which cells drive a prediction (Appendix D.2), which GCN does not offer.

Table A2: *Cellina* architecture comparison. CRC edge perturbation results for variational vs. standard (deterministic) autoencoder (AE) variants of *Cellina*, and Graph Attention (GAT) vs. Graph Convolution (GCN) encoders for *Cellina-GAT*. Same protocol and reporting as Table 1.

Model	Pearson ↑	$\text{Precision}_{\text{signed}}$ ↑	E-distance ↓	RMSE_{LFC} ↓
Cellina	0.86 ± 0.17	0.45 ± 0.19	7.00 ± 1.05	1.14 ± 0.56
Cellina-AE	0.89 ± 0.10	0.45 ± 0.17	6.29 ± 1.13	1.07 ± 0.44
Cellina-GAT	0.89 ± 0.09	0.42 ± 0.20	8.59 ± 1.90	1.09 ± 0.61
Cellina-GCN	0.89 ± 0.10	0.43 ± 0.20	8.64 ± 1.34	1.07 ± 0.60

F Related Methods

scVI [Lopez et al., 2018] is a conditional VAE for single-cell RNA-seq data. It models raw counts with a negative binomial (or zero-inflated negative binomial) likelihood, decomposing each cell's expression into a low-dimensional latent representation and a separately inferred library-size factor. Batch effects (and other covariates) are mitigated by conditioning the encoder and decoder on the corresponding labels, typically encoded as one-hot vectors. scVI is a popular choice for dimensionality reduction in the single-cell field, and its code base has since been expanded into **scvi-tools** [Gayoso et al., 2022], a probabilistic modeling framework for single-cell omics; several of the related methods in this section, as well as *Cellina*, are built on scvi-tools.

scANVI [Xu et al., 2021] is a semi-supervised extension of scVI in which the latent space is structured by cell-type identity. On top of scVI's per-cell latent representation, scANVI introduces a second, label-conditional latent variable and a classifier that predicts the cell-type label from scVI's latent. During training, labeled cells contribute an additional cross-entropy term, resulting in an explicitly supervised latent. scANVI is widely used for cell-type label transfer, particularly when annotating a query dataset against a labeled reference.

scGen [Lotfollahi et al., 2019] is a VAE that predicts perturbation responses via latent-space arithmetic. The model is trained to reconstruct normalized gene expression profiles through a low-dimensional latent space; the effect of a perturbation is then summarized by a difference vector δ, calculated as the difference between the mean latent representations of perturbed and unperturbed training cells.

To predict the response of an unseen (test) cell-type population, δ is added to the latent representation of each unperturbed test cell, and the resulting vector is decoded back to gene expression space. For the two largest slides (120 and 210), scGen's `predict` functions failed with internal errors, which were resolved by training on subsets of the slides; we therefore sub-sampled the data for scGen to 30% on these slides.

CPA [Lotfollahi et al., 2023] models single-cell gene expression as additive compositions of disentangled latent factors: a basal cell state plus separate embeddings for each perturbation and covariate (e.g., cell type, drug). Only the basal state is produced by the encoder; perturbation and covariate embeddings are learned per label (or dosage) and added before decoding. Disentanglement is enforced adversarially: auxiliary classifiers are trained to recover the labels from the basal embedding, and the encoder is penalized whenever they succeed.

scVIVA [Levy et al., 2025] is a VAE-based model designed for spatial transcriptomics data. It learns a shared embedding that captures both intrinsic cell state and microenvironment context. To effectively add spatial information in the latent embedding, scVIVA predicts niche gene expression and neighbor composition (proportion of each cell type in a given neighborhood) in addition to denoised gene expression of query cells. It has no mechanism for disentanglement and cannot answer counterfactual questions; it therefore cannot generate predictions under an altered neighborhood and is excluded from the in-silico perturbation benchmarks.

SIMVI [Dong et al., 2025] is a spatially-informed VAE that disentangles gene expression variability into two latent factors: an intrinsic variable z and a spatial variable s. The spatial latent s is inferred by aggregating intrinsic representations of neighboring cells via a Graph Attention Network. To promote independence between z and s, SIMVI uses an additional unsupervised regularization: an MMD-based term that promotes independence between (z, s) or, alternatively, a mutual-information penalty. We excluded SIMVI from our counterfactual benchmarks because it exceeds available memory at 10^5 cells on an NVIDIA GeForce RTX 4090 (24 GB VRAM) in our scalability tests (Figure A9), well below the size of our slides, and does not natively support counterfactual donor-swap or neighbor-feature interventions.

MintFlow [Akbarnejad et al., 2025] is a flow-matching generative model with an underlying graphVAE-style encoding and count-specific decoding that disentangles single-cell gene expression into intrinsic and microenvironment-induced components. It learns three latent variables per cell (an intrinsic state and incoming/outgoing spatial signals). MintFlow supports in-domain counterfactual queries via in-silico perturbation of the tissue (e.g., relabeling or deleting cells and re-sampling from the generative model), but conditions each cell on its cell-type label and therefore cannot extrapolate to cell types or contexts unseen during training, whereas *Cellina* takes no categorical code at inference. Consequently, we provide MintFlow with all cells, including those held out for other methods during training, giving it a strictly in-domain evaluation setting that constitutes an advantage over the other benchmarked approaches. Because MintFlow does not natively support edge swapping of neighbors, we adapted its inference procedure to align it as closely as possible with *Cellina*: for each target cell, we replace its gene expression vector with that of a randomly selected control cell of the same cell type and then call `generate_insilico_ST_data()` to produce counterfactual counts (matching *Cellina*'s sampling procedure). We note that certain trained models raised an internal numerical error (`Expected parameter rate to satisfy the constraint GreaterThan(lower_bound=0.0)`) at arbitrary checkpoints; in these cases, we evaluated on the last stable checkpoint preceding the error.

SpatialProp [Sun et al., 2025] is a graph neural network model that predicts how perturbations to neighboring cells propagate to a center cell by inferring its gene expression from masked k-hop (2-hop by default) neighborhood graphs. SpatialProp is tailored to in-domain predictions, and its SparseRenorm post-processing calibrates predictions using an empirical error distribution derived from unperturbed (in-distribution) base predictions, which we omit for out-of-distribution perturbation evaluated in this work. Like *Cellina*, SpatialProp provides functionality to predict downstream effects of perturbations on the spatial microenvironment, enabling direct comparison in the neighbor node perturbation setting, via adaptation of this vignette: `https://github.com/abuendia/spatial-prop/blob/main/notebooks/api_demo.ipynb`. We also implemented a custom edge perturbation API for SpatialProp modifying its neighborhood steering function (`batch_steering_cell()`) and evaluated only on the MERFISH cohort due to the high computa-

tional cost of the CRC dataset, seeing similar results to the node perturbation experiment on the same cohort (Pearson 0.68 for edge vs. 0.66 for node perturbation).

Concert [Lin et al., 2025] predicts spatially-resolved perturbation responses by disentangling spot-level expression into learnable embeddings for basal cell state and perturbation or covariate identities, following the LORD framework [Gabbay and Hoshen, 2019, Piran et al., 2024]. Perturbation effects are propagated across the tissue via Gaussian process priors with perturbation-specific Cauchy kernels. Concert simulates perturbations by swapping embeddings for categorical attributes (e.g., perturbation identity, disease state) or interpolating learned projections of continuous attributes (e.g., time, dose). Such swaps can be applied to the cells surrounding a target and propagated inward through the learned kernel. Unlike *Cellina*, however, Concert defines no operation on the tissue graph itself: propagation is a continuous function of cell coordinates rather than of neighborhood membership, and only a subset of its latent dimensions is spatially coupled. Neither a cell's neighbors (the edges) nor their expression (the nodes) can be altered, so neither of our perturbation operators is natively available to it. Concert therefore admits only a domain identity swap (e.g. REF $\rightarrow$ CRC), which we adopt as its closest available analogue of an edge perturbation; this restriction follows from Concert's design rather than from any simplification on our part, and Concert is otherwise run at its default settings.

Celcomen [Megas et al., 2025] is a generative graph neural network model that disentangles intra- and inter-cellular gene regulation in spatial transcriptomics via a maximum-entropy formulation. The model's parameters are guaranteed to yield identifiability for the gene-gene interaction matrices. It consists of an inference module that learns intra- and inter-cellular gene-gene interaction matrices (under an acyclic regulatory assumption), and a generative simulation module that produces counterfactual spatial samples by intervening on selected nodes (e.g., gene knockouts in specific tissue locations). As such, unlike *Cellina*, Celcomen targets in-silico gene-level perturbations rather than perturbations directly on tissue graphs: its interventions act through a single tissue-wide gene–gene interaction matrix rather than on the expression of a focal cell's own neighbors, so it can be modified to approximate node perturbation only at the level of global gene regulation and offers no edge perturbation at all; its generative module (Simcomen) likewise samples tissue-wide counterfactuals rather than responses of a focal cell to its own neighbors. Our best-effort adaptation also did not scale, attempting to allocate $\sim$42.6 GiB on the smallest slide (232) from Crowell et al. [2025].

Table A3 contrasts these methods component by component.

Table A3: Comparison of related methods against the requirements of tissue graph counterfactual inference. ✓ indicates the property is satisfied; — indicates it is not. ✓* indicates partial or adapted support (see text for details). *Italicized* methods are discussed in this section but are not included in any benchmark (counterfactual, disentanglement, or scalability).

Method	Spatial graph	Disen-tanglement	Counterfactual inference	Edge pert.	Node pert.	Unseen cell type $\times$ domain
scVI	—	—	—	—	—	—
scANVI	—	—	—	—	—	—
scGen	—	—	✓	—	—	—
CPA	—	✓	✓	—	—	—
MISTy	✓	—	—	—	—	—
NCEM	✓	—	—	—	—	—
scVIVA	✓	—	—	—	—	—
SIMVI	✓	✓	—	—	—	—
MintFlow	✓	✓	✓	✓*	—	—
SpatialProp	✓	—	✓	—	✓	✓*
Concert	✓	✓	✓	✓*	—	✓
Celcomen	✓	—	✓	—	✓*	—
Cellina	✓	✓	✓	✓	✓	✓
Cellina-GAT	✓	✓	✓	✓	✓	✓

Additional notes. All competing methods are evaluated using their default parameters under the same protocol as *Cellina* (Section 4.2) with identical leave-one-cell-type-out splits, except for MintFlow as described in Appendix F where no cells are held out. Each method receives the input its

formulation supports: scGen and CPA are given only cell expression and domain labels, as in their original formulations, and are not provided neighbor composition features. SpatialProp, which takes spatial neighborhoods as input, is evaluated in *the node perturbation* task: a partial neighbor-node perturbation restricted to the top 200 genes, matching the setting used for *Cellina*$_{\text{node-pert}}$ is applied in inference.

G Extended Results

Extended Metrics for Leave-One-Cell-Type-Out Evaluation. Here, we discuss results at the cell-type level for CRC and MERFISH datasets across an extended list of metrics. As a reminder, for the colorectal cancer data, we design our experiments around the in-silico perturbation query to predict the effect of the cancer region on healthy cells (REF $\rightarrow$ CRC). We employ a leave-one-cell-type-out strategy for a comprehensive evaluation of models across samples from six patients. In CRC, *Cellina* consistently achieves the best performance across all held-out cell types on *Pearson* and *Spearman* ρ, with the exception of Epithelial, where MintFlow is ahead, likely because this is the most abundant cell type, and MintFlow observes it during training. On *Signed Precision*, *Cellina*'s variants lead all competing methods for every cell type, and on RMSE_{LFC} for every cell type except Epithelial, where MintFlow and Concert are ahead. $\text{RMSE}_{\text{counts}}$ is largely comparable across methods, with MintFlow worse; E-distance instead separates the mean shift, Concert and SpatialProp, which score several-fold higher, from the remaining methods (Figure A3). Notably, the contribution of supervision (i.e., the gain of the full *Cellina* over *Cellina (ablated)*) is largest for Epithelial ($+0.35$ Pearson) and Fibroblast ($+0.20$) cells, the two types whose predictions depend most on their neighbors (Tables A9 and A10), whereas the gain is marginal for T cells, whose state is largely intrinsic. To further assess whether *Cellina*'s advantage over competitors generalizes to other datasets, we repeat the same evaluation strategy on a whole mouse brain dataset (3 slides chosen, most abundant cell types). We use Thalamus as the control region, while Fiber-tracts and Isocortex are held out. Both settings are discussed in the main text. Similar results are seen on the MERFISH mouse brain data (Figures A4 and A5), suggesting that *Cellina* maintains the highest scores in log-fold change correlation metrics (as measured by Spearman and Pearson) as well as Signed Precision and RMSE_{LFC}, while remaining competitive in distributional metrics ($\text{RMSE}_{\text{counts}}$ and E-distance).

Both distributional metrics above are computed on log-normalized expression, which is the space scGen is trained to reconstruct, whereas the count-based methods are fit to a negative-binomial likelihood on raw counts. To represent both expression distributions, we additionally score the same predictions with NB deviance (Appendix A), which compares predicted means against observed counts under a common per-gene dispersion. Under this measure scGen's deviance is roughly twice *CPA*'s (Table A5) and roughly $3\times$ *Cellina*'s on CRC, with the gap widening even further on MERFISH, where scGen's deviance is roughly tenfold *Cellina*'s (0.33 vs. 0.03, averaged over both held-out domains). We therefore read the two metrics as complementary, since each rewards its matched likelihood family.

Table A4 extends the MERFISH evaluation of Table 2 (Section 4.3) with the results disaggregated by held-out domain, and Table A6 extends the *in vivo* Perturb-FISH evaluation of Table 3 with the KO-specific decomposition and the randomized-control comparisons discussed in Section 4.3.

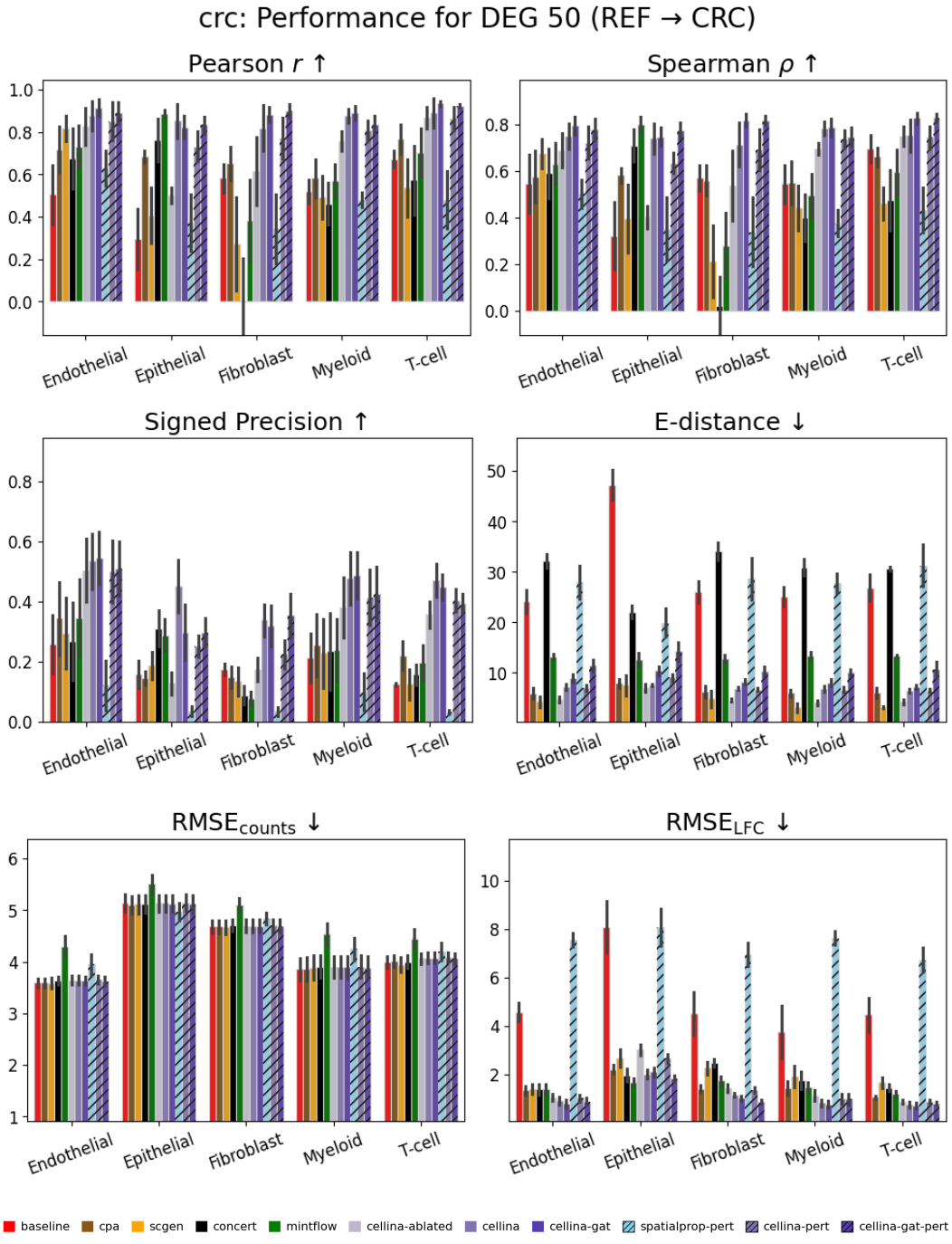

Figure A3: Colorectal cancer per-cell-type results with extended metrics for CRC held-out region. Striped bars with **-pert** in method names denote evaluation under *node perturbation* on a subset of features. The mean shift baseline is denoted as "baseline". Bar heights indicate mean over slides, error bars indicate standard error of the mean.

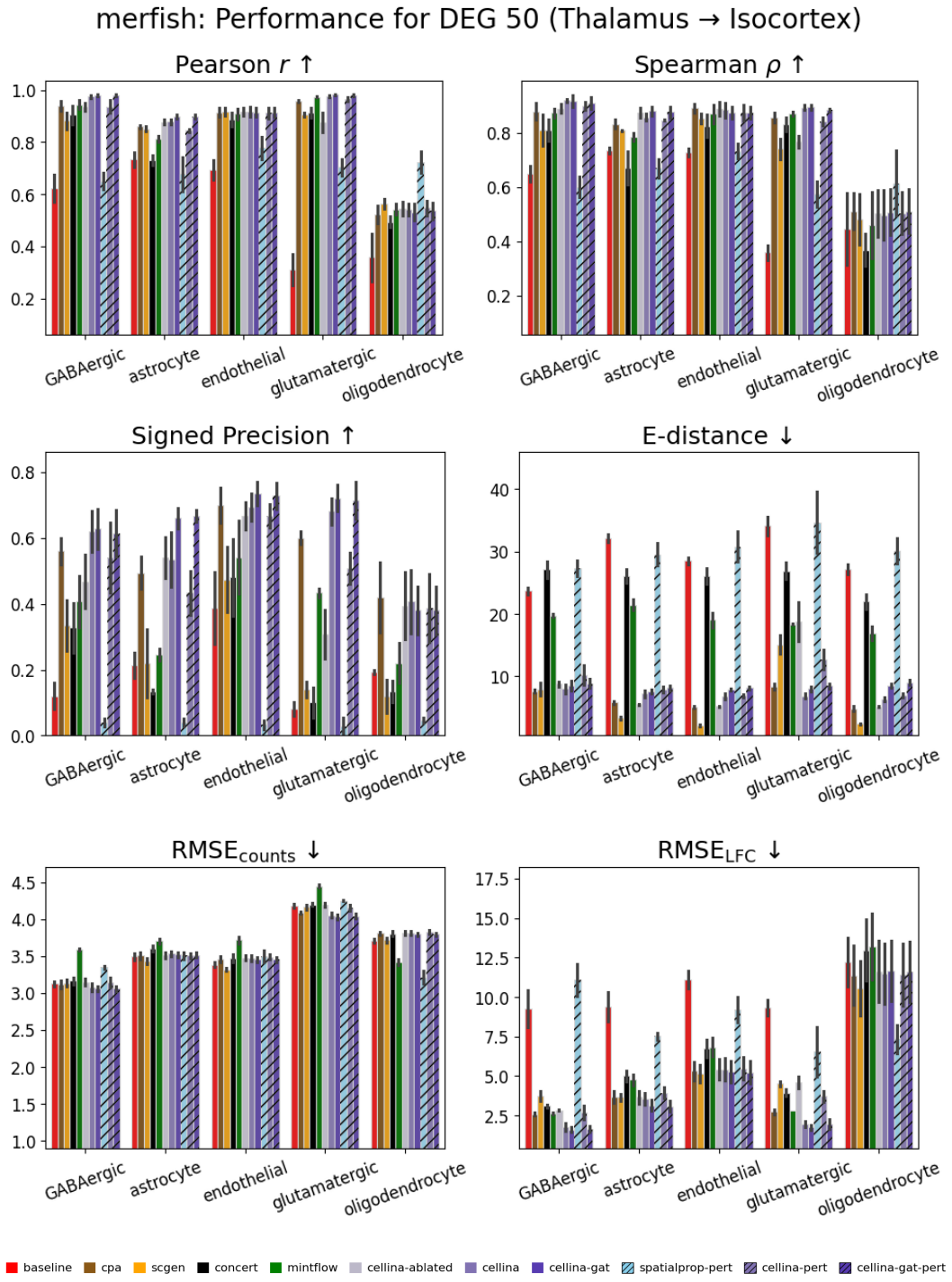

Figure A4: MERFISH per-cell-type results with extended metrics for Isocortex held-out region. Striped bars with **-pert** in method names denote evaluation under *node perturbation* on a subset of features. The mean shift baseline is denoted as "baseline". Bar heights indicate mean over slides, error bars indicate standard error of the mean.

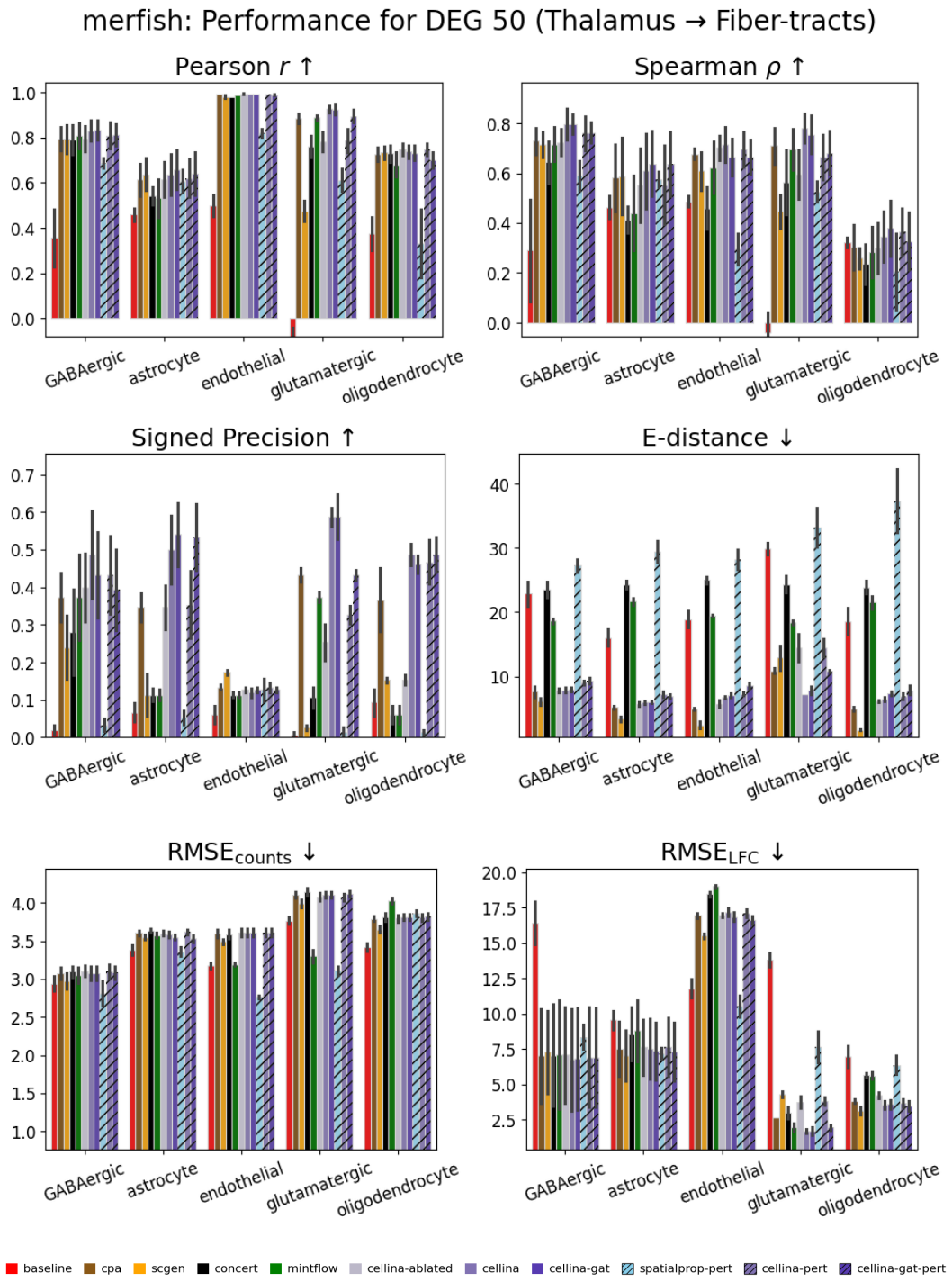

Figure A5: MERFISH per-cell-type results with extended metrics for Fiber-tracts held-out region. **-pert** denotes node perturbation models. The mean shift baseline is denoted as "baseline". Bar heights indicate mean over slides, error bars indicate standard error of the mean.

Table A4: Leave-one-cell-type-out performance (top 50 DEGs) for the in-silico perturbations Thalamus → Isocortex, Fiber-tracts (MERFISH). Mean ± std across cell types and three slides, averaged over both held-out domains (*Average*) and per held-out domain. Best per metric within each block (edge- vs. node-perturbation) in **bold**; **np** refers to the node perturbation task; *Cellina* variants in gray, spatially-informed competitors in orange. On average, *Cellina* and *Cellina-GAT* perform largely on par, and Concert scores similarly to the spatially-uninformed methods (0.18 Signed Precision against scGen's 0.20; E-distance 24.91 against the mean shift's 25.18).

Holdout	Method	Pearson ↑	$\text{Precision}_{\text{signed}}$ ↑	E-distance ↓	RMSE_{LFC} ↓
Average	Mean shift	0.43 ± 0.23	0.12 ± 0.09	25.18 ± 4.75	10.98 ± 2.78
	CPA	0.82 ± 0.16	0.44 ± 0.13	6.53 ± 1.99	6.36 ± 4.71
	scGen	0.77 ± 0.16	0.20 ± 0.14	**5.73 ± 4.84**	6.48 ± 3.97
	Concert	0.77 ± 0.16	0.18 ± 0.15	24.91 ± 2.08	7.45 ± 4.99
	MintFlow	0.81 ± 0.17	0.29 ± 0.16	19.52 ± 1.81	7.27 ± 5.43
	Cellina (ablated)	0.81 ± 0.15	0.37 ± 0.15	8.33 ± 4.76	6.78 ± 4.54
	Cellina	**0.84 ± 0.16**	0.51 ± 0.17	6.93 ± 0.83	6.08 ± 5.06
	Cellina-GAT	**0.84 ± 0.16**	**0.53 ± 0.17**	7.67 ± 0.83	**5.97 ± 5.07**
	$\text{Cellina}_{\text{np}}$	0.82 ± 0.15	0.42 ± 0.15	8.96 ± 2.92	6.65 ± 4.63
	$\text{Cellina-GAT}_{\text{np}}$	**0.83 ± 0.16**	**0.51 ± 0.16**	**8.61 ± 1.15**	**5.98 ± 4.98**
	$\text{SpatialProp}_{\text{np}}$	0.66 ± 0.14	0.04 ± 0.04	30.79 ± 5.18	8.18 ± 2.05
Fiber-tracts	Mean shift	0.31 ± 0.26	0.05 ± 0.05	21.29 ± 5.51	11.71 ± 3.64
	CPA	0.80 ± 0.15	0.33 ± 0.13	6.70 ± 2.48	7.58 ± 5.80
	scGen	0.72 ± 0.19	0.14 ± 0.10	**5.28 ± 4.44**	7.44 ± 5.02
	Concert	0.76 ± 0.16	0.13 ± 0.11	24.23 ± 1.53	8.55 ± 6.10
	MintFlow	0.78 ± 0.18	0.21 ± 0.16	19.95 ± 1.64	8.50 ± 6.59
	Cellina (ablated)	0.79 ± 0.14	0.26 ± 0.13	7.98 ± 3.75	7.93 ± 5.58
	Cellina	**0.83 ± 0.15**	**0.44 ± 0.20**	6.82 ± 0.75	7.34 ± 6.19
	Cellina-GAT	**0.83 ± 0.14**	0.43 ± 0.19	7.23 ± 0.91	**7.26 ± 6.07**
	$\text{Cellina}_{\text{np}}$	0.79 ± 0.15	0.34 ± 0.16	8.99 ± 3.11	7.85 ± 5.74
	$\text{Cellina-GAT}_{\text{np}}$	**0.81 ± 0.15**	**0.39 ± 0.18**	**8.68 ± 1.54**	**7.25 ± 5.93**
	$\text{SpatialProp}_{\text{np}}$	0.61 ± 0.20	0.05 ± 0.05	31.12 ± 5.62	8.02 ± 1.87
Isocortex	Mean shift	0.54 ± 0.20	0.20 ± 0.14	29.07 ± 3.99	10.25 ± 1.92
	CPA	0.84 ± 0.17	0.55 ± 0.13	6.35 ± 1.51	5.14 ± 3.63
	scGen	0.82 ± 0.14	0.26 ± 0.18	**6.17 ± 5.23**	5.52 ± 2.92
	Concert	0.79 ± 0.17	0.23 ± 0.18	25.60 ± 2.64	6.36 ± 3.89
	MintFlow	0.84 ± 0.16	0.37 ± 0.16	19.10 ± 1.97	6.04 ± 4.26
	Cellina (ablated)	0.83 ± 0.15	0.47 ± 0.17	8.67 ± 5.77	5.62 ± 3.51
	Cellina	**0.86 ± 0.17**	0.59 ± 0.15	7.05 ± 0.91	4.83 ± 3.94
	Cellina-GAT	**0.86 ± 0.18**	**0.62 ± 0.15**	8.10 ± 0.74	**4.68 ± 4.07**
	$\text{Cellina}_{\text{np}}$	0.84 ± 0.16	0.51 ± 0.15	8.94 ± 2.73	5.45 ± 3.51
	$\text{Cellina-GAT}_{\text{np}}$	**0.86 ± 0.17**	**0.62 ± 0.15**	**8.54 ± 0.75**	**4.71 ± 4.03**
	$\text{SpatialProp}_{\text{np}}$	0.71 ± 0.08	0.04 ± 0.02	30.45 ± 4.74	8.34 ± 2.23

Table A5: Negative-binomial deviance of the same in-silico perturbation predictions scored in count space (Appendix A), for the leave-one-cell-type-out evaluation on both cohorts. Lower is better; best per metric within each block (edge- vs. node-perturbation) in **bold**; *Cellina* variants in gray, spatially-informed competitors in orange. All methods are scored under a common per-gene dispersion estimated from the observed target population, so no method contributes its own. Mean ± std across cell type-slide experiments.

	CRC		MERFISH	
Method	NB deviance ↓	E-distance ↓	NB deviance ↓	E-distance ↓
Mean shift	0.44 ± 0.13	29.89 ± 10.49	2.78 ± 0.97	25.18 ± 4.75
CPA	0.04 ± 0.05	6.44 ± 2.27	0.05 ± 0.04	6.53 ± 1.99
scGen	0.09 ± 0.03	**4.65 ± 3.49**	0.33 ± 0.05	**5.73 ± 4.84**
MintFlow	0.50 ± 0.14	13.01 ± 1.76	2.44 ± 0.83	19.52 ± 1.81
Concert	0.05 ± 0.04	29.98 ± 5.22	0.19 ± 0.07	24.91 ± 2.08
Cellina (ablated)	0.04 ± 0.04	4.93 ± 1.66	0.08 ± 0.09	8.33 ± 4.76
Cellina	**0.03 ± 0.03**	7.00 ± 1.05	0.03 ± 0.02	6.93 ± 0.83
Cellina-GAT	**0.03 ± 0.04**	8.59 ± 1.90	**0.02 ± 0.02**	7.67 ± 0.83
$\text{Cellina}_{\text{node-pert}}$	**0.03 ± 0.04**	**7.20 ± 1.45**	0.06 ± 0.07	8.96 ± 2.92
$\text{Cellina-GAT}_{\text{node-pert}}$	**0.03 ± 0.05**	11.35 ± 3.26	**0.03 ± 0.02**	**8.61 ± 1.15**
$\text{SpatialProp}_{\text{node-pert}}$	6.60 ± 4.22	27.12 ± 8.51	9.92 ± 5.36	30.79 ± 5.18

Table A6: Extended table for the *in vivo* Perturb-FISH tumor xenograft cohort. *Full* scores the complete predicted T cell response; *KO-specific* isolates the component attributable to knockout identity. *-rand* are specificity controls in which neighbors are set to the mean of randomly drawn cancer cells (300 cells, 5 draws) instead of the KO-specific population, decoupling the predicted shift from the true intervention. Values are means over 9 KOs and ± is the std across 5 random seeds. Best per metric per setting is highlighted in **bold**; *Cellina* variants in gray, spatially-informed competitors in orange. All models drop in Pearson when the correct knockout identity is withheld (random control), confirming that the predicted shift is tied to the specific intervention.

Type	Method	Pearson ↑	$\text{Precision}_{\text{signed}}$ ↑	E-distance ↓	RMSE_{LFC} ↓
KO-specific	Cellina	0.33 ± 0.08	**0.68 ± 0.03**	4.13 ± 0.15	0.43 ± 0.01
	Cellina-GAT	**0.40 ± 0.07**	0.66 ± 0.03	4.89 ± 0.19	**0.41 ± 0.01**
	SpatialProp	0.16 ± 0.06	0.55 ± 0.06	7.44 ± 0.04	0.64 ± 0.04
	Mean shift	0.06 ± 0.06	0.51 ± 0.03	**0.13 ± 0.01**	0.46 ± 0.02
	Cellina-rand	-0.00 ± 0.05	0.48 ± 0.04	4.13 ± 0.18	0.47 ± 0.03
	Cellina-GAT-rand	0.07 ± 0.03	0.51 ± 0.03	4.88 ± 0.19	0.46 ± 0.03
	SpatialProp-rand	0.11 ± 0.05	0.52 ± 0.06	7.48 ± 0.02	0.66 ± 0.04
Full	Cellina	0.53 ± 0.05	**0.76 ± 0.03**	4.14 ± 0.16	**0.50 ± 0.02**
	Cellina-GAT	**0.55 ± 0.10**	**0.76 ± 0.05**	4.87 ± 0.22	**0.50 ± 0.03**
	SpatialProp	0.26 ± 0.16	0.61 ± 0.08	7.45 ± 0.02	0.68 ± 0.09
	Mean shift	0.51 ± 0.13	0.73 ± 0.08	**0.13 ± 0.01**	0.54 ± 0.03
	Cellina-rand	0.11 ± 0.23	0.55 ± 0.06	4.12 ± 0.18	0.59 ± 0.03
	Cellina-GAT-rand	0.14 ± 0.30	0.55 ± 0.12	4.88 ± 0.18	0.59 ± 0.03
	SpatialProp-rand	0.19 ± 0.17	0.58 ± 0.09	7.47 ± 0.02	0.70 ± 0.09

H Colorectal Cancer Application: CRC Subdomains and Pathway-Specific Perturbations

Model fitting. *Cellina* was fit in-domain to the largest CRC section (210), using the shared graph pre-processing of Appendix B ($\sigma = 100\,\mu\text{m}$, $k = 200$). For this analysis, cells were split at random; no cell type or domain was withheld. The model used 64 latent dimensions per encoder, two hidden layers, a negative-binomial likelihood with observed library size, and unit classifier and adversary weights, and was trained for at most 100 epochs (batch size 2048, learning rate 10^{-3}, early stopping on validation reconstruction loss with patience 5, best checkpoint retained). Both latents were then extracted for every cell.

Subdomain identification. Cells whose histopathological annotation is CRC (all cell types) were clustered with Hotspot [DeTomaso and Yosef, 2021] over the spatial latent s: an unweighted k-nearest-neighbor graph ($k = 30$) in s, an autocorrelation test on raw counts under a depth-adjusted negative-binomial null (top $1{,}200$ genes at FDR < 0.05), and module detection from the pairwise local correlations among those genes (core genes only, ≥ 100 genes per module, FDR < 0.05). This yields three modules; each CRC-domain cell is assigned to its highest-scoring module.

Pathway annotation. A module $\times$ gene matrix was assembled from the Hotspot local-correlation coefficients of module-assigned genes and scored against PROGENy [Schubert et al., 2018] and MSigDB Hallmark gene sets using the univariate linear model of `decoupler` [Badia-i Mompel et al., 2022]. The two modules with the most distinct pathway profiles, denoted CRC1 (TGFβ-dominant) and CRC2 (NFκB/MAPK-dominant), are those analyzed in Section 4.4; the third is not examined further.

Subdomain-conditioned edge perturbations. For each of the five cell types, control cells are the cells of that type outside the CRC domain, and their neighborhoods are rewired (Definition 1) to donors drawn either from the whole CRC domain or from a single subdomain, with predictions returned at a library size of 10^4. Predictions are scored by the correlation between predicted and observed $\log_2$ fold changes over the 200 genes with the largest observed $|\log_2 \text{FC}|$. The observed reference is the pseudobulk of all cells residing in the CRC domain (global arm) or in the subdomain (subdomain arm); both arms therefore share the same control population and reference construction, and differ only in the neighborhoods on which the prediction is conditioned.

Pathway-guided node perturbations. Alteration vectors δ_g were taken directly from PROGENy weights ($|\text{weight}| > 0.5$, clipped to ± 5), restricted to genes with pseudobulk log fold change between the subdomain and the control region ($|\log \text{FC}| > 0.01$): TGFβ for CRC1 and NFκB for CRC2. The shift is applied to the log-normalized expression of every neighbor node on fixed topology (Definition 2) and scored as above over the top 100 genes; Figure 3(d) highlights the 50 most affected genes.

I Seed Stability

To assess the stability of the evaluations across random seeds, we conduct a multi-seed analysis on the edge perturbation task of Tables 1 and 2. The goal of this analysis is to evaluate whether *Cellina*'s performance advantages are random (as the result of a lucky seed value) or systematic (consistent over many seed values). We compare *Cellina* to CPA and scGen, training each on three different seeds per slide per cell type. As seen in Table A7, *Cellina* maintains its performance advantage over competitors and the variance across seeds is negligible. The *in vivo* Perturb-FISH evaluation (Tables 3 and A6) is likewise reported as the mean over five random seeds for every method, since we observed seed-to-seed variation in that smaller dataset.

Table A7: Evaluation metrics on random seeds. Mean ± standard deviation is reported across three random seeds for each model; means are therefore averaged over seeds and differ from the corresponding single-run values in Tables 1 and 2. *Cellina* in gray. Low variance indicates seed selection does not contribute to model performance.

Dataset	Method	Pearson ↑	Spearman ↑	$\text{Precision}_{\text{signed}}$ ↑	RMSE_{LFC} ↓
CRC	Cellina	0.87 ± 0.01	0.76 ± 0.01	0.45 ± 0.01	1.13 ± 0.01
	CPA	0.69 ± 0.02	0.59 ± 0.02	0.23 ± 0.01	1.49 ± 0.02
	scGen	0.50 ± 0.01	0.43 ± 0.01	0.19 ± 0.00	2.09 ± 0.09
MERFISH	Cellina	0.84 ± 0.00	0.73 ± 0.00	0.52 ± 0.01	6.08 ± 0.04
	CPA	0.82 ± 0.00	0.70 ± 0.01	0.44 ± 0.00	6.36 ± 0.01
	scGen	0.77 ± 0.01	0.63 ± 0.00	0.21 ± 0.02	6.47 ± 0.02

J Cross-Patient Generalization

All CRC evaluations reported in Section 4.3 are within-patient: focal cells and donor neighborhoods are drawn from the same tissue slide (Appendix B.6). This design removes patient-level variation from the comparison, but it does not answer whether the learned relationship between neighborhood and expression generalizes to unseen patients. We therefore repeat the edge-perturbation evaluation under a leave-one-patient-out split.

Protocol. We run the evaluation on all six CRC slides. We hold out one patient at a time and, on the remaining five, additionally hold out the target cell type y in the CRC domain exactly as in Section 4.1: cells of type y in the target domain are excluded from the donor neighborhoods and from the neighborhood graph W at both training and inference. The model is fit on the five remaining patients and applied to REF_y cells of the held-out patient, whose neighborhoods are rewired following the donor-sampling convention of Appendix B.6; predictions are scored against the observed CRC_y cells. Consequently *neither the patient nor the cell-type context was seen during training*, whereas the within-slide evaluation withholds only the latter. Crossing six held-out patients with five held-out cell types gives thirty folds per method; Table A8 reports the mean and standard deviation over folds.

The thirty-fold protocol multiplies the cost of an evaluation that is already the most expensive in this work (Appendix C.1), which is why it is restricted to three efficient models: the mean shift, which sees only the average REF → CRC change; CPA, the strongest non-*Cellina* baseline on Tables 1 and 2, which sees the domain as a label but never a cell's own neighbors; and *Cellina*, which sees the neighborhood itself.

Table A8: Cross-patient edge perturbation on the CRC cohort (top 50 DEGs). One patient and one cell type are held out simultaneously, thirty folds per method (six patients and five cell types); mean ± std over folds. Best per metric in **bold**; *Cellina* in gray. Margins between methods can be compared with Table 1 (same slides and cell types, within-patient split); absolute Pearson is lower here because the split is harder.

Method	Pearson ↑	$\text{Precision}_{\text{signed}}$ ↑	E-distance ↓	RMSE_{LFC} ↓
Mean shift	0.45 ± 0.24	0.15 ± 0.16	30.98 ± 10.34	4.85 ± 2.55
CPA	0.52 ± 0.18	0.12 ± 0.20	15.05 ± 4.22	2.88 ± 0.76
Cellina	$\mathbf{0.79 \pm 0.13}$	$\mathbf{0.30 \pm 0.20}$	$\mathbf{6.55 \pm 1.18}$	$\mathbf{1.37 \pm 0.68}$

Results. Absolute Pearson is lower here than in Table 1 for every method: *Cellina* 0.79 against 0.86 Pearson, CPA 0.52 against 0.68, the mean shift 0.45 against 0.51. Nevertheless, *Cellina* leads CPA by +0.27 Pearson here ($p = 1.23 \times 10^{-7}$, Wilcoxon signed-rank test with Holm-Bonferroni correction) against +0.18 within-slide, and by +0.18 Signed Precision ($p = 3.5 \times 10^{-6}$) against +0.23 within-slide. Notably, *Cellina* also leads on E-distance (6.55 against CPA's 15.05), the one metric on which CPA is ahead within-slide (6.44 against 7.00). These results indicate that the spatial latent s encodes transferable signal driven by neighborhood composition and expression rather than patient-specific confounding.

K Neighborhood Composition and Size

The subdomain analysis of Section 4.4 contrasts how a focal cell could differ across two cancer neighborhoods. Here, we vary the neighborhoods by i) their composition (cell types) as well as by ii) the number of cells the neighborhood graph admits (each via edge perturbation).

Neighbor composition. Because an edge perturbation can simulate the effect of any neighborhood, it can construct neighbor compositions that never occur in the tissue. Following the leave-one-cell-type-out protocol of Section 4.1, we rewire each held-out cell type to a neighborhood drawn entirely from one CRC cell type at a time (homotypic neighborhood), including its own, in which case the held-out cells themselves serve as neighbors at inference; the main evaluation instead excludes same-type neighbors, so as not to give *Cellina* an unfair advantage. All other elements of the protocol are unchanged: the held-out population is excluded from the neighborhood graph W during training, and predictions are scored against the observed CRC cells of that type.

We see that predictions depend on neighbor identity most strongly for Fibroblast and Epithelial cells (a spread of 0.24 and 0.22 across neighbor types), followed by Myeloid cells (0.12), and least for T and Endothelial cells (0.07 and 0.05; row-wise read of Table A9). Notably, homotypic draws, where the held-out cell type and the neighbor cell type are the same, are not systematically better or worse than heterotypic ones. We see that they are the best draw for Fibroblasts, the worst for T cells, which suggests that *Cellina* does not just pass the effect of neighbors to the focal cell. If *Cellina* merely passed the neighbors' expression through to the focal cell, the homotypic neighborhood would be the easiest case. This is best contradicted by an Epithelial holdout which scores 0.60 among Myeloid cells, yet only 0.44 among Epithelial cells. The model therefore predicts a distinct response to each neighbor type. Similarly, reading the table column-wise, we see that the results also differ across focal cell types. If the predictions were influenced by the neighborhood alone, all five focal types placed among Fibroblasts (the Fibroblast column) would score alike. Together, these results suggest that *Cellina*'s predictions depend on both the identity of the focal cell and its neighborhood.

Table A9: Neighborhood composition sweep on the CRC cohort (Pearson r on top-50 DEG log fold changes). Each held-out cell type (rows) is rewired to a neighborhood drawn entirely from a single CRC cell type (columns); mean over slides 231, 232 and 242. Homotypic draws lie on the shaded diagonal. Along a row the focal cell type is fixed and its neighborhood varied, including pairings that do not occur in the tissue; along a column the neighborhood is fixed and the focal cell type varied.

	Neighbor cell type				
Held-out cell type	Endothelial	Epithelial	Fibroblast	Myeloid	T cell
Endothelial	0.88	0.89	0.90	0.89	0.85
Epithelial	0.45	0.44	0.58	0.60	0.38
Fibroblast	0.46	0.50	0.66	0.65	0.42
Myeloid	0.75	0.87	0.81	0.78	0.75
T cell	0.86	0.90	0.87	0.88	0.83

Neighborhood size. The weighted adjacency matrix W determines how many cells count as neighbors, set via k (nearest neighbors) (Appendix B). We therefore refit *Cellina* at $k \in \{10, 100, 200, 1000, 10000\}$ with the kernel bandwidth fixed to ∞, so that k alone determines the neighborhood, and repeat the leave-one-cell-type-out evaluation at each setting with the held-out population masked from both the adjacency graph and the spatial aggregation. Because the model is refit at every setting, the sweep is run on a single slide (CRC 232) with five random seeds per setting.

We observe that sensitivity to k is cell-type-dependent (Table A10). T cells are essentially unaffected across three orders of magnitude (0.97 to 0.94), suggesting that their state is defined largely intrinsically. Endothelial and Myeloid cells show a modest degradation, while Fibroblast and Epithelial predictions strongly depend on k throughout and collapse at $k = 10^4$ (0.10 and 0.11), where every neighborhood converges on the domain average and carries no local signal.

Table A10: Sensitivity to neighborhood-graph construction (Pearson r on top-50 DEG log fold changes). The number of nearest neighbors k retained in W is varied with the kernel bandwidth fixed to ∞ and the model refit at each setting; $k = 200$ is the value used for *Cellina* on CRC and MERFISH (Appendix B). CRC slide 232; mean $\pm$ std over five random seeds. Neither the absolute values nor the ordering across cell types are directly comparable to Table 1 and Figure A3, which average over all six slides under the default Gaussian kernel ($\sigma = 100\,\mu$m, $\tau = 0.1$).

k	Endothelial	Epithelial	Fibroblast	Myeloid	T cell
10	0.85 ± 0.06	0.49 ± 0.12	0.52 ± 0.15	0.89 ± 0.03	0.97 ± 0.01
100	0.83 ± 0.04	0.48 ± 0.11	0.71 ± 0.14	0.87 ± 0.02	0.96 ± 0.01
200	0.83 ± 0.04	0.41 ± 0.04	0.65 ± 0.16	0.87 ± 0.02	0.96 ± 0.01
1000	0.79 ± 0.04	0.40 ± 0.22	0.54 ± 0.18	0.85 ± 0.01	0.96 ± 0.01
10000	0.68 ± 0.04	0.11 ± 0.17	0.10 ± 0.24	0.77 ± 0.07	0.94 ± 0.01

L Node Perturbation as a Function of the Perturbed Neighbor Fraction

The gene-set sweep of Appendix B.7 varies how many genes are shifted in every neighbor. Here we instead vary the proportion of neighbors perturbed by the same δ shift. We reuse the $k = 200$ checkpoints of the neighborhood-size sweep (Appendix K), one per held-out cell type on CRC slide 232, without any retraining. The perturbation is the cell-type-specific shift of Appendix B.7 over the top-200 genes per type, with the held-out type y excluded from the pseudobulk and assigned the global shift $\delta_{\setminus y}$ as before. The only change is that the shift is applied to a random fraction $f \in \{0.1, 0.25, 0.5, 0.75, 1\}$ of cells, the remainder keeping their observed expression, so that after aggregation each held-out REF cell has only a fraction f of its neighborhood perturbed. We report the Pearson correlation between observed and predicted log fold changes over the top-50 DEGs, mean $\pm$ std over five seeds.

We see that prediction accuracy rises largely monotonically; from $f = 0.1$, where only a handful of neighbors are shifted, toward each type's ceiling at $f = 1$, for every cell type except T cells, which are already at ceiling at $f = 0.1$ (Table A11). The ceiling ordering (T cell $>$ Myeloid $>$ Endothelial $>$ Fibroblast $>$ Epithelial) mirrors the neighborhood-size sweep (Table A10), and the two most neighborhood-dependent types, Fibroblast and Epithelial, are also the ones whose response is most strongly associated with f. *Cellina* thus shows a dose-dependent response across most cell types.

Table A11: Node-perturbation dose response (Pearson r on top-50 DEG log fold changes). The cell-type-specific logFC shift is applied to a random fraction f of neighbor cells before neighborhood aggregation, using the $k = 200$ checkpoints of Table A10 with each cell type held out. CRC slide 232; mean $\pm$ std over five random seeds.

f	Endothelial	Epithelial	Fibroblast	Myeloid	T cell
0.10	0.65 ± 0.06	0.02 ± 0.37	0.32 ± 0.23	0.78 ± 0.03	0.95 ± 0.01
0.25	0.68 ± 0.05	0.05 ± 0.38	0.39 ± 0.22	0.79 ± 0.03	0.95 ± 0.01
0.50	0.72 ± 0.05	0.14 ± 0.35	0.50 ± 0.19	0.81 ± 0.03	0.95 ± 0.01
0.75	0.77 ± 0.05	0.32 ± 0.15	0.62 ± 0.15	0.84 ± 0.02	0.96 ± 0.01
1.00	0.80 ± 0.05	0.43 ± 0.04	0.69 ± 0.13	0.87 ± 0.02	0.96 ± 0.01

M Disentanglement Benchmark

One of our core claims in this work is that supervised disentanglement improves perturbation simulations on graph-structured data. To demonstrate that *Cellina*'s latent factors z and s accordingly absorb cell type and spatial domain information, respectively, we use the single-cell integration benchmark (scIB) package [Luecken et al., 2022] – a comprehensive benchmarking tool for latent factors of single-cell data. Given a categorical label, such as cell type, scIB assigns an aggregate "Bio Conservation" score summarizing multiple clustering metrics such as K-means Normalized Mutual Information (NMI), K-means Adjusted Rand Index (ARI) and Silhouette score. We use the CRC cohort to assess both cell type and spatial domain (niche) conservation (Figure A6). As points of comparison, we take standard single-cell methods such as scVI [Lopez et al., 2018], scANVI

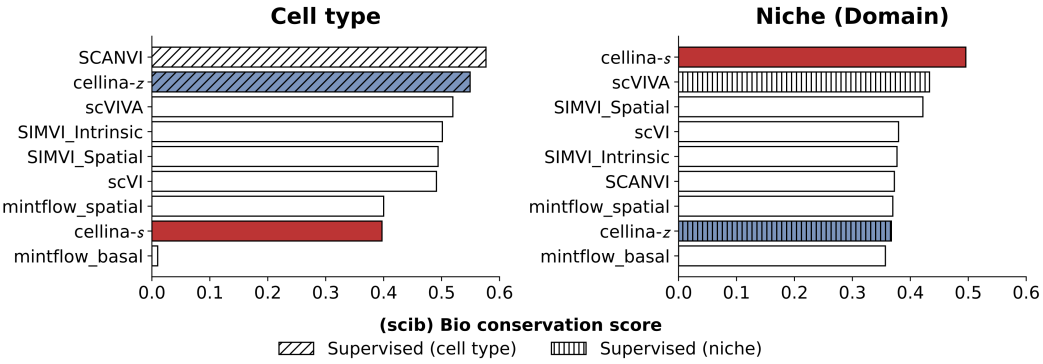

Figure A6: scIB disentanglement benchmark across methods. Bio-conservation score is an aggregate measure of clustering with respect to given label (cell type or spatial domain). *Cellina* shows strong competitive performance across common disentanglement metrics.

[Xu et al., 2021] and scVIVA [Levy et al., 2025] (details in Appendix F). Additionally, we compare to MintFlow [Akbarnejad et al., 2025] and SIMVI [Dong et al., 2025], which are designed for disentanglement of spatial and non-spatial variation in single cells. Each model is trained once on the entirety of CRC data (all six slides together) and aggregate scores for each label are reported on the training set, except for SIMVI, which does not scale to millions of cells (see Section 4 and Appendix F). For SIMVI, we take coherent regions of two slides (231, 242) containing every domain label present in the cohort, for a total of 40K cells. As SIMVI is evaluated on a smaller and distinct subset, this comparison is not directly controlled; we include it as the best available evidence given its scalability constraints. We observe that not only do *Cellina*'s latent z and s encode the desired source of variation, but s also contains less nuisance cell-type information (0.40 vs. 0.49 for SIMVI's spatial latent), while z's niche score is on par with non-spatial models such as scVI. The advantage over SIMVI and MintFlow is particularly notable as all three methods are spatially-informed models designed for this task. Our results suggest that *Cellina*'s explicitly supervised disentanglement benefits each latent space as opposed to the implicit or unsupervised disentanglement strategies adopted by MintFlow and SIMVI, respectively.

Finally, because *Cellina* anchors z with a cell-type classifier, high cell-type conservation could in principle be achieved by collapsing cells onto their label centroids – a failure mode that label-based integration metrics reward rather than detect [Wang et al., 2026]. As a sanity check, we therefore also scored *Cellina*'s z with the scGraph metrics of Wang et al. [2026], which ask whether an embedding preserves smooth transitions of cell type populations rather than separating them into well-formed but arbitrarily placed clusters or so-called "floating islands." *Cellina*'s z scores far above the CPA latent of Table 1 (0.51 ± 0.14 against 0.00 ± 0.22 across the CRC cohort), indicating that its intrinsic latent reflects retained biological structure rather than label collapse. This is further supported by plotting UMAPs of three slides (231, 232, 242) as examples in which *Cellina*'s latents show no evidence of floating islands (Figures A7, A8).

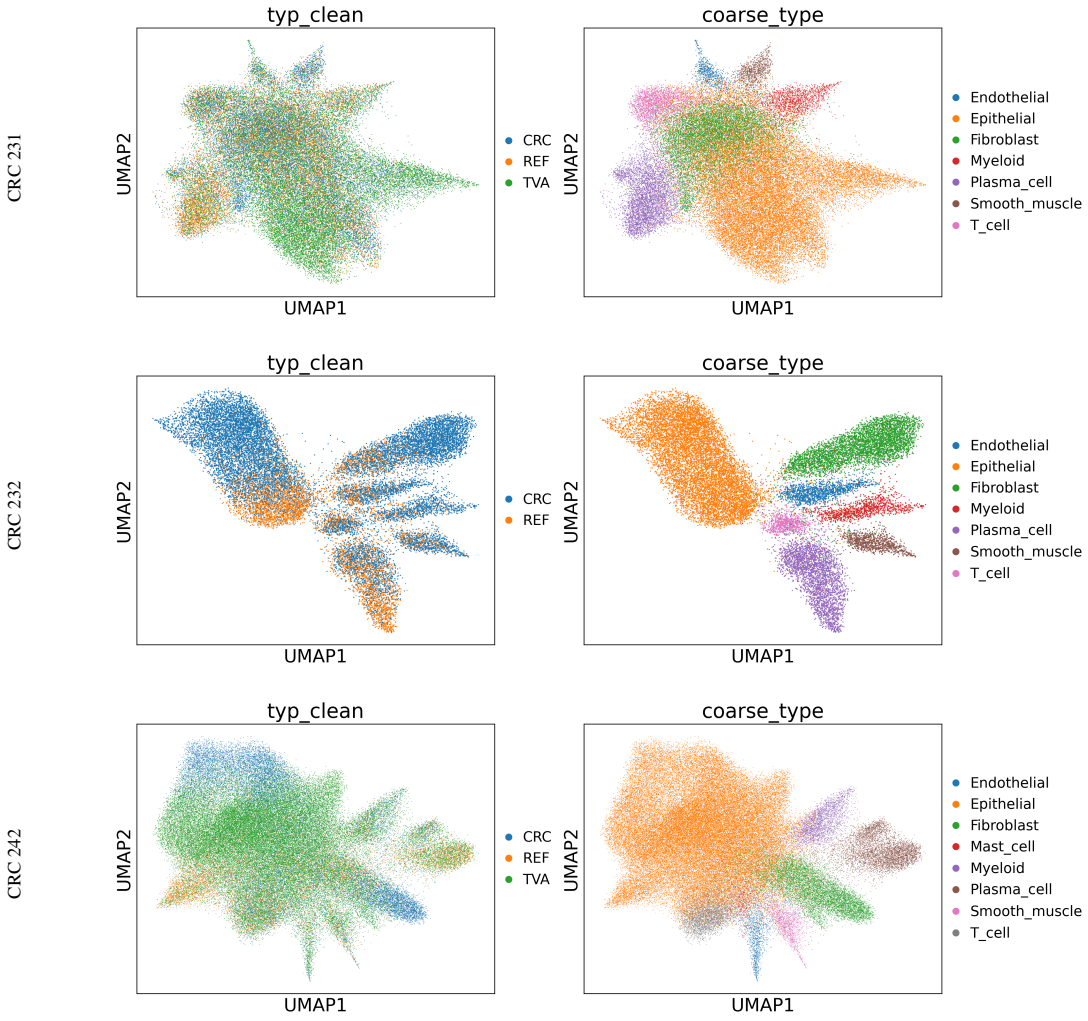

Figure A7: UMAPs of *Cellina* latent space z for each donor used in the scGraph analysis of Appendix M, colored by spatial region (left; TVA: tubulovillous adenoma) and cell type (right).

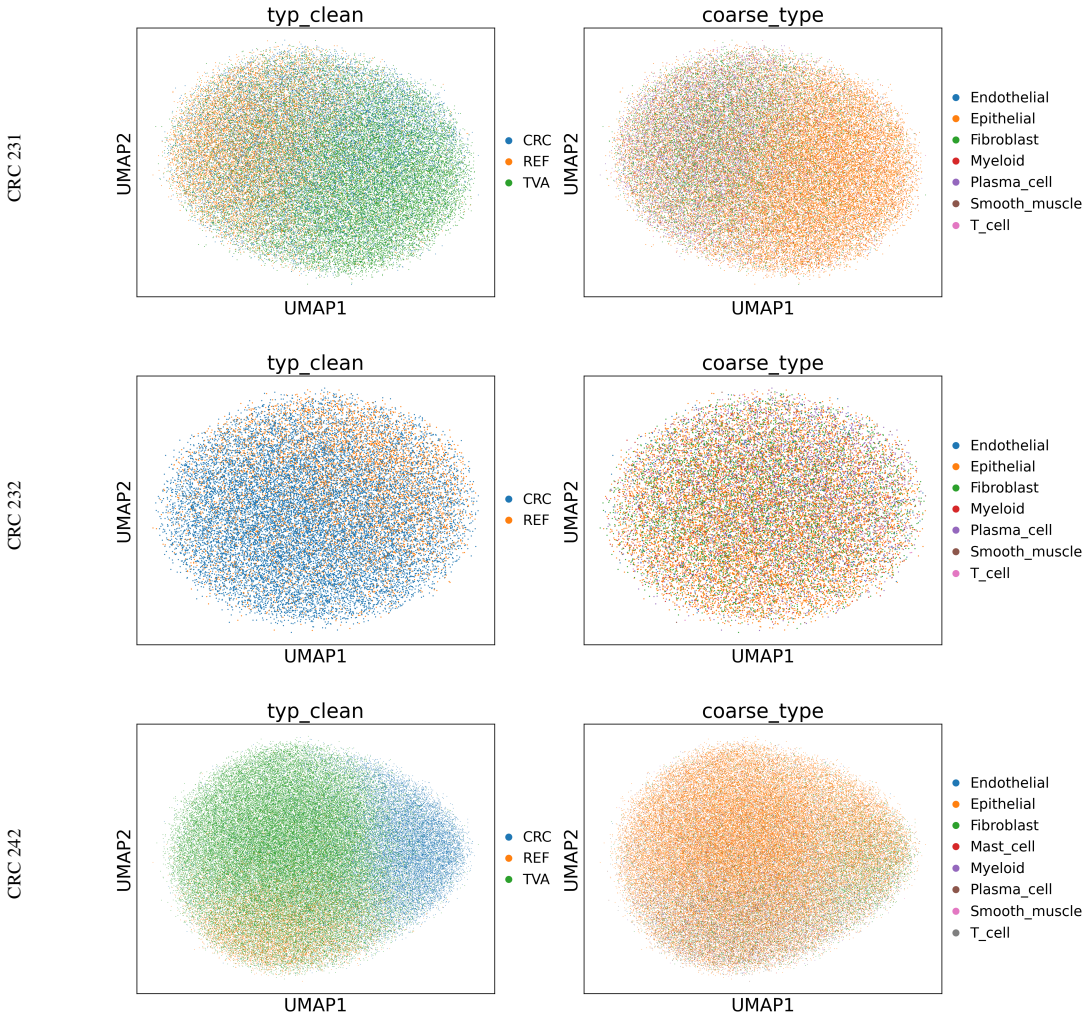

Figure A8: UMAPs of *Cellina* latent space s for each donor used in the scGraph analysis of Appendix M, colored by spatial region (left; TVA: tubulovillous adenoma) and cell type (right).

N Scalability Benchmark

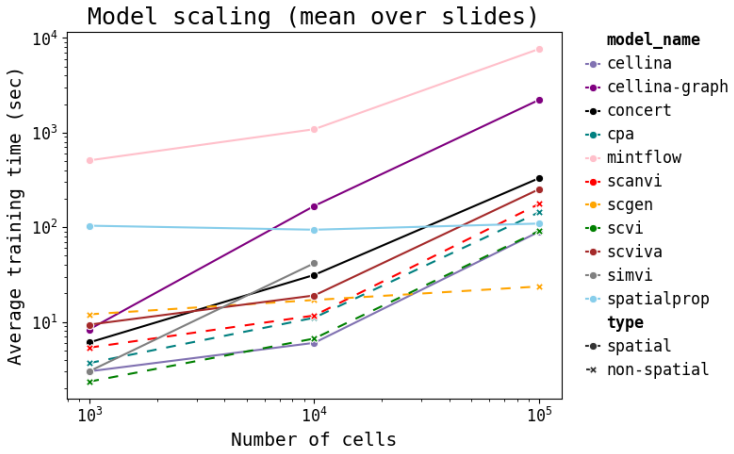

Figure A9: Training time (wall clock, seconds) vs. dataset size (number of cells) for *Cellina* and related methods. *Cellina* is one of the fastest, remaining competitive with scVI and substantially faster than spatial competitors (SIMVI, scVIVA, MintFlow, Concert). SpatialProp's training time remains stable due to efficient cell-type-based subsampling, while *Cellina-GAT*'s explicit message passing adds a computational penalty over *Cellina*, yet it remains faster than MintFlow. In the legend, cellina-graph denotes *Cellina-GAT*.

Training models on emerging spatial datasets which capture millions of cells can incur major costs in user wall-clock time. It is therefore important that models are not only relatively quick to train, but also able to scale as the number of cells increases. We assess the scalability of all models shown in this work and report findings on three CRC slides (221, 242, 232) with subsampling to 10^3, 10^4 and 10^5 cells (Figure A9). For a fair comparison, we set the same batch size and number of epochs for each model and only compare the train loop (without pre-processing workloads). *Cellina* comes out as one of the fastest-to-train models in the benchmark suite, owing to the efficient pseudobulk-based $\varphi(v)$ computed a priori, omitting the expensive message passing from model training over graphs. Training then proceeds with the same complexity as standard scVI with no graph-structure overhead at training time. This contrasts with GAT-based methods (including *Cellina-GAT*, MintFlow, and SIMVI) that require subgraph sampling during training.

All methods were trained and evaluated on the same GPU machine with NVIDIA GeForce RTX 4090 GPUs, AMD Ryzen Threadripper PRO 7975WX 32-Cores, and 500GB of RAM.

O Edge vs. Node Perturbations

Three properties of neighborhood topology. A focal cell's neighborhood topology is characterized by three properties: *proximity* (how close each neighbor is), *cardinality* (how many neighbors there are), and *composition* (which cells occupy those positions). An edge perturbation (Definition 1) intervenes on $(\mathcal{E}, W)$ and can therefore act on all three; a node perturbation (Definition 2) acts on none of them, since it holds the topology fixed and rewrites only the features X of the neighbors already present. In other words, an edge perturbation modifies the connections between cells, whereas a node perturbation modifies the features of the cells themselves. As such, the two operators are complementary, and coincide only in the narrow case described below.

Operations available only to edge perturbation. Any change to cardinality, proximity, or composition. For example, our edge perturbation is exactly such an intervention at inference: every original edge on the focal cell is dropped and replaced by the observed neighborhood of a same-type cell in the target domain, so both composition and cardinality are set by construction (Appendix B.6). Consistent with this, predictions depend on the neighborhood size used to construct W and degrade once k reaches the slide scale, where each neighborhood converges on the domain average (Appendix K). Edge perturbation also permits neighborhoods that never occur in the tissue — e.g., a fully homotypic one (Appendix K), or by extension any desired composition and size. None of

these are expressible as node perturbations. Experimentally, an edge perturbation is the in-silico counterpart of rewiring a cell's surroundings, as in a co-culture, an organoid transplant, or a xenograft; our REF→CRC evaluation represents one such scenario, asking what a healthy colon cell would express once its neighbors are replaced by tumor tissue. The same operator is also implicitly used by MintFlow [Akbarnejad et al., 2025], which the authors use to deplete regulatory T cells from lesional atopic dermatitis skin, and tertiary-lymphoid-structure macrophages from clear-cell renal cell carcinoma. Similarly, removing a cell type, e.g., with a drug that kills it, in our formalization, represents an edge perturbation, since the killed cells (nodes) are removed from the graph altogether, and hence disconnected from any focal cell rather than having their expression altered.

Operations available only to node perturbation. On a fixed topology, altering a chosen gene set $\mathcal{S}$ of the already connected neighbors and nothing else: the pathway shift of Figure 3(d), a single-ligand stimulation, or a double knockout in a neighbor. Edge perturbation cannot express these, because it can only connect the focal cell to, or disconnect it from, cells (typically) present in the data. SpatialProp's applications focus on this operator [Sun et al., 2025], rescaling a chosen gene set across the neighbors while leaving the graph untouched. Specifically, their causal benchmark exploits that fixed topology to test response directionality: for a set of mediators, such as metabolites, lipids, ions, cytokines, and hormones, they perturb the neighbors' genes associated with *producing* the mediator and ask whether the focal cells' *response* matches known pathways.

Where the two operations coincide. The two operators agree only when (i) the original and target neighborhoods have the same proximity and size, so that nothing needs rewiring, (ii) every neighbor's entire feature vector is overwritten, and (iii) each neighbor is overwritten to match an observed or simulated cell. Our node perturbations do not satisfy (iii), since δ shifts neighbors toward a population-average profile rather than onto observed cells. Consistently, node perturbation saturates below the edge-perturbation reference for every gene set (Appendix B.7), and in the held-out benchmarks it trails edge perturbation for *Cellina*, with a smaller gap for *Cellina-GAT* (Tables 1 and 2).

Beyond tissue graphs. The two operator definitions require only a graph whose nodes carry features; as such, both operators transfer to other relational data. In a hospital contact network such as that of Vanhems et al. [2013], extended so that each individual carries a measured readout (e.g., blood screen values or an omics profile) alongside their role, an edge perturbation asks how that readout would change if their contacts were those of a peer in another ward. In a labeled collaboration graph, where each person carries a set of skills, a node perturbation asks how an individual's own skills would develop if their collaborators' skills changed. However, while the pair of operators together with the intrinsic/extrinsic decomposition are in theory transferable, *Cellina* as implemented is specifically targeted to spatial omics data. Specifically, cell-type and spatial-domain labels y_v and d_v are required, by default, its encoder aggregates neighbors through a Gaussian kernel over spatial coordinates, assuming a diffusion gradient, and its decoder models integer counts with a negative-binomial likelihood.

Table A12: Every perturbation experiment in this work and the operator it instantiates. Edge perturbations rewire $\mathcal{N}(v)$; node perturbations rewrite the features of a fixed $\mathcal{N}(v)$.

Experiment	Reference	Operator
CRC leave-one-cell-type-out (REF → CRC)	Table 1, edge block	Edge
Mouse brain leave-one-cell-type-out	Table 2, edge block	Edge
Held-out Fibroblast IGF2 illustration	Figure 2	Edge
Subdomain-conditioned rewiring (CRC1/CRC2)	Figure 3(a,b)	Edge
Cross-patient generalization	Table A8	Edge
Neighborhood composition sweep	Table A9	Edge
CRC and mouse brain node perturbation ($\delta_{y,g}$)	Tables 1, 2, node blocks	Node
Perturbed gene-set sweep	Figure A1	Node
Perturbed neighbor-fraction sweep	Table A11	Node
Pathway-guided neighbor perturbation	Figure 3(d)	Node
In vivo CRISPR xenograft (Perturb-FISH)	Table 3	Node

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction define querying counterfactuals on tissue graphs and propose a method using supervised disentanglement. Definitions are provided in Section 3. We claim improved performance on predicting unseen cell types in alternative domains and validate this experimentally in Section 4 and the Appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed along with future work in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper is primarily empirical. While we introduce formal definitions of tissue graph counterfactuals in the context of tissue graphs (Section 3.3), this is a conceptual formalism rather than a theoretical result requiring proofs. All claims are validated experimentally.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide full code for the proposed method, reproducible environments, and analysis notebooks and scripts, alongside dataset links. All experiments are described in detail in text, covering hyperparameters, evaluation metrics, and dataset splits.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: *Cellina* is released under the BSD 3-Clause License at `https://github.com/PMBio/cellina`, with tutorials at `https://cellina.readthedocs.io`. Scripts and configurations to download all data and reproduce every experiment are at `https://github.com/PMBio/cellina-reproducibility` (Appendix B). All datasets used are publicly available, and we provide scripts for data access and preprocessing.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and evaluation details are reported in Section 4 and in the Appendix, including data splits, in-silico perturbation settings, hyperparameters, optimizer choice, and selection criteria.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All benchmark tables report mean $\pm$ standard deviation, with the source of variation (cell types, patient samples or slides, random seeds) stated in each caption, and figure error bars are defined in their captions (Section 4 and the Appendix). For the

cross-patient comparison, we additionally report paired Wilcoxon signed-rank tests with Holm-Bonferroni correction (Appendix J); no other formal significance testing is performed.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources, including GPU type, memory, and wall-clock runtimes per experiment, are reported in Appendix C.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our work complies with it in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper does not include a dedicated societal-impact discussion. This work is a conceptual and methodological contribution to spatial single-cell modeling; the expected scientific impact on the field is discussed in Section 5. It does not introduce a deployed system or decision-making tool with immediate societal consequences; any impact would be indirect and realized through downstream biological and biomedical research applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not include any assets that have dual-use risk. All datasets used in our work are publicly available.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and third-party methods are credited through citations to their original publications. Datasets (Appendix B): the colorectal cancer cohort of Crowell et al. [2025] (Zenodo record 15574384), the MERFISH whole mouse brain atlas of Zhang et al. [2023] (CZI CELLxGENE portal) and the Perturb-FISH melanoma xenograft of Binan et al. [2025] are used under their respective licenses. Baselines (scVI, scANVI, scGen, CPA, scVIVA, SIMVI, MintFlow, SpatialProp, Concert, Celcomen) were run from the official repositories of their authors, each under its respective open-source license, whose terms we respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The code introduced in this work is documented with function-level docstrings and follows the API design conventions of scvi-tools [Gayoso et al., 2022] and scverse [Virshup et al., 2023], which facilitates consistency with existing tools in the single-cell analysis ecosystem. In addition, we provide example usage (e.g., scripts, notebooks, config files) to support reproducibility. Our code uses the BSD 3-Clause License.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our work uses only publicly available data: anonymized human single-cell data [Crowell et al., 2025], a mouse brain atlas [Zhang et al., 2023], and a melanoma xenograft screen [Binan et al., 2025]. We did not collect any new data.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our work uses only publicly available, anonymized data [Crowell et al., 2025, Zhang et al., 2023, Binan et al., 2025]. We did not collect any new data; the datasets were released by their original authors in compliance with applicable ethical guidelines and approvals. Therefore, this work does not constitute human subjects research requiring additional IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core research in the paper does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.