PERTURBATION MODELING VIA SIGNED PROPAGATION AND LOCALIZED CONDITIONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell perturbation modeling is formulated as conditional generation of a perturbed cell population. Recent models use gene co-expression graphs to pretrain static gene embeddings or to mask dense attention backbones, which score every gene pair. The response direction among gene pairs is an important factor in perturbation modeling, and graphs built from absolute correlation values discard the sign that sets response direction. We present **SPARC**, an attention-free flow matching velocity field in which a signed graph represents the gene-mixing operator. Genes exchange information through separate positive and negative edges, normalized independently and gated at every layer by a cell state. A shared recurrent block extends propagation depth at the parameter cost of one block. Every perturbed genetic target gets a conditioning vector and is kept in every training batch. Evaluated on genetic, drug, and cytokine benchmarks, SPARC improves Spearman correlation of log-fold changes over the strongest baseline in all settings by up to 0.31, and leads in direction match by up to 0.039. Generated cells also recover more known transcription factor-target pairs and perturbed pathways than the strongest baseline.

1 INTRODUCTION

Modeling cell response to gene knockouts and drug treatments is an important task in functional genomics and drug discovery (Qi et al., 2024; Lotfollahi et al., 2023). CRISPR screens and chemical profiling allow us to measure hundreds of such conditions, but the intervention space expands combinatorially with number of individual perturbations, making exhaustive coverage infeasible (Roohani et al., 2024; Viñas Torné et al., 2025). Deep learning models must therefore synthesize outcomes of combinatorial interventions from observed results of individual perturbations to generalize to previously unseen combinations (Norman et al., 2019). Individual cells cannot be observed before and after perturbation, since current sequencing destroys the cell, which makes transcriptomic data unpaired. Consequently, one must rely on population-level datasets for training models.

Lacking paired cells, regression models match each control to a random perturbed cell and learn only the average response distribution; benchmarks show that simply predicting the average training response matches deep models (Ahlmann-Eltze et al., 2025; Csendes et al., 2025). Recent approaches have focused on conditional flow matching (Lipman et al., 2022) for this problem. For example, CellFlow uses optimal transport coupled flow in a 50-dimensional PCA space (Klein et al., 2025) and scDFM models perturbations directly on per-gene expression values (Yu et al., 2026). The key challenge in these and related methods is the modeling of information exchanged among genes within the velocity field. Current single cell perturbation models keep the co-expression graph separate from the generative process. GEARS implements a GNN over co-expression to create static gene embeddings. scDFM applies a Transformer over genes using the co-expression graph as a binary attention mask. The gene-mixing operator is commonly implemented via dense attention or completely absent. This has three issues: a binary mask limits where the attention mechanism is applied, so the K-sparse prior enters only as a constraint on a dense operator; binary masks produced from absolute correlation disregard co-variation signs, which lets the model infer it from data, but the models are still evaluated on direction of change; and a prior introduced during embedding stage is fixed and cannot adapt as the generated cell moves from noise to perturbed state.

SPARC (Fig. 1) proposes a new framework for combinatorial perturbation modeling: it treats the signed graph as the gene mixing operator with no attention in the velocity field. Two separately

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

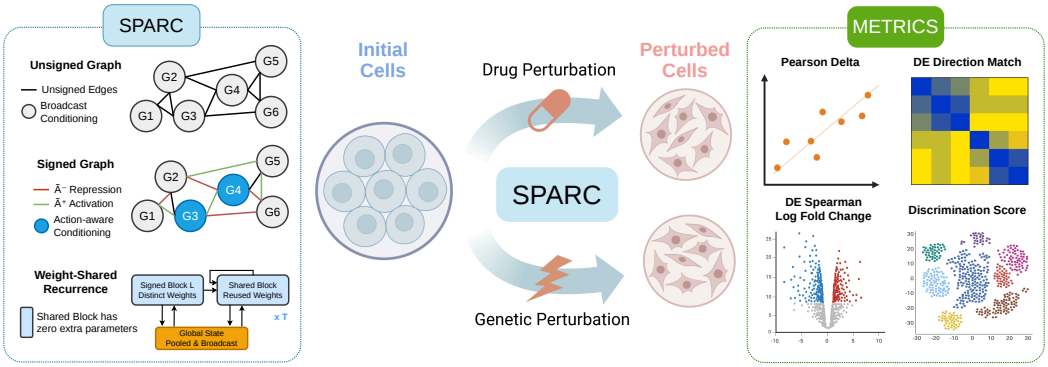


Figure 1: *Left:* **SPARC** assigns edges into positive and negative adjacencies, and scatters target action vectors to intervened nodes (blue). It extends reach of graph by reusing one shared weight block T times with a pooled global state. *Middle:* **SPARC** generates perturbed cell state population conditioned on control cells, under a drug treatment or a genetic perturbation. *Right:* Evaluation metrics suite from STATE (Adduri et al., 2026) cell-eval protocol.

normalized positive and negative adjacencies allow information exchange between genes. Each of these is gated by a global cell state, allowing positive and negative evidence to be weighted through condition and time. To avoid severing non-neighbors, the top K derived sparse operator is paired with a mean-pooled global cell state, giving every gene an all-to-all coarse path through a shared channel. Message passing advances one hop per layer, and to extend reach, a weight-shared block is applied recurrently with input injection. Broadcast conditioning only indicates that a perturbation occurred but not where. We apply a per target action vector for perturbed gene nodes, and an action aware sampler keeps these gene nodes present in every batch of training. Across genetic, drug, and cytokine benchmarks, SPARC improves Spearman LFC by up to 0.31 and direction match by up to 0.039 over the strongest baselines with a 10–13M-parameter model trained in 4 hours on one GPU, enabling direction-aware perturbation prediction without dense attention.

Contributions: **(1)** An attention free backbone that lets genes interact through separately normalized positive & negative adjacencies. To the best of our knowledge, no previous single cell perturbation prediction method propagates hidden states along signed edges in a generative velocity field. **(2)** A global cell state that gives an all-to-all path between genes, and a shared-weight recurrence that extends nominal graph reach at the parameter cost of one block. **(3)** A per target action vector that is scattered onto intervened gene nodes, with a training sampler that uses these nodes in every batch.

2 Related Work

scGen (Lotfollahi et al., 2019) models perturbation as a vector shift in a VAE and CPA (Lotfollahi et al., 2023) disentangles cell state from perturbation identity to support combinatorial knockouts; both recover population averages and treat genes as independent outputs. Foundation models trained on millions of transcriptomes, Geneformer (Theodoris et al., 2023) and scGPT (Cui et al., 2024), learn transferable gene representations but fail to beat nominal baselines out of distribution (Ahlmann-Eltze et al., 2025; Viñas Torné et al., 2025). CellOT (Bunne et al., 2023) target populations directly and learns a neural OT map between unpaired control and perturbed cells but cannot generalize to unseen perturbations. STATE (Adduri et al., 2026) trains a large transformer with a kernel population alignment loss. CellFlow (Klein et al., 2025) applies conditional flow matching to a 50-dimensional PCA space and scDFM (Yu et al., 2026) applies it to the full per-gene expression. Working per gene raises the question of how thousands of gene tokens should interact. GEARS (Roohani et al., 2024) answers this question using a GNN over co-expression and Gene Ontology graphs, and scDFM (Yu et al., 2026) with a Transformer under a binary co-expression mask. Both build the co-expression graphs from absolute correlation. Relational message passing is established for graphs with signed or typed edges (Derr et al., 2018; Schlichtkrull et al., 2018), and signed regulatory structure has been modeled for network inference; we use this as the gene-mixing operator of a generative velocity field (Karaaslanli et al., 2022; Xi et al., 2025). Replacing dense attention with a structured operator is established (Tolstikhin et al., 2021; Deng & Gu, 2024; Kipf & Welling, 2016), but they assume an

ordered token axis or a graph as ground truth, while gene tokens are unordered and their graph is a noisy prior. Sparse mixing over genes has two costs that dense attention avoids: restricting interaction to K neighbors severs genes that are not co-expressed but do interact, and message passing advances one hop per layer (Dehghani et al., 2018). Conditioning on simultaneous interventions requires permutation invariance. GEARS uses sum-then-MLP composition and CellFlow deep sets (Zaheer et al., 2017), while scDFM mean-pools target embeddings, normalizing by set size. All broadcast one condition vector to every gene, discarding target location.

3 METHOD

Problem formulation. Let $\mathcal{G} = \{g_1, \ldots, g_G\}$ be a fixed gene vocabulary and a cell an expression vector $\mathbf{c} \in \mathbb{R}^G_{\geq 0}$ of log-normalized counts. A perturbation $\rho = \{g_{j_1}, \ldots, g_{j_M}\}$ is a set of intervened targets. Given unperturbed cells and cells measured under ρ, we learn a velocity field $v_\theta(\mathbf{x}_t, t, \mathbf{c}_{\text{ctrl}}, \rho)$ that models transport from a source distribution toward $p_1(\cdot \mid \rho)$ over $t \in [0, 1]$, conditioned on a control cell $\mathbf{c}_{\text{ctrl}}$, and generalizing to ρ held out from training. Genes carry token indices τ_i over gene vocabulary, and a training batch operates on a gene subset $\mathcal{S} \subseteq \mathcal{G}$ of size N. All cells in a batch share a perturbation, so batch-level distributional losses compare two populations under one condition.

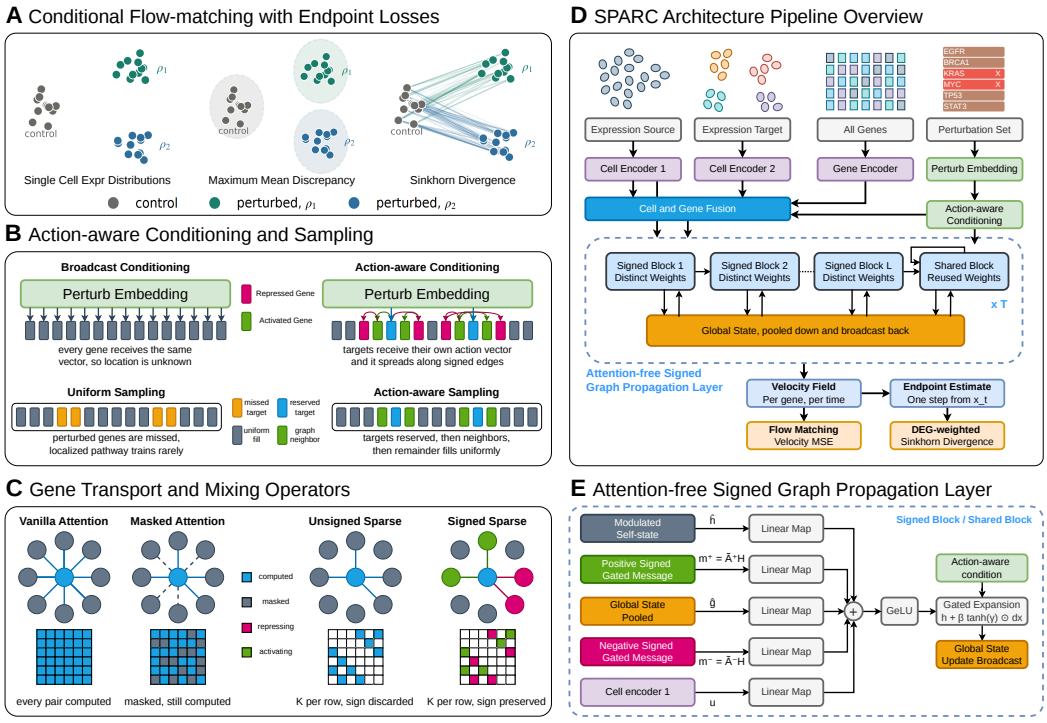


Figure 2: **Method overview. (A)** The one-step endpoint proxy is matched to the observed population via Sinkhorn divergence, which couples samples through a transport plan. **(B)** Action-aware conditioning gives each target its own vector that propagates along signed edges. Action-aware sampling reserves targets, then neighbors, then fills uniformly. **(C)** Unsigned sparse mixing keeps the top-K co-expression neighbors but drops sign; SPARC keeps those neighbors with sign preserved in two channels. **(D)** Gene expressions, gene identity, and perturbation set are encoded and fused, refined by L distinct-weight signed blocks then T applications of one shared block, all exchanging information with a pooled global state. The output velocity field is trained under flow-matching MSE plus an endpoint term. **(E)** A single layer contracts five sources into a shared hidden width, applies a zero-initialized gated expansion, and updates the global state by pooling and broadcast.

Signed graph construction. Let $\mathbf{X} \in \mathbb{R}^{C \times G}$ be the log-normalized expression of training cells (controls and training conditions), so held-out perturbations contribute nothing to the prior. We compute Pearson correlation $\mathbf{R} = \text{corr}(\mathbf{X})$ and keep, for each gene, its top-K partners $j \neq i$ by $|R_{ij}|$ and symmetrize, giving a neighbor table $\mathcal{N}(i)$ capped at 128 entries per gene; no external regulatory

network is used. Prior work keeps only $|R_{ij}|$, merging positive and negative co-variation. We split the retained edges by sign,

$$w_{ij}^{+} = \max(R_{ij},\, 0), \qquad w_{ij}^{-} = \max(-R_{ij},\, 0), \qquad j \in \mathcal{N}(i), \tag{1}$$

with $\mathcal{N}_{\mathcal{S}}(i) = \mathcal{N}(i) \cap \mathcal{S}$, scatter each into its own row-normalized adjacency over sampled subset,

$$\tilde{A}_{ij}^{\pm} = \frac{w_{ij}^{\pm}\, \mathbf{1}[j \in \mathcal{N}_{\mathcal{S}}(i)]}{\max\big(\sum_{j'} w_{ij'}^{\pm}\, \mathbf{1}[j' \in \mathcal{N}_{\mathcal{S}}(i)],\, \delta\big)}. \tag{2}$$

The two channels are normalized independently, with $\delta = 10^{-6}$ guarding empty rows, so a gene with many positive and few negative neighbors does not have its negative evidence drowned by degree.

Spectral gene coordinates. From Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ on the symmetrized $|\mathbf{R}|$-sparsified adjacency, with $D_{ii}^{-1/2} = 0$ for isolated nodes, we take the eigenvectors of the K_s smallest nonzero eigenvalues; gene i's entries form $\mathbf{s}_i \in \mathbb{R}^{K_s}$. The decomposition is computed once and frozen, fixing eigenvector sign conventions across training and inference. The gene representation is given by $\mathbf{e}_i = \mathrm{LN}(\mathrm{Emb}(\tau_i)) + \mathrm{MLP}_s(\mathbf{s}_i)$.

3.1 Input Representation and Perturbation Conditioning

Expression encoding. Values are encoded as $f_\phi(\max(c_i, 0))$, with f_ϕ a two-layer GELU MLP followed by layer normalization. The same rectification $r(\mathbf{x}) = \max(\mathbf{x}, 0)$ is applied to $\mathbf{x}_t$ in training and at every solver evaluation, so the network input has the support of expression data in both. Write $\cdot$ for $(t, \mathbf{c}_{\text{ctrl}}, \rho)$ and let $u_t(\mathbf{x}_t) = \mathbb{E}[\mathbf{x}_1 - \mathbf{x}_0 \mid \mathbf{x}_t, \cdot\,]$ be the marginal velocity. For a fixed gene subset, the flow-matching loss is minimized by

$$v^* = \mathbb{E}\big[\mathbf{x}_1 - \mathbf{x}_0 \mid r(\mathbf{x}_t), \cdot\,\big] = \mathbb{E}\big[u_t(\mathbf{x}_t) \mid r(\mathbf{x}_t), \cdot\,\big], \tag{3}$$

where the second equality is the tower property, since $r(\mathbf{x}_t)$ is a function of $\mathbf{x}_t$. r is the identity for $x_{t,i} > 0$ and hides only the magnitude of negative entries from the network input. Hence $v^* = u_t$ wherever u_t does not depend on those magnitudes; otherwise v^* is the projection of u_t onto the rectified state, and transport to p_1 is approximate. Since $\mathbf{x}_1 \geq 0$, negative coordinates arise only from the source. For $x_{1,i} > 0$, $P(x_{t,i} < 0 \mid x_{1,i}) = \Phi\big(-t\, x_{1,i}/(1-t)\big) \to 0$ as $t \to 1$, with Φ the standard normal CDF. For $x_{1,i} = 0$, $x_{t,i} = (1-t)\, x_{0,i}$ is negative with probability $1/2$ for all $t < 1$. Hidden entries have target velocity $(x_{1,i} - x_{t,i})/(1-t) > 0$ and rise until visible; clipping zeroes those left. Independent encoders f_{ϕ_1}, f_{ϕ_2} handle the interpolated state $\mathbf{x}_t$, control cell, input token fuses identity, current state, and control:

$$\mathbf{h}_i^{(0)} = \mathrm{MLP}_{\text{in}}([\,\mathbf{e}_i \,\|\, f_{\phi_1}(\max(x_{t,i}, 0)) \,\|\, f_{\phi_2}(c_{\text{ctrl},i})\,])\,. \tag{4}$$

Control encoding $\mathbf{u}_i = f_{\phi_2}(c_{\text{ctrl},i})$ is retained as a reference consulted at every layer. We aggregate target embeddings by permutation-invariant sum pooling, without normalizing by set size:

$$\mathbf{p} = \mathbf{1}[\textstyle\sum_m \nu_m > 0]\, \omega_\psi\left(\sum_{m=1}^{M} \nu_m\, \phi_\psi(\mathbf{e}_{j_m})\right), \qquad \nu_m = \mathbf{1}[\tau_{j_m} \neq \tau_{\text{ctrl}}]\,, \tag{5}$$

with ϕ_ψ, ω_ψ two-layer SiLU MLPs, so $\mathbf{p} = \mathbf{0}$ under genetic control. For genetic perturbations $\mathbf{e}_{j_m}$ comes from the gene encoder, placing intervened and measured genes in a common basis. For drugs and cytokines we use a layer-normalized learned embedding of the identifier, the control identifier is embedded like any other, and $\nu_m = 1$.

Action conditioning and sampling. Genetic targets are measured genes, so we scatter a per-target action vector onto their nodes:

$$\mathbf{a}_i = \mathbf{p} + \sum_{m=1}^{M} \nu_m\, \mathbf{1}[\pi(\tau_{j_m}) = i]\, (W_a \mathbf{e}_{j_m} + \mathbf{b}_a)\,, \qquad \mathbf{h}_i^{(0)} \leftarrow \mathbf{h}_i^{(0)} + W_{\text{proj}}\, \mathbf{a}_i, \tag{6}$$

where π maps a token to its column in $\mathcal{S}$ (none if absent). W_{proj} is bias-free and initialized with s.d. 10^{-2}, so $\mathbf{a}_i$ initially enters near zero. Uniform sampling includes a target with probability N/G, so the localized term would rarely receive gradient. For genetic perturbations, we therefore fill $\mathcal{S}$ with all targets, then their neighbors (up to K each, ranked by $|R_{ij}|$ over the uncapped graph), then uniformly sample genes, and shuffle. Since $M \leq N$, each target is present in every batch. For drugs and cytokines the second term is dropped and sampling is uniform. At evaluation, $\mathcal{S}$ is the benchmark's evaluated gene set, over which $\tilde{A}^{\pm}$ is renormalized, and the second term acts on targets within it.

3.2 Attention-Free Signed Propagation

Each layer updates a per-gene state $\mathbf{H} \in \mathbb{R}^{B\times N\times d}$ and a global cell state $\mathbf{g} \in \mathbb{R}^{B\times d}$, initialized as the mean of the input gene tokens. All layers share one condition vector combining flow time and perturbation, $\mathbf{q} = \mathrm{MLP}\big([\,\mathrm{MLP}(\mathrm{SinCos}(t)) \,\|\, \mathbf{p}\,]\big)$, with $\mathbf{p}$ from Eq. 5. In each layer, a linear map of $\mathbf{q}$, initialized to zero (Peebles & Xie, 2023), outputs a scale $\mathbf{s}$, a shift $\mathbf{b}$, and a gate $\boldsymbol{\gamma}$, separately for genes (h) and the global state (g), giving $\hat{\mathbf{h}}_i = \mathrm{LN}(\mathbf{h}_i)\odot(1+\mathbf{s}_h)+\mathbf{b}_h$ and $\hat{\mathbf{g}} = \mathrm{LN}(\mathbf{g})\odot(1+\mathbf{s}_g)+\mathbf{b}_g$. Since the map starts at zero, the gates $\tanh(\boldsymbol{\gamma})$ start closed and every layer is initially the identity.

Gated signed transport. Positive and negative messages are gathered from the unmodulated states along their own adjacencies and gated per cell by the global state: $\mathbf{m}^{\pm} = \tilde{A}^{\pm}\mathbf{H}$, $[\boldsymbol{\eta}^{+} \,\|\, \boldsymbol{\eta}^{-}] = \sigma(\mathrm{MLP}_{\eta}([\,\hat{\mathbf{g}} \,\|\, \mathbf{q}\,]))$. This operator replaces attention as it is the only gene-specific channel between genes, its support is the sparse prior derived from the top-K graph, and its two sign types have separate weights. The layer update contracts five sources (self, positive neighbors, negative neighbors, global state, control reference) into a shared width d_{hid}:

$$\mathbf{z}_i = W_{\text{self}}\hat{\mathbf{h}}_i + W_{+}\big(\boldsymbol{\eta}^{+}\odot \mathrm{LN}(\mathbf{m}_i^{+})\big) + W_{-}\big(\boldsymbol{\eta}^{-}\odot \mathrm{LN}(\mathbf{m}_i^{-})\big) + W_{\text{glob}}\hat{\mathbf{g}} + W_{\text{ctrl}}\mathrm{LN}(\mathbf{u}_i), \tag{7}$$

$$\mathbf{h}_i \leftarrow \mathbf{h}_i + \beta\tanh(\boldsymbol{\gamma}_h)\odot W_{\text{exp}}\,\mathrm{GELU}(\mathbf{z}_i), \tag{8}$$

with $\beta = 0.1$. Contracting the control reference at each layer plays the role of cross-attention to the control in a Transformer backbone, as a per-gene linear map not a pairwise interaction.

The global channel. Sparsification to K neighbors severs communication between genes that are not co-expressed but do interact. We add a coarse all-to-all channel, the global state pools the updated gene field and the control, and is broadcast back into every gene's update through W_{glob},

$$\mathbf{g} \leftarrow \mathbf{g} + \beta\tanh(\boldsymbol{\gamma}_g)\odot \mathrm{MLP}_g\big(\big[\,\hat{\mathbf{g}} \,\|\, \tfrac{1}{N}\textstyle\sum_i \mathbf{h}_i \,\|\, \tfrac{1}{N}\sum_i \mathbf{u}_i\,\big]\big)\,. \tag{9}$$

Gene interaction thus decomposes into a signed sparse component carrying signed co-expression structure and a pooled component carrying cell-wide state.

Recurrent refinement and output. Each layer passes gene-specific messages one hop along the graph; the global channel is all-to-all but shared by all genes, so gene-specific paths between distant genes require depth. Instead of stacking more layers, we reuse one block $\mathcal{B}_{\text{rec}}$ for T steps after the L distinct layers. $\mathcal{B}_{\text{rec}}$ is a signed layer as above and therefore residual: it maps $(\tilde{\mathbf{H}}, \tilde{\mathbf{g}})$ to $(\tilde{\mathbf{H}} + \Delta_h, \tilde{\mathbf{g}} + \Delta_g)$, with $\Delta_{h,i} = \beta\tanh(\boldsymbol{\gamma}_h)\odot W_{\text{exp}}\,\mathrm{GELU}(\mathbf{z}_i)$ and $\Delta_g = \beta\tanh(\boldsymbol{\gamma}_g)\odot \mathrm{MLP}_g(\cdot)$ computed from its input. Let $(\mathbf{H}_{\text{init}}, \mathbf{g}_{\text{init}})$ be the output of the L distinct layers and $(\mathbf{H}^{(0)}, \mathbf{g}^{(0)}) = (\mathbf{H}_{\text{init}}, \mathbf{g}_{\text{init}})$. Each step injects the initial state (Dehghani et al., 2018) and keeps a fraction κ of the block's update:

$$\big(\Delta_h^{(k)}, \Delta_g^{(k)}\big) = \Delta\big(\mathbf{H}^{(k)} + \mathbf{H}_{\text{init}},\ \mathbf{g}^{(k)} + \mathbf{g}_{\text{init}}\big), \mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \kappa\,\Delta_h^{(k)}, \mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} + \kappa\,\Delta_g^{(k)}, \tag{10}$$

for $k = 0, \ldots, T-1$, so the block's input at $k = 0$ is $(2\mathbf{H}_{\text{init}}, 2\mathbf{g}_{\text{init}})$. Here $\kappa = \sigma(\ell_\kappa)$ is a learned scalar initialized to 0.1. The recurrent gene update is $\kappa\beta\tanh(\boldsymbol{\gamma}_h)\odot W_{\text{exp}}\,\mathrm{GELU}(\mathbf{z}_i)$, and W_{exp} and $\boldsymbol{\gamma}_h$ are trained jointly with κ. The step size is therefore set by this product, not by κ alone. This extends the nominal reach of $\tilde{A}^{\pm}$ to $L + T$ hops while adding the parameters of a single block. The block's condition modulation and control projection do not change across steps, so they are computed once. The final gene states are layer-normalized and concatenated with $\mathbf{p}$. MLP decoder maps each gene to a velocity, multiplied by learned sigmoid gate.

3.3 Training Objective

Conditional flow matching. For each training cell we draw Gaussian noise $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, an observed perturbed cell $\mathbf{x}_1$, and a time $t \sim \mathcal{U}(0, 1)$, and form the intermediate state $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$. Along this straight path the velocity is $\mathbf{x}_1 - \mathbf{x}_0$, and the network learns to predict it from the current state, the control cell, and the perturbation: $\mathcal{L}_{\text{FM}} = \mathbb{E}\big\|v_\theta(\mathbf{x}_t, t, \mathbf{c}_{\text{ctrl}}, \rho) - (\mathbf{x}_1 - \mathbf{x}_0)\big\|^2$. The control cell only conditions the network; it is not where the flow starts.

Endpoint objective. Flow matching uses sample-wise velocity supervision; we add a batch-level distributional term. Following scDFM (Yu et al., 2026), we form the one-step endpoint proxy, $\hat{\mathbf{x}}_1 = \mathbf{x}_t + (1-t)\,v_\theta$, and compare its empirical distribution with that of observed perturbed cells.

Since $\mathbf{x}_1 - \mathbf{x}_t = (1-t)(\mathbf{x}_1 - \mathbf{x}_0)$, the proxy under the marginal velocity u_t equals $\mathbb{E}[\mathbf{x}_1 \mid \mathbf{x}_t, \cdot]$. Under v^* of Eq. 3 it equals

$$\hat{\mathbf{x}}_1 = \mathbb{E}\big[\mathbf{x}_1 \mid r(\mathbf{x}_t), \cdot\big] + \Big(\mathbf{x}_t - \mathbb{E}\big[\mathbf{x}_t \mid r(\mathbf{x}_t), \cdot\big]\Big), \tag{11}$$

where the second term is zero on every coordinate with $x_{t,i} > 0$. The auxiliary term can shift the flow-matching optimum, so we give it a small weight. It compares proxy endpoints, not samples obtained by integrating the learned flow. We use the debiased Sinkhorn divergence (Feydy et al., 2019), with a squared cost in which gene i is weighted by its training response to ρ,

$$w_i = 0.1 + 0.9\,\frac{|\bar{x}_i^{\rho} - \bar{x}_i^{\text{ctrl}}|}{\max_{j\in\mathcal{G}} |\bar{x}_j^{\rho} - \bar{x}_j^{\text{ctrl}}|}, \tag{12}$$

where $\bar{x}^{\rho}, \bar{x}^{\text{ctrl}}$ are training means under ρ and control. This formula applies when its denominator is positive; w enters only the loss. Batches without a DEG weight vector skip the endpoint term (App. A). Entropic regularization is 0.1 times median batch cost, with 50 iterations, and $\mathcal{L}_{\text{OT}}$ is ramped in over the first $15{,}000$ steps: $\mathcal{L} = \mathcal{L}_{\text{FM}} + 0.1\,\min(1, k/15{,}000)\,\mathcal{L}_{\text{OT}}$, where k is the training step. Starting from Gaussian noise, we integrate v_θ with 19 fixed RK4 steps and set negative values of the final sample to zero.

4 Experiments

Datasets and splits. Norman (Norman et al., 2019) contains CRISPRa single and double perturbations in K562 cell lines. ComboSciPlex (Srivatsan et al., 2020; Lotfollahi et al., 2023) contains combinatorial drug perturbations. Parse-PBMC (Biosciences, 10) contains cytokine perturbations of primary immune cells from multiple donors. Drug and cytokine identifiers are not gene tokens, so Eq. 6 reduces to broadcast, and sampling is uniform on these two datasets. On Norman, the *additive* split holds out 37 double perturbations per fold, all of whose constituent singles remain in training, and the benchmark evaluates a fixed subset of 10 of them; the *holdout* split removes 12 random genes and withholds every single and double perturbation containing them. ComboSciPlex withholds five fixed drug combinations and two single drugs, each of which appears in training combinations and so has a trained embedding. Parse-PBMC withholds 70% of the cytokine conditions of four held-out donors, so test cytokines are observed only in other donors. The model has no donor input, so predictions for held-out donors are cross-donor responses. Training pairs controls with target cells of the same cell type; evaluation draws controls from the full pool, as in scDFM. Genes are the $5{,}000$ most variable in training cells, plus all Norman targets and all evaluated genes, and the gene prior uses training cells only. Splits follow the cited benchmarks; Fig. 4A,B use the holdout split. We report mean $\pm$ s.d. over seeds $\{42, 43, 44\}$; baseline s.d. values are reported by the source papers.

Baselines. We compare against scDFM (Yu et al., 2026), which shares the ambient-space flow matching formulation but differs in backbone, endpoint objective, set pooling and conditioning; CellFlow (Klein et al., 2025); STATE (Adduri et al., 2026); GEARS (Roohani et al., 2024); CPA (Lotfollahi et al., 2023); Geneformer (Theodoris et al., 2023); scGPT (Cui et al., 2024); and, on Parse-PBMC, scVI (Lopez et al., 2018) and Lingshu (Zhang et al., 2026). Baselines and their numbers are those of Yu et al. (2026) (Norman, ComboSciPlex) and Zhang et al. (2026) (Parse-PBMC); SPARC is evaluated on the same held-out conditions, gene sets, control pool and cell-eval protocol. The Control row predicts the unperturbed population for every condition. It is untrained with no s.d., and direction match reflects sampling differences between drawn and reference control cells, not response.

Metrics and protocol. Metrics are computed with cell-eval (Roohani et al., 2025) on the benchmark's fixed set of $1{,}000$ evaluated genes, following the protocol of Yu et al. (2026) and the benchmarks it builds on (App. B). This gene set defines the evaluated genes for every method and the input tokens for gene-token models such as scDFM, so it is shared across methods. SPARC's localized term uses only the identity of the targets of ρ, which every method receives as its condition, and never their measured response. We evaluate the benchmark's 10 held-out conditions per split (all 7 on ComboSciPlex); for each, 128 cells are generated, each conditioned on a randomly drawn control cell, and compared with all observed cells of that condition. DE genes are called against control with a Wilcoxon rank-sum test at BH-adjusted $p < 0.05$. Spearman LFC correlates predicted and observed log-fold changes over observed DE genes, direction match is the fraction of those genes whose predicted change has the observed sign, and DE recall is the fraction also called DE in the

Table 1: Norman (additive CRISPRa split). Best in green, second best in blue.

Method	Spearman LFC (sig. DE) ↑	Direction match ↑	Pearson Δ ↑	MSE ($\times 10^{-3}$) ↓	Discrim. cosine ↑	DE recall ↑
Control	N.A.	0.674	N.A.	18.4	0.514	0.681
scGPT	0.007 ± 0.018	0.588 ± 0.015	0.530 ± 0.020	13.5 ± 0.3	0.540 ± 0.025	0.694 ± 0.012
Geneformer	0.374 ± 0.021	0.613 ± 0.010	0.773 ± 0.015	4.1 ± 0.2	0.824 ± 0.022	0.752 ± 0.014
GEARS	0.562 ± 0.018	0.590 ± 0.012	0.742 ± 0.016	13.9 ± 0.3	0.860 ± 0.020	0.788 ± 0.011
CPA	0.071 ± 0.025	0.636 ± 0.014	0.385 ± 0.022	34.4 ± 0.4	0.602 ± 0.030	0.703 ± 0.015
STATE	0.529 ± 0.020	0.602 ± 0.011	0.411 ± 0.025	3.6 ± 0.3	0.514 ± 0.028	0.690 ± 0.013
CellFlow	0.550 ± 0.019	0.878 ± 0.008	0.868 ± 0.012	3.9 ± 0.2	0.932 ± 0.015	0.795 ± 0.010
scDFM	$\mathbf{0.571 \pm 0.016}$	$\mathbf{0.885 \pm 0.007}$	$\mathbf{0.885 \pm 0.011}$	$\mathbf{3.2 \pm 0.2}$	$\mathbf{0.974 \pm 0.012}$	$\mathbf{0.818 \pm 0.009}$
SPARC	$\mathbf{0.884 \pm 0.018}$	$\mathbf{0.916 \pm 0.009}$	$\mathbf{0.876 \pm 0.012}$	$\mathbf{3.0 \pm 0.2}$	$\mathbf{0.941 \pm 0.018}$	$\mathbf{0.919 \pm 0.010}$

Table 2: Norman (holdout CRISPRa split). Best in green, second best in blue.

Method	Spearman LFC (sig. DE) ↑	Direction match ↑	Pearson Δ ↑	MSE ($\times 10^{-3}$) ↓	Discrim. cosine ↑	DE recall ↑
Control	N.A.	0.622	N.A.	9.5	0.522	0.675
scGPT	-0.114 ± 0.020	0.496 ± 0.012	0.450 ± 0.018	8.0 ± 0.2	0.568 ± 0.022	0.641 ± 0.014
Geneformer	0.367 ± 0.018	0.503 ± 0.014	0.696 ± 0.016	3.6 ± 0.3	0.807 ± 0.024	0.709 ± 0.011
GEARS	0.357 ± 0.021	0.551 ± 0.011	0.665 ± 0.019	7.5 ± 0.4	0.827 ± 0.018	0.737 ± 0.013
CPA	0.117 ± 0.022	0.618 ± 0.016	0.284 ± 0.024	35.6 ± 0.5	0.580 ± 0.027	0.725 ± 0.016
STATE	0.612 ± 0.024	0.607 ± 0.013	0.247 ± 0.021	3.3 ± 0.4	0.524 ± 0.031	0.680 ± 0.012
CellFlow	0.286 ± 0.016	0.785 ± 0.009	$\mathbf{0.711 \pm 0.014}$	3.5 ± 0.2	0.807 ± 0.017	0.774 ± 0.011
scDFM	$\mathbf{0.696 \pm 0.014}$	$\mathbf{0.821 \pm 0.008}$	$\mathbf{0.713 \pm 0.013}$	$\mathbf{3.0 \pm 0.1}$	$\mathbf{0.891 \pm 0.014}$	$\mathbf{0.802 \pm 0.008}$
SPARC	$\mathbf{0.772 \pm 0.012}$	$\mathbf{0.835 \pm 0.009}$	0.697 ± 0.009	$\mathbf{2.9 \pm 0.2}$	$\mathbf{0.829 \pm 0.015}$	$\mathbf{0.890 \pm 0.005}$

Table 3: ComboSciPlex (held-out drug combinations). Best in green, second best in blue.

Method	Spearman LFC (sig. DE) ↑	Direction match ↑	Pearson Δ ↑	MSE ($\times 10^{-3}$) ↓	Discrim. cosine ↑	DE recall ↑
Control	N.A.	0.571	N.A.	32.4	0.571	0.715
scGPT	-0.126 ± 0.025	0.820 ± 0.009	0.832 ± 0.011	3.1 ± 0.2	0.857 ± 0.014	0.806 ± 0.008
CPA	0.791 ± 0.014	0.882 ± 0.007	0.815 ± 0.013	2.9 ± 0.2	$\mathbf{0.898 \pm 0.009}$	0.871 ± 0.009
scDFM	$\mathbf{0.829 \pm 0.009}$	$\mathbf{0.899 \pm 0.005}$	$\mathbf{0.893 \pm 0.007}$	$\mathbf{2.8 \pm 0.1}$	0.878 ± 0.011	$\mathbf{0.905 \pm 0.006}$
SPARC	$\mathbf{0.908 \pm 0.006}$	$\mathbf{0.938 \pm 0.011}$	$\mathbf{0.935 \pm 0.010}$	$\mathbf{2.0 \pm 0.2}$	$\mathbf{0.905 \pm 0.014}$	$\mathbf{0.973 \pm 0.001}$

Table 4: Parse-PBMC (held-out donor-cytokine pairs). Best in green, second best in blue.

Method	Spearman LFC (sig. DE) ↑	Direction match ↑	Pearson Δ ↑	MSE ($\times 10^{-3}$) ↓	Discrim. cosine ↑	DE recall ↑
scVI	0.267 ± 0.012	0.605 ± 0.008	0.036 ± 0.011	4.5 ± 0.2	0.726 ± 0.014	0.742 ± 0.009
scGPT	0.154 ± 0.015	0.683 ± 0.007	0.358 ± 0.013	5.6 ± 0.2	0.752 ± 0.009	0.781 ± 0.008
STATE	$\mathbf{0.607 \pm 0.011}$	$\mathbf{0.721 \pm 0.006}$	0.460 ± 0.008	$\mathbf{3.8 \pm 0.1}$	0.870 ± 0.011	$\mathbf{0.865 \pm 0.007}$
Lingshu	0.509 ± 0.009	0.704 ± 0.008	$\mathbf{0.593 \pm 0.010}$	4.0 ± 0.2	$\mathbf{0.882 \pm 0.012}$	0.850 ± 0.006
SPARC	$\mathbf{0.732 \pm 0.012}$	$\mathbf{0.734 \pm 0.018}$	$\mathbf{0.587 \pm 0.012}$	$\mathbf{2.4 \pm 0.1}$	$\mathbf{0.910 \pm 0.015}$	$\mathbf{0.995 \pm 0.004}$

prediction; it ignores false positives, so we treat it as secondary. Pearson Δ correlates predicted and observed mean shifts from control across genes, MSE compares mean expression, and discrimination ranks each predicted pseudobulk against all observed pseudobulks under cosine distance. Control's shift from the reference control is sampling noise not a response, so its Spearman LFC and Pearson Δ are not reported. Hyperparameters are given in App. A.

Results. On the Norman additive split (Table 1), SPARC leads on Spearman LFC, direction match and MSE, with the largest margin on Spearman LFC (0.884 against 0.571 for scDFM), while scDFM leads on Pearson Δ and discrimination. On the holdout split (Table 2), SPARC raises Spearman LFC from 0.696 to 0.772 and trails scDFM on Pearson Δ and discrimination. The interaction between target coverage of the evaluated gene set and the two localization mechanisms is discussed in App. C. On ComboSciPlex (Table 3), SPARC leads on all five primary metrics, reducing MSE from 2.8 to 2.0×10^{-3}, and on Parse-PBMC (Table 4) it leads on most metrics, raising Spearman LFC from 0.607 to 0.732. Neither dataset has gene-token targets, so localized conditioning and action-aware sampling are inactive; for remaining components, only endpoint objective is ablated on these datasets.

Ablations. On ComboSciPlex (Fig. 3A), DEG-weighted Sinkhorn is best on all four metrics throughout training, most clearly on Pearson Δ (0.940 against 0.902 for the next best). DEG weighting improves Sinkhorn on every metric but improves MMD only on direction match, Pearson Δ and final MSE, so weighting helps more under a transport plan than under a kernel discrepancy. Sampling is the most consequential choice (Fig. 3B). Both uniform-sampling variants peak between 90k and 140k steps and regress, with MSE rising above 10^{-2}, before partially recovering. With $N = 1{,}000$ of roughly $5{,}000$ genes, uniform sampling includes a given target in about one batch in five; training loss keeps decreasing through the regression, and we hypothesize that without targets the model fits training conditions through $\mathbf{p}$ alone. Under action-aware sampling, localized conditioning improves over broadcast on every metric, most on Pearson Δ (0.84 against 0.77), and passes 0.85 Spearman LFC before 100k steps, a level broadcast reaches only near the end; under broadcast, $T = 8$ does

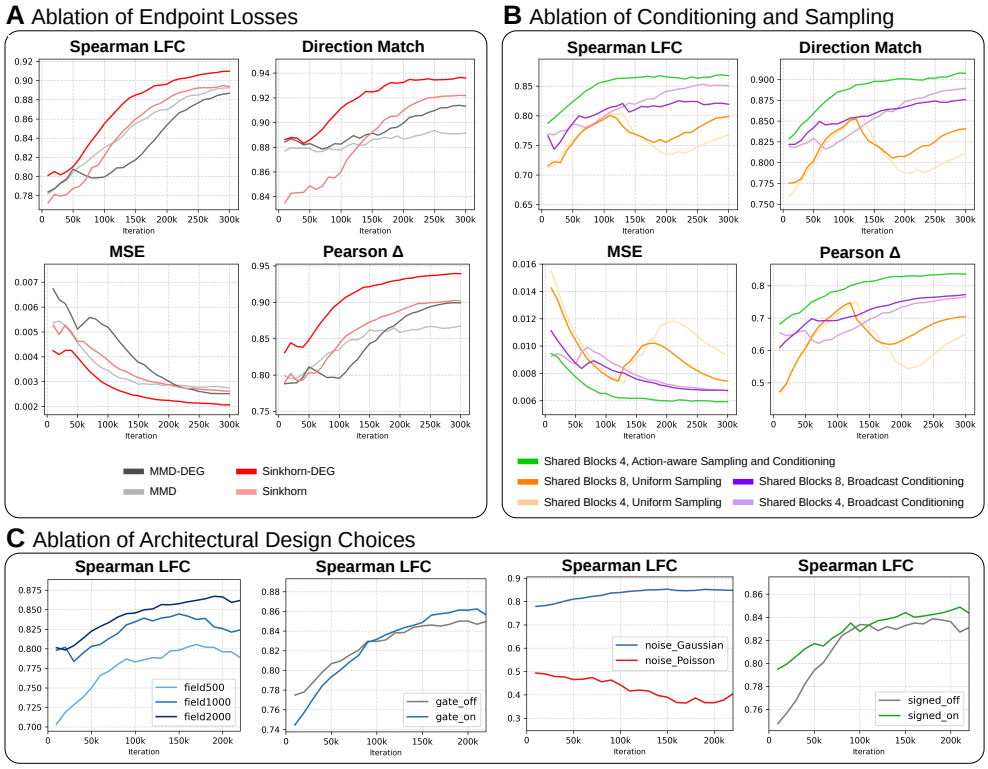


Figure 3: **Ablations.** Metrics on held-out conditions vs training iteration (single seed 45), one factor varied per panel; values quoted in the text are read from these curves. **(A)** Endpoint objective on ComboSciPlex: Sinkhorn vs MMD, each with and without DEG weighting. **(B, C)** Norman (additive split). **(B)** Action-aware vs broadcast conditioning and action-aware vs uniform gene sampling, at $T \in \{4, 8\}$ shared recurrent blocks. **(C)** Field size N, global-state gating, source distributions, and signed vs unsigned adjacency.

not improve on $T = 4$. The source distribution has the largest effect (Fig. 3C): a Poisson source, a log-normalized Poisson–Gamma resampling of the control cell, lowers Spearman LFC from about 0.85 to 0.40 and degrades over training, likely because a source near the control leaves a short path that bypasses the condition. Spearman LFC increases with field size ($500 < 1000 < 2000$) on the same $1{,}000$ evaluated genes. Signed edges improve Spearman LFC over an unsigned single-channel variant with unit edge weights. Global-state gating overtakes ungated variant after 100k steps.

Distributional fit and per-perturbation behavior. Fig. 4A compares predicted and observed cells on the Norman holdout split in a shared UMAP. With the MMD endpoint, pooled predictions cover the observed cells but also form a separate cluster with no observed counterpart, and for CEBPE+control and FOSB+UBASH3B nearly all predicted cells fall in it. With the Sinkhorn endpoint, most predicted cells for both conditions lie within the observed population, and a smaller off-support mode remains. MMD compares kernel-smoothed averages, so a batch can score well with part of it off-support, whereas Sinkhorn assigns every predicted cell to observed cells and pays for each off-support cell directly. Per perturbation on the holdout split (Fig. 4B), all variants reach Spearman LFC between 0.84 and 0.93 on BPGM+SAMD1, CBL+CNN1 and CEBPE+control. The full model's advantage concentrates where variants diverge: on FOSB+UBASH3B it reaches 0.86 against at most 0.81, and on MAP2K3+MAP2K6 0.77 against at most 0.70, with direction match rising from 0.80 to 0.88.

Regulatory and pathway recovery. To see if SPARC preserves biological structure beyond gene-level prediction accuracy, we performed regulatory and pathway recovery analysis on additive split. Against TRRUST (Han et al., 2018) (Fig. 4C), SPARC reaches 0.606% precision for annotated TF–target relationships, compared with 0.331% for the static co-expression prior and 0.303% for scDFM. The gain is reduced with a Poisson source (0.308%), without gating (0.364%), or with unsigned edges (0.545%). Interestingly, $N = 2000$ gives the lowest TRRUST precision (0.175%) despite the highest

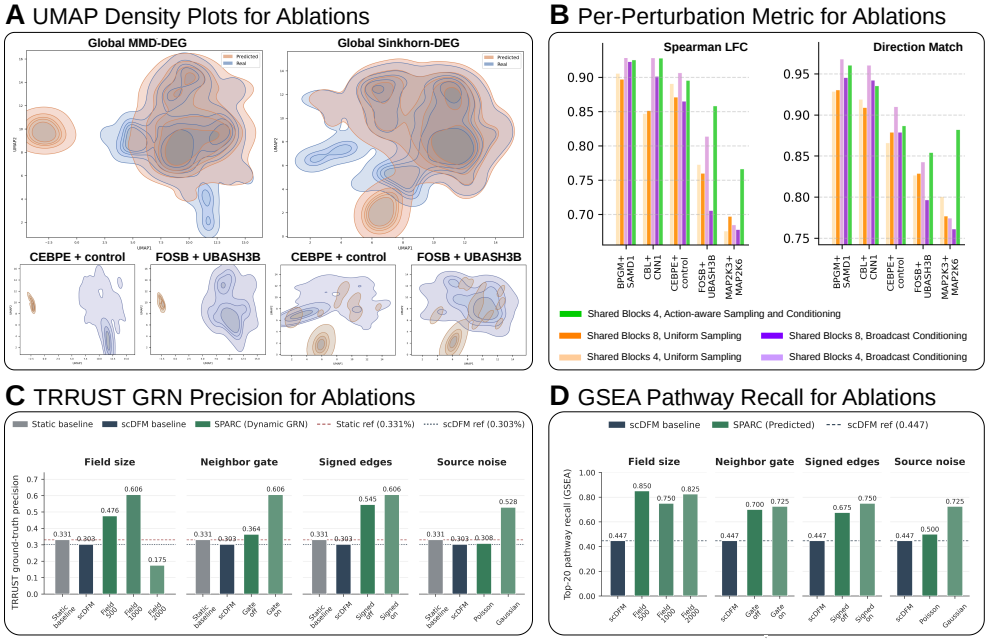


Figure 4: **Distributional & biological evaluation.** **(A)** UMAP density on holdout split of predicted (orange) and observed (blue) cells for models trained with DEG-weighted MMD and DEG-weighted Sinkhorn endpoints, pooled over conditions (top) and for CEBPE+control and FOSB+UBASH3B (bottom). **(B)** Per-perturbation Spearman LFC and direction match for the conditioning and sampling ablations. **(C)** TRRUST precision (%) of retained TF-target pairs for ablations, with the static co-expression prior. **(D)** Top-20 GSEA pathway recall for ablations on additive split. (single seed 45)

Spearman LFC among field sizes, showing that predictive and regulatory recovery need not coincide. At the pathway level, SPARC also recovers more of the pathways observed after perturbation. In top-20 Gene Set Enrichment Analysis (GSEA) pathway recall (Fig. 4D), all SPARC variants exceed scDFM (0.447), with Poisson-source closest to the baseline (0.500). Results are consistent with gains extending beyond gene-level response prediction to regulatory and pathway-level structure.

5 Discussion

SPARC's gains concentrate on Spearman LFC and direction match, the metrics most sensitive to the sign and ranking of gene-level responses. Replacing signed edges with unsigned ones lowers Spearman LFC, TRRUST precision and pathway recall. SPARC trails scDFM on Pearson delta and discrimination on Norman, which reward broad transcriptome-wide shifts. A sparse operator with one pooled channel may represent such diffuse shifts less precisely than dense attention. In exchange, the signed adjacency does not depend on hidden states, is built once per batch and is shared across all layers, and recurrence adds depth at the parameter cost of one block. The resulting 10–13M-parameter model trains in about 4 hours on one H100, against about 18 for scDFM, while leading Spearman LFC in all four settings. Response direction and ranking can therefore be predicted without a dense pairwise operator. Gene sampling and the source distribution are the most consequential choices: uniform sampling deprives the localized pathway of gradient, and a source near the control shortens the path enough to bypass the condition. Sinkhorn outperforms MMD because it pays for every off-support cell directly. The field size with the best Spearman LFC gives the lowest TRRUST precision, so predictive and regulatory evaluation should be reported together. Code is in supplementary. **Limitations.** The prior is a correlation graph with fixed support, signed but not causal, and localized conditioning applies only to measured gene targets. The unsigned ablation changes sign, weights, and sampler order. Rectifying the interpolated state makes the learned field an approximation of the marginal velocity (Eq. 3).

AI USE STATEMENT

Generative AI tools were only used to improving grammar and readability of the manuscript, which falls under the recommended disclosure category.

ETHICS STATEMENT

Not Applicable.

REPRODUCIBILITY STATEMENT

The source code is submitted as Supplementary Material in a zip folder.

REFERENCES

Abhinav K Adduri, Dhruv Gautam, Beatrice Bevilacqua, Mohsen Naghipourfar, Alishba Imran, Rohan Shah, Noam Teyssier, Rishi Verma, Christopher Carpenter, Basak Eraslan, et al. Predicting cellular responses to perturbation across diverse contexts with state. *Cell*, 2026.

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.

Parse Biosciences. million human pbmcs in a single experiment, 2023. *URL https://www. parsebiosciences. com/datasets/10-million-human-pbmcs-in-a-single-experiment*, 10.

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.

Gerold Csendes, Gema Sanz, Kristóf Z Szalay, and Bence Szalai. Benchmarking foundation cell models for post-perturbation rna-seq prediction. *BMC genomics*, 26(1):393, 2025.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480, 2024.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Rui Deng and Tianpei Gu. Cu-mamba: Selective state space models with channel learning for image restoration. In *2024 IEEE 7th international conference on multimedia information processing and retrieval (MIPR)*, pp. 328–334. IEEE, 2024.

Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE international conference on data mining (ICDM)*, pp. 929–934. IEEE, 2018.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd international conference on artificial intelligence and statistics*, pp. 2681–2690. PMLR, 2019.

Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46 (D1):D380–D386, 2018.

Abdullah Karaaslanli, Satabdi Saha, Selin Aviyente, and Tapabrata Maiti. scsgl: kernelized signed graph learning for single-cell gene regulatory network inference. *Bioinformatics*, 38(11):3011–3019, 2022.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 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*, pp. 2025–04, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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.

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.

Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Xiaoning Qi, Lianhe Zhao, Chenyu Tian, Yueyue Li, Zhen-Lin Chen, Peipei Huo, Runsheng Chen, Xiaodong Liu, Baoping Wan, Shengyong Yang, et al. Predicting transcriptional responses to novel chemical perturbations using deep generative model for drug discovery. *Nature Communications*, 15(1):9256, 2024.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature biotechnology*, 42(6):927–935, 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.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

Sanjay R Srivatsan, José L McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A Pliner, Dana L Jackson, Riza M Daza, Lena Christiansen, et al. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473):45–51, 2020.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

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*, pp. 1–10, 2025.

Rijie Xi, Weikang Xu, Wei Xiong, Yuannong Ye, and Bin Zhao. Gene regulatory network inference algorithm based on spectral signed directed graph convolution. *arXiv preprint arXiv:2512.11927*, 2025.

Chenglei Yu, Chuanrui Wang, Bangyan Liao, and Tailin Wu. scdfm: Distributional flow matching model for robust single-cell perturbation prediction. *arXiv preprint arXiv:2602.07103*, 2026.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Han Zhang, Guo-Hua Yuan, Chaohao Yuan, Tingyang Xu, Tian Bian, Hong Cheng, Wenbing Huang, Deli Zhao, and Yu Rong. Lingshu-cell: A generative cellular world model for transcriptome modeling toward virtual cells. *arXiv preprint arXiv:2603.25240*, 2026.

A Implementation Details

Architecture. Table 5 lists the configuration used for all main results. Gene tokens have width $d = 128$ and every contraction in a layer maps to $d_{\text{hid}} = 512$. The model has $L = 8$ distinct signed blocks followed by $T = 8$ applications of one shared block, giving a nominal reach of 16 hops with the parameters of 9 blocks. Expression encoders f_{ϕ_1}, f_{ϕ_2} map a scalar to $\mathbb{R}^d$ through $\mathrm{Linear}(1, d)$, GELU, dropout, $\mathrm{Linear}(d, d)$ and layer normalization. MLP_{in} maps $3d \to d \to d$ with GELU and a final layer normalization. The time embedding is a 256-dimensional sinusoidal embedding followed by a two-layer SiLU MLP, and the condition vector $\mathbf{q}$ is a two-layer SiLU MLP on $[\,\cdot \,\|\, \mathbf{p}\,]$. The neighbor gate MLP_η maps $2d \to d_{\text{hid}} \to 2d$ with SiLU and a sigmoid output, split into $\boldsymbol{\eta}^+, \boldsymbol{\eta}^- \in \mathbb{R}^{B\times d}$. The gates vary with cell, condition and time and are shared across the genes of a cell, so gene specificity enters only through $\mathbf{m}_i^{\pm}$. MLP_g maps $3d \to d_{\text{hid}} \to d$ with GELU. The decoder is a three-layer LeakyReLU MLP on $[\,\mathbf{h}_i \,\|\, \mathbf{p}\,]$, and the velocity gate is a two-layer SiLU MLP on the same input with a sigmoid output. Linear layers use Xavier-uniform initialization with zero bias. Block contractions are initialized with truncated normal s.d. 0.02 and the expansion W_{exp} with s.d. 0.01. The adaptive modulation map is zero-initialized. The recurrence coefficient is $\kappa = \sigma(\ell_\kappa)$ with ℓ_κ initialized to -2.1972, so $\kappa = 0.1$ at initialization.

Table 5: Architecture hyperparameters.

Hyperparameter	Value
Token width d	128
Contraction width d_{hid}	512
Distinct signed blocks L	8
Shared recurrent steps T	8
Residual scale β	0.1
Initial recurrence coefficient κ	0.1
Dropout	0.1
Spectral coordinates K_s	32
Neighbors per gene K	30
Empty-row guard δ	10^{-6}
Action projection init s.d.	10^{-2}
Velocity gate bias init	-2
Parameters	≈10–13M
Parameters per block	≈1.0M

Gene graph and spectral coordinates. The vocabulary contains the 5,000 most variable genes in training cells, plus all Norman targets and all evaluated genes. Pearson correlation is computed on the log-normalized expression of the training split, including controls. NaN entries from constant genes are set to zero. The neighbor table is the symmetrized top-K co-expression graph with $K = 30$, capped at 128 neighbors per gene; at $K = 30$ no gene exceeds the cap, so the table is the symmetrized top-K graph. Signed edge weights are read from the correlation matrix at those entries. In the unsigned ablation, every retained edge has weight one in a single channel and the negative channel is empty; the sampler then reserves each target's first K neighbors in token order. Spectral coordinates are the eigenvectors of the $K_s = 32$ smallest nonzero eigenvalues of the normalized Laplacian of the same graph, computed on training cells and cached.

Gene sampling. Each training batch uses $N = 1{,}000$ gene columns. For genetic perturbations, the sampler first reserves every target that is in the vocabulary. It then adds up to $K = 30$ neighbors per

target, ranked by $|R_{ij}|$ and excluding the targets themselves. If the neighbors exceed the remaining budget, a random subset is kept. The rest of the batch is filled with uniformly sampled columns, and the selection is shuffled. Norman conditions have at most two targets, so reserved columns number at most 62 and target coverage is one in every batch. For drug and cytokine datasets, all N columns are sampled uniformly.

Training. Table 6 lists the optimization settings. All cells in a batch share one condition, and each perturbed cell is paired with a control of the same cell type. Gradients are clipped to norm 1. Training runs in bfloat16 mixed precision on a single GPU. The endpoint weight follows $\lambda_k = 0.1 \min(1, k/15{,}000)$. The Sinkhorn divergence uses entropic regularization 0.1 times the median batch cost and 50 log-domain iterations. It is skipped for a batch whose condition has no DEG weight vector. DEG weights use a floor of 0.1, giving $w_i \in [0.1, 1]$. Checkpoints are evaluated every $10{,}000$ steps. Reported numbers use the final checkpoint at $300{,}000$ steps. The curves in Fig. 3 are computed on the held-out conditions and were not used to select checkpoints.

Table 6: Training hyperparameters, shared across datasets.

Hyperparameter	Value
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$)
Learning rate	5×10^{-5}
Schedule	cosine annealing to 10^{-6}
Training steps	300,000
Cells per batch	48
Genes per batch N	1,000
Gradient clipping	1.0
Precision	bfloat16
Endpoint weight λ	0.1
Endpoint warm-up	15,000 steps
Sinkhorn ε scale	0.1
Sinkhorn iterations	50
DEG weight floor	0.1
Source distribution	$\mathcal{N}(\mathbf{0}, \mathbf{I})$
Seeds	42, 43, 44
Hardware	$1\times$ H100 GPU
Wall-clock per run SPARC	$\approx$4 hours
Wall-clock per run scDFM	$\approx$18 hours

Sampling. Generation integrates v_θ from Gaussian noise with a fixed-step RK4 solver over 20 uniformly spaced time points on $[0, 1]$, which gives 19 steps and 76 velocity evaluations per sample. Negative values of the final state are set to zero. Each of the 128 generated cells per condition is conditioned on a control cell drawn uniformly without replacement from the full control pool.

Ablation settings. Each ablation changes one factor from the configuration above and is a single run with seed 45, disjoint from the seeds of the main tables, so full-model values in Fig. 3 differ from the table means. Endpoint, conditioning and sampling ablations are also run on the Norman holdout split for Fig. 4A,B.

- Broadcast conditioning removes the scattered term of Eq. 6.
- Uniform sampling replaces the action-aware sampler with uniform column sampling.
- The Poisson source replaces $\mathbf{x}_0$ with a log-normalized Poisson–Gamma (negative binomial) resampling of the paired control cell with dispersion $\alpha = 0.8$, at both training and generation.
- The ungated variant removes MLP_η, so $\boldsymbol{\eta}^{\pm} = \mathbf{1}$.
- Field-size variants change $N \in \{500, 1000, 2000\}$ during training only.
- The recurrent-depth variants use $T \in \{4, 8\}$.
- Endpoint variants replace DEG-weighted Sinkhorn with unweighted Sinkhorn, MMD, or DEG-weighted MMD, all at the same λ and warm-up.

Learned recurrence coefficient and model size. At the final checkpoint, $\kappa = \sigma(\ell_\kappa) \in [0.10, 0.13]$ across datasets, endpoint objectives and seeds. The recurrent gene update is $\kappa\beta \tanh(\boldsymbol{\gamma}_h) \odot$

$W_{\text{exp}}\text{GELU}(\mathbf{z})$ with W_{exp} and $\boldsymbol{\gamma}_h$ trained jointly, so the magnitude of the update is determined by the product and not by κ; the value of κ is not a measure of how much the recurrent block changes the gene states. The shared block has ≈ 1.0M parameters, the same as each distinct block, so recurrence adds 8–10% of the model's parameters independent of T. The $L = 8$ distinct blocks account for 62–80% of all parameters.

B Evaluation Protocol

Shared benchmark protocol. Results in Tables 1–4 follow the benchmark protocol of Yu et al. (2026) for Norman and ComboSciPlex and of Zhang et al. (2026) for Parse-PBMC, which in turn follows the Parse-PBMC setup of Adduri et al. (2026). Datasets, preprocessing, train/test splits, held-out conditions, seeds, the evaluated gene set, the control pool, the number of generated cells per condition, and the cell-eval implementation (Roohani et al., 2025) are taken unchanged from these works. Baseline numbers are reported from the cited papers and SPARC is evaluated under the same protocol, so numbers within a table are comparable. Fig. 4C,D require generated cells, which published numbers do not provide; for those analyses we train scDFM with its released code and configuration on the same split and generate cells under the protocol used for SPARC. The 10 evaluated conditions per Norman split, a fixed subset of each split's held-out conditions, and the 7 held-out ComboSciPlex conditions are the benchmark's fixed test sets.

Evaluated gene set and input tokens. The 1,000 evaluated genes are the top 1,000 highly variable genes, fixed per dataset by the benchmark protocol (Ahlmann-Eltze et al., 2025; Yu et al., 2026). The set plays two roles for gene-token models. It defines where metrics are computed, as for every method, and it defines the token axis at evaluation, so scDFM and SPARC condition on the same genes. Whether a perturbation target falls inside this set is therefore fixed by the benchmark not by the model: on Norman, targets that lie in the evaluated set receive the localized term of Eq. 6 and targets outside it enter only through $\mathbf{p}$. The localized term is a function of target identity, which is the condition every method receives, and of the gene encoder; it reads no expression value from the evaluated cells. Metric computation, DE calling and discrimination are unchanged from the benchmark.

Behavior of the signed operator under gene subsampling. The operator is defined on a gene subset not on the full vocabulary, so it is worth stating what changes with $\mathcal{S}$ and what does not. Let $s_i^{\pm} = \sum_j w_{ij}^{\pm} \mathbf{1}[j \in \mathcal{N}_{\mathcal{S}}(i)]$. Then $\sum_j \tilde{A}_{ij}^{\pm} = s_i^{\pm} / \max(s_i^{\pm}, \delta) \leq 1$, with equality when $s_i^{\pm} \geq \delta$. In these rows, the gathered messages $\mathbf{m}_i^{\pm} = \sum_j \tilde{A}_{ij}^{\pm} \mathbf{h}_j$ are convex combinations of neighbor states; for $0 < s_i^{\pm} < \delta$, the weights sum to less than one. Normalization avoids unnormalized accumulation with degree, but subsampling can change both message direction and magnitude. Layer normalization reduces sensitivity to scale without making the messages invariant to the gene subset. Rows with no retained neighbor of a sign give $\mathbf{m}_i^{\pm} = \mathbf{0}$. The global state behaves the same way: $\mathbf{g}$ is read by every consumer through $\text{LN}(\mathbf{g})$ with adaptive modulation, and the pooled means $\frac{1}{N}\sum_i \mathbf{h}_i$, $\frac{1}{N}\sum_i \mathbf{u}_i$ do not accumulate with N. Changing $\mathcal{S}$ from a uniform training sample to V changes which genes the pool summarizes and can shift these means; layer normalization of $\mathbf{g}$ reduces the effect on scale but does not remove it.

Semantics of the signed channels. Edge weights $w_{ij}^{\pm}$ are nonnegative, so $\mathbf{m}_i^{+}$ and $\mathbf{m}_i^{-}$ are both convex combinations of neighbor states. Sign enters through routing: negatively correlated neighbors reach gene i only through $\boldsymbol{\eta}^{-}$ and W_{-}, positively correlated ones only through $\boldsymbol{\eta}^{+}$ and W_{+}. W_{-} is learned and unconstrained, so the operator does not impose negation; it gives the two co-variation types separate transforms, as in signed and relational message passing (Derr et al., 2018; Schlichtkrull et al., 2018). A single unsigned channel applies one transform to both types.

Under uniform sampling without replacement, conditional on $i \in \mathcal{S}$, a gene with $\deg^{\pm}(i)$ signed neighbors retains $\deg^{\pm}(i)(N-1)/(G-1)$ neighbors in expectation. For positive degree, the expected retained fraction is $(N-1)/(G-1) \approx N/G \approx 0.2$. The evaluated gene set is the top 1,000 highly variable genes, not a uniform sample, so retention at evaluation can differ from training. Random training subsets vary graph support by design; Fig. 3C varies N. Action-aware sampling explicitly reserves targets and selected neighbors, at most 62 of 1,000 columns on Norman. Uniform sampling instead includes a given target with probability N/G, so its localized pathway receives direct supervision in about one batch in five (Fig. 3B).

What Spearman LFC and Pearson Δ measure. The two metrics are computed over different gene sets and different statistics. Spearman LFC is a rank correlation between predicted and observed log-fold changes restricted to the genes called significantly DE in the observed data, a subset of the $1{,}000$ evaluated genes. Pearson Δ is a linear correlation of mean shifts from control over all $1{,}000$. The first reads the ordering of response magnitudes among genes that respond; the second reads a transcriptome-wide shift vector whose entries are dominated by the large majority of genes with near-zero shift. The two can move independently, and on Norman they do.

They also differ in resolution on these benchmarks. On Norman additive, every method except scGPT, CPA and STATE lies in $[0.742, 0.885]$ on Pearson Δ and the top three lie within 0.017 of each other, so the statistic is compressed near the top of its attainable range: the 0.009 separating scDFM (0.885 ± 0.011) from SPARC (0.876 ± 0.012) is smaller than the reported seed s.d. of either method, while the 0.033 discrimination gap is of the same order as the reported across-seed variation. Because the baseline and SPARC results use different seeds and run counts, we treat these comparisons descriptively rather than as significance tests. Spearman LFC over the same conditions spans $[0.007, 0.884]$ across methods, and the gap between SPARC and the next method (0.313) is an order of magnitude larger than any reported s.d. On the holdout split, Fig. 4B shows per perturbation that the full model's advantage over its ablations is concentrated on FOSB+UBASH3B and MAP2K3+MAP2K6, where variants diverge, rather than spread evenly across conditions.

Reading DE recall. DE recall is the fraction of observed DE genes also called DE in the prediction. It does not penalize genes called DE in the prediction but not in the observed data, and generated cells can differ from measured cells in distributional shape (continuous values, no dropout structure, a point mass at zero from clipping the final sample) in ways a rank-sum test against measured controls detects independently of the response. The Control row, which contains no response, reaches 0.681 on Norman additive, 0.675 on the holdout split and 0.715 on ComboSciPlex, which sets the level above which recall reflects response. DE recall is part of the fixed benchmark protocol and we report it for comparability, but we treat it as secondary and base no claim on it. SPARC's DE recall on ComboSciPlex and Parse-PBMC (0.973, 0.995) is likewise not read as evidence of response accuracy.

TRRUST precision. We use human TRRUST v2 (Han et al., 2018) to assess whether gene relationships recovered by each model agree with curated transcriptional regulation. Let V denote the evaluated gene set and let $\mathcal{F}_V \subseteq V$ be the genes annotated as transcription factors in TRRUST. The reference set

$$\mathcal{T}_V = \{(f, g) \in \text{TRRUST} : f \in \mathcal{F}_V,\ g \in V,\ f \neq g\}$$

contains directed TF–target pairs whose two genes are both represented in the evaluation.

For each model, we pool the generated cells of the 10 evaluated conditions of the Norman additive split ($1{,}280$ cells) and compute the Pearson correlation R_{fg} across these cells, setting entries for constant genes to zero. For every $f \in \mathcal{F}_V$, candidate targets $g \in V$, $g \neq f$, are ranked by $|R_{fg}|$ with ties broken by gene index, and the top 30 are retained, matching the partner budget $K = 30$ of the static prior. The resulting directed edge set is denoted by E. The prior applies the same procedure to training cells. Because cells are pooled across conditions, these correlations include between-condition response covariation as well as within-condition co-expression. Selecting 30 targets per transcription factor uniformly at random gives expected precision $|\mathcal{T}_V|/(|\mathcal{F}_V|(|V| - 1))$. We then report

$$\text{Precision} = \frac{|E \cap \mathcal{T}_V|}{|E|}.$$

Values in Fig. 4C are reported as percentages. Thus, the metric measures the fraction of candidate TF–target relationships supported by existing regulatory annotation rather than the fraction of the full TRRUST network recovered.

Because TRRUST is sparse and incomplete, the absolute precision should not be interpreted as the fraction of biologically valid interactions learned by the model. We instead use it as a comparative measure of agreement with curated TF–target relationships across models and ablations.

GSEA pathway recall. For each evaluated condition, genes in V are ranked by log-fold change against control, once from observed cells and once from generated cells, and preranked GSEA is run on each ranking against the MSigDB Hallmark gene sets. Recall is the fraction of the 20 top-ranked

pathways of the observed ranking that also appear among the 20 top-ranked pathways of the predicted ranking, averaged over conditions.

C Scope of the Evaluation and Open Measurements

The protocol of Sec. 4 and App. B is taken unchanged from Yu et al. (2026), Zhang et al. (2026) and Adduri et al. (2026), which fixes what is comparable across methods. Four properties of the protocol bear on how the results should be read; protocol-inherited choices and disclosed limitations are collected in App. D.

Target coverage of the evaluated gene set. The evaluated gene set V is the top $1{,}000$ highly variable genes, selected once per dataset by the benchmark; the gene vocabulary is the $5{,}000$ most variable genes plus all Norman targets and all evaluated genes. Whether a CRISPRa target lies in V is a property of the benchmark and is the same for every method. Eq. 6 is sensitive to this split: a target in V occupies a column of $\mathcal{S}$ at evaluation and receives its own action vector, while a target outside V contributes only through $\mathbf{p}$, so the conditioning reduces to broadcast for that target.

The localized term and the action-aware sampler act at different times, and this distinction matters for how the two should be credited. The sampler reserves every in-vocabulary target column in every training batch regardless of V, so the gene encoder receives gradient at target nodes under all training conditions; the localized term additionally places an action vector at those nodes at inference, for whichever targets V happens to cover. The conditioning and sampling ablations are pooled over held-out conditions in Fig. 3B and reported per condition, without grouping by coverage, in Fig. 4B, so both mix both regimes. A finer reading requires grouping the held-out conditions by whether their targets are covered and reporting Spearman LFC and direction match within each group for the full model and the broadcast ablation, which would separate the inference-time contribution of the action vector from the training-time contribution of column reservation. Fig. 3B bounds both: at $300{,}000$ steps, broadcast conditioning reaches 0.82–0.85 Spearman LFC and uniform sampling 0.77–0.80, versus 0.87 for the full model, so neither adds over 0.10 and each surpasses every baseline. On ComboSciPlex and Parse-PBMC no target is a gene token, both mechanisms are inactive, and $\mathbf{a}_i = \mathbf{p}$ throughout.

Baseline numbers and the DE call. Baseline numbers in Tables 1–4 are taken from Yu et al. (2026) and Zhang et al. (2026). SPARC is evaluated on the same splits, held-out conditions, evaluated gene set, control pool and cell-eval implementation, so every entry of a table uses the same metric definitions on the same test conditions. Spearman LFC and direction match are restricted to genes called DE in the observed data; the call depends only on the observed cells and the fixed rank-sum test, both shared across a table. Pearson Δ and MSE are computed over all $1{,}000$ evaluated genes with no restriction step. Reported values are used so no baseline depends on our retraining; the scDFM retrain from its released code only generates cells for Fig. 4C,D.

Dispersion of generated cells. DE recall does not penalize genes called DE only in the prediction, so it is not read as evidence of response accuracy (App. B). Discrimination and MSE are computed on pseudobulk means and do not measure per-gene dispersion, so they are not used to interpret it either. On the additive split the gaps to scDFM on Pearson Δ and discrimination are 0.009 and 0.033, both small relative to the reported across-seed variation. On the holdout split SPARC trails scDFM on discrimination, 0.829 against 0.891, a gap of 0.062, which is larger than either method's reported seed s.d. (0.015 and 0.014); we report this descriptively as a loss. Fig. 4A shows that a smaller off-support mode persists under the Sinkhorn endpoint, which shifts predicted pseudobulk away from the observed one and is consistent with lower discrimination on a split whose held-out targets are absent from training.

Control pairing. Training pairs each perturbed cell with a control of the same cell type, while evaluation draws controls from the full pool, as in Yu et al. (2026). Evaluation therefore conditions the model on controls of any cell type, a pairing not seen in training. This is shared with the benchmark protocol and is a train/test conditioning mismatch rather than a property of SPARC.

D PROTOCOL-INHERITED CHOICES AND DISCLOSED LIMITATIONS

This section collects choices that are fixed by the benchmark protocol or disclosed elsewhere in the paper, with the reason for each.

Held-out curves and checkpoints. Fig. 3 tracks metrics on the benchmark's held-out conditions over training, as a diagnostic of training dynamics. Every number in Tables 1–4 uses the fixed final checkpoint at $300{,}000$ steps, no checkpoint is selected from these curves, and all datasets share Table 6, not tuned per dataset.

Baseline variance. Baseline means and s.d. are taken from Yu et al. (2026) and Zhang et al. (2026), which report them for their own baselines in the same way. Their seeds and run counts are those of the source papers and differ from ours (42–44). App. B and App. C compare gaps against these s.d. values only to qualify gaps where SPARC trails, never to claim a gain.

Parse-PBMC table. Held-out conditions, rows and metrics follow Zhang et al. (2026), and we keep its rows unchanged.

TRRUST and GSEA reference levels. Both analyses are comparative. Every model and ablation is scored by the same procedure on the same gene set V, conditions and number of generated cells, so the level expected under random selection is common to all entries and does not change their ordering. Absolute values are not interpreted (App. B). The random-selection expectation for TRRUST precision is given in App. B.

Evaluation token axis. At evaluation $\mathcal{S} = V$ for SPARC, as for scDFM. Extending $\mathcal{S}$ with targets or neighbors outside V would give SPARC input columns that the gene-token baselines do not condition on. Targets outside V therefore receive broadcast conditioning at inference (App. C).