PERTURBRIFT: RESPONSE-INFORMED FLOW MATCHING FOR PREDICTING SINGLE-CELL RESPONSES TO UNSEEN PERTURBATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting single-cell responses to unseen genetic perturbations requires modeling both population-level expression changes and cell-to-cell variability. Here we introduce PerturbRIFT, a generative framework that incorporates measured response statistics into a perturbation-specific source distribution for conditional flow matching. Gene embeddings and knowledge graphs guide the learned aggregation of gene-wise means, zero probabilities, and nonzero-expression variances from related training perturbations. These statistics parameterize a factorized zero-inflated normal (ZINormal) prior, providing a query-specific starting point for generation. Residual latent flow matching then transforms samples from this prior into predicted single-cell response populations, with a latent maximum mean discrepancy objective aligning transported and observed populations during training. Across four within-dataset unseen-perturbation benchmarks in K562, RPE1, HepG2, and Jurkat, PerturbRIFT improves mean-response accuracy and multiple distributional metrics relative to the published prediction baselines evaluated here. K562 ablations show that biologically guided prior estimation contributes the largest gains in mean-response accuracy, while residual flow matching with latent distribution alignment improves multiple measures of single-cell distributional fidelity. Together, our findings support using measured response statistics to define generative source distributions for predicting single-cell responses to unseen perturbations.

1 INTRODUCTION

Single-cell perturbation assays couple targeted genetic interventions with transcriptome-wide readouts, enabling direct characterization of how gene knockout, knockdown, or activation reshapes cellular state (Dixit et al., 2016; Datlinger et al., 2017). However, experimentally profiling the complete intervention space remains infeasible. Each perturbation requires sufficient cell numbers and sequencing depth across relevant cellular contexts, and the intervention space expands further when combinations are considered. Predicting responses to experimentally unobserved perturbations is therefore a central task in virtual-cell modeling. Beyond estimating average expression changes, single-cell prediction requires characterizing how these responses vary across individual cells.

Existing methods represent unseen perturbations using response-derived features, biological knowledge graphs, or embeddings obtained from language models and pretrained single-cell models. These representations encode useful biological information, but do not by themselves specify the distribution of transcriptional responses to a target perturbation in a given cellular context. Predictive models must learn this relationship from the available perturbation experiments. Controlled evaluations have shown that simple linear models using pretrained representations can match or outperform corresponding deep-learning predictors on unseen perturbations (Ahlmann-Eltze et al., 2025). These findings motivate closer attention to how observed response information is incorporated into predictive models, rather than relying on increased model capacity alone.

Building on this perspective, we introduce PerturbRIFT (**R**esponse-**I**nformed **F**low **T**ransport), a framework that combines response-prior estimation with conditional single-cell generation. Biological relationships guide the selection and weighting of relevant training perturbations, whose measured response statistics provide the basis for a query-specific prior. The prior and the generative model

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.

serve complementary roles: the former encodes estimated gene-wise response properties, while the latter learns a conditional transformation through residual latent flow matching to model joint expression patterns across individual cells. Experimental response information thus informs both the distribution from which generation begins and the context guiding its transformation, connecting measured responses from related experiments to the prediction of unseen response populations.

Our contributions are threefold:

1. We develop relation-guided aggregation of measured response statistics to construct query-specific generative priors.
2. We integrate these priors with conditional latent flow matching and population-level distribution alignment.
3. We evaluate unseen-perturbation prediction in K562, RPE1, HepG2, and Jurkat, and use K562 ablations to distinguish the contributions of prior estimation, residual generation, and output sampling.

2 Related Work

2.1 Biological prior representations

Prior-informed perturbation models incorporate biological information through graphs, language-derived embeddings, pretrained transcriptomic representations, or latent intervention factors. GEARS (Roohani et al., 2024) uses co-expression and Gene Ontology graphs, while TxPert (Wenkel et al., 2026) integrates multiple knowledge graphs across perturbations and cellular contexts. Scouter (Zhu & Li, 2026), scLAMBDA (Wang et al., 2024), and PRESAGE (Littman et al., 2025) use language-derived gene representations for discriminative or generative response prediction. scGPT (Cui et al., 2024), Geneformer (Theodoris et al., 2023), and scFoundation (Hao et al., 2024) instead leverage representations pretrained on large transcriptomic datasets.

2.2 Response-informed prediction

Methods that explicitly reuse observed perturbation responses incorporate experimental information at different stages of prediction. LangPert (Märtens et al., 2025) retrieves related interventions and averages their pseudobulk responses, while SLIM (Hu et al., 2026) predicts a mean response from STRING embeddings and rescales sampled training cells to that mean. PT-RAG (Di Francesco et al., 2026) uses GenePT retrieval and a cell-state-aware selector to condition a STATE-based generator, and DELPHAI (Zhang et al., 2026) combines cell-fitness filtering with retrieval in expression space. In PerturbRIFT, aggregated response statistics parameterize the generative source distribution itself, while also informing the conditioning context.

2.3 Single-cell distribution modeling

Optimal transport and generative models provide approaches to learning response distributions from unpaired single-cell populations. CellOT (Bunne et al., 2023) and CINEMA-OT (Dong et al., 2023) construct population mappings or couplings, while Squidiff (He et al., 2026), CellFlow (Klein et al., 2025), and PRiMeFlow (Yan et al., 2026) learn conditional generative dynamics. scDFM (Yu et al., 2026) combines expression-space flow matching with population-level MMD and co-expression-guided attention. GeneGeoFlow (Li et al., 2026) learns a biologically conditioned residual path from noise-perturbed control cells.

3 Method

Problem setup. Let $\mathcal{C}_{\mathrm{tr}}$, $\mathcal{C}_{\mathrm{val}}$, and $\mathcal{C}_{\mathrm{te}}$ denote disjoint sets of perturbation conditions. For condition c, the observed population is $X_c = \{x_{c,i}\}_{i=1}^{N_c}$, where $x_{c,i} \in \mathbb{R}^G$ is a normalized, $\log(1+x)$-transformed expression vector over G modeled genes. We additionally have a control population with mean $m_{\mathrm{ctrl}} \in \mathbb{R}^G$ and a fixed external embedding $e_c \in \mathbb{R}^E$ for each perturbation target. For a test query $q \in \mathcal{C}_{\mathrm{te}}$, the response cells X_q are unavailable during training and prediction. Our objective is

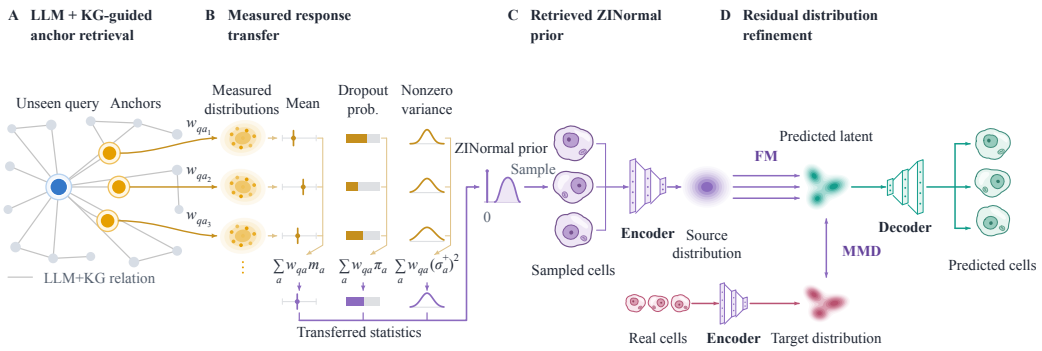


Figure 1: **Overview of PerturbRIFT.** (A) Biological relationships guide the selection of related training perturbations for an unseen query, while gene embeddings and knowledge-graph features encode query–anchor relationships. (B) Learned query–anchor weights aggregate gene-wise means, zero probabilities, and nonzero-expression variances from measured training responses. (C) These statistics parameterize a query-specific ZINormal source distribution, providing a response-informed starting point for conditional generation. (D) During training, a frozen encoder maps prior samples and observed response cells into latent space for conditional residual flow matching and latent MMD alignment. At inference, prior samples are encoded, transported by the learned flow, and decoded into predicted single-cell expression profiles.

to model the response distribution for q and generate a population of predicted single-cell expression profiles.

Model overview. PerturbRIFT estimates a query-specific response prior and transforms samples from this prior through a conditional latent flow (Figure 1). The autoencoder, anchor-weighting networks, and residual flow are trained in separate stages, with the first two components frozen during residual learning. Complete training and inference procedures are provided in Algorithms 1 and 2 in Appendix B.5.

3.1 Relation-guided response weighting

For query q, we construct a candidate pool containing only training perturbations, excluding q and interventions on the same target gene. Eligible perturbations are ranked using STRING associations, CollecTRI regulatory relationships, and shared pathway membership. The top A perturbations form the anchor set $\mathcal{A}(q)$. The ranking procedure and query–anchor relation features are specified in Appendix B.1.

Within this biologically defined neighborhood, gene embeddings and relation features determine query-specific aggregation weights. A shared embedding encoder and a relation encoder produce

$$h_q = f_e(e_q), \qquad h_a = f_e(e_a), \qquad h_{qa} = f_r(r_{qa}), \tag{1}$$

where r_{qa} denotes the relation feature vector for query q and anchor a. The anchor logit and normalized weight are

$$\ell_{qa} = f_w([h_q, h_a, h_{qa}]), \qquad w_{qa} = \frac{\exp(\ell_{qa}/\tau)}{\sum_{b\in\mathcal{A}(q)} \exp(\ell_{qb}/\tau)}, \tag{2}$$

where τ is a fixed temperature. The final scoring layer is initialized to zero, yielding uniform initial weights. These weights determine each anchor's contribution to the response statistics used to parameterize the query-specific prior.

3.2 Response-informed ZINormal prior

For anchor a, let m_a denote its population mean and $\delta_a = m_a - m_{\text{ctrl}}$ its control-relative mean response. We additionally estimate its gene-wise zero probabilities π_a and conditional nonzero-

expression variances $(\sigma_a^+)^2$. We define

$$\delta_0(q) = \sum_{a \in \mathcal{A}(q)} w_{qa}\delta_a, \qquad m_0(q) = m_{\text{ctrl}} + \delta_0(q), \tag{3}$$

$$\pi_0(q) = \sum_{a \in \mathcal{A}(q)} w_{qa}\pi_a, \qquad (\sigma_0^+(q))^2 = \sum_{a \in \mathcal{A}(q)} w_{qa}(\sigma_a^+)^2. \tag{4}$$

These parameters specify the expression levels, zero probabilities, and nonzero-expression variability of a query-specific source distribution. Because $m_0(q)$ is the unconditional mean, the conditional nonzero mean is

$$\mu_0^+(q) = \frac{m_0(q)}{1 - \pi_0(q)}, \tag{5}$$

where division is elementwise. Small positive lower bounds are imposed on nonzero-expression probabilities and variances for numerical stability.

We parameterize the source as a factorized zero-inflated normal (ZINormal) distribution over normalized, log1p-transformed expression (Finak et al., 2015). For a given query, source cells are sampled independently across genes as

$$\begin{aligned} b_{i,g} &\sim \text{Bernoulli}(1 - \pi_{0,g}), \qquad y_{i,g} \sim \mathcal{N}(\mu_{0,g}^+, (\sigma_{0,g}^+)^2), \\ x_{i,g}^{(0)} &= b_{i,g} y_{i,g}. \end{aligned} \tag{6}$$

This parameterization provides a sampleable, response-informed starting distribution for conditional generation.

Response information also enters the conditioning of the residual flow. The query representation, weighted anchor and relation representations, and encodings of δ_0, π_0, and $\log\big((\sigma_0^+)^2\big)$ are combined into a context vector c_q.

For a training query q, let m_q be its observed population mean and $\delta_q = m_q - m_{\text{ctrl}}$ its control-relative mean response. The anchor-weighting networks are trained using mean-response supervision:

$$\mathcal{L}_{\text{prior}} = \lambda_\rho \left[1 - \rho(\delta_0(q), \delta_q)\right] + \lambda_1 \|m_0(q) - m_q\|_1 / G, \tag{7}$$

where ρ denotes Pearson correlation across genes. The anchor-weighting checkpoint is selected using validation conditions. The selected parameters are combined with the frozen autoencoder checkpoint before residual training.

3.3 Conditional latent flow matching

We learn a conditional transformation of the response-informed source distribution in a latent space. For a cell population $X \in \mathbb{R}^{M \times G}$ with nonzero mask $B = \mathbf{1}[X \neq 0]$, separate expression and mask branches produce per-cell representations $Z = E_\phi(X, B) \in \mathbb{R}^{M \times D}$. The encoder and a probabilistic ZINormal decoder are pretrained using the reconstruction likelihood described in Appendix B.2. Both remain frozen during residual training.

For a training query q, we sample a source population X_0 from Equation (6) and an equally sized population X_1 from the observed response cells. Entropic optimal transport in expression space provides a stochastic coupling for source–target pairing. The pairing procedure is detailed in Appendix B.3. The frozen encoder maps the paired populations to $Z_0 = E_\phi(X_0, B_0)$ and $Z_1 = E_\phi(X_1, B_1)$. For $t \sim \mathcal{U}(0, 1)$, we define the linear conditional path and its target velocity as

$$Z_t = (1 - t)Z_0 + tZ_1, \qquad u_t = Z_1 - Z_0. \tag{8}$$

The velocity field $v_\theta(Z_t, t, c_q, m_0)$ is parameterized by a set Transformer that attends across cells. It takes the current latent population, time, response-informed context c_q, and prior mean $m_0(q)$ as inputs. The conditional flow-matching objective is

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{q, X_0, X_1, t}\left[\|v_\theta(Z_t, t, c_q, m_0) - (Z_1 - Z_0)\|_2^2\right]. \tag{9}$$

To additionally align endpoint populations, we integrate the current velocity field to obtain transported latents $\widehat{Z}_1$ and compute a multi-scale latent RBF MMD against the stop-gradient target latents Z_1. The combined residual objective is

$$\mathcal{L}_{\text{res}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{MMD}}\mathcal{L}_{\text{MMD}}, \tag{10}$$

with the biased MMD V-statistic specified in Appendix B.3. Only the velocity network and prior moment-to-context encoders are updated at this stage; the autoencoder and anchor-weighting networks remain frozen.

At prediction time, source samples are encoded and transported by the learned velocity field. The decoder maps the transported latents to cell-specific ZINormal parameters: a zero probability $\hat{\pi}_{i,g}$, a raw expected-expression value $\tilde{\mu}_{i,g}$, and a nonzero-expression scale $\hat{\sigma}^{+}_{i,g}$. For a prescribed population mean m, expected expression is calibrated as

$$\hat{m}_{i,g} = \tilde{\mu}_{i,g} \frac{m_g}{M^{-1} \sum_{j=1}^{M} \tilde{\mu}_{j,g}}, \qquad \hat{\mu}^{+}_{i,g} = \frac{\hat{m}_{i,g}}{1 - \hat{\pi}_{i,g}}. \tag{11}$$

We use $m = m_0(q)$ when generating predictions. Under the decoder distribution before output clipping, this gives

$$\frac{1}{M} \sum_{i=1}^{M} \mathbb{E}[\hat{x}_{i,g} \mid \widehat{Z}_1, m_0(q)] = m_{0,g}(q). \tag{12}$$

The calibration retains the prior mean in expectation while allowing the transported latents to determine cell-specific expression, zero probabilities, and nonzero-expression variability. Finite-sample means can deviate from $m_0(q)$ because of sampling variability and output clipping.

3.4 Training and inference

Training proceeds in three stages: autoencoder pretraining, response-prior estimation, and residual flow learning. At inference, samples from the query-specific prior are encoded, transported by the conditional flow, and decoded into predicted single-cell expression profiles. The complete procedures are provided in Algorithms 1 and 2 in Appendix B.5.

4 Experiments

4.1 Experimental setup

Datasets and splits. We evaluate unseen-perturbation prediction on four datasets: Replogle K562 and RPE1, and Nadig HepG2 and Jurkat. Within each dataset, we restrict the benchmark to perturbations covered by GenePT and the required graph resources. All methods use the same five perturbation-level train–validation–test splits and are evaluated on identical perturbation and gene panels. Preprocessing, resource filtering, and split construction are detailed in Appendix C.1.

Baselines. We compare PerturbRIFT with statistical controls, linear models, and neural perturbation predictors. These include no-change and training-mean controls; linear models RidgeSC and SLIM; graph- and embedding-informed predictors GEARS, TxPert-public, and Scouter; Transformer-based models scGPT and STATE; and generative models CellFlow and PerturbDiff. Method-specific training adaptations are summarized in Appendix C.2.

Mean-response metrics. We compare predicted and observed condition-level mean expression using control-centered Pearson correlation (ρ_{Δ}), Pearson correlation centered on the training-perturbation mean ($\hat{\rho}_{\Delta}$) (Viñas Torné et al., 2026), normalized mean squared error (NormMSE), and mean absolute error (MAE). Both Pearson metrics and NormMSE are evaluated on query-specific Top100 DEG panels and the full modeled gene panel (allHVG), with MAE additionally reported on Top100 genes. We also report an L1-based perturbation discrimination score (PDS), which measures how highly each prediction ranks its matching observed response among the test perturbations (Roohani et al., 2025).

Single-cell distribution metrics. For methods that generate cell populations, we evaluate marginal expression fidelity using gene-wise Wasserstein distance (GeneW1) and mean absolute errors in gene-wise standard deviations (StdMAE) and zero fractions (ZeroRateMAE). Population-level distributional agreement is assessed using expression-space MMD, PCA64MMD, and sliced Wasserstein distance along 64 random projections (SWD64). PCA64MMD computes MMD in a shared 64-dimensional PCA space fitted exclusively on training cells (Schäfer et al., 2026). We additionally assess gene dependencies by measuring agreement between predicted and observed correlation changes relative to training controls.

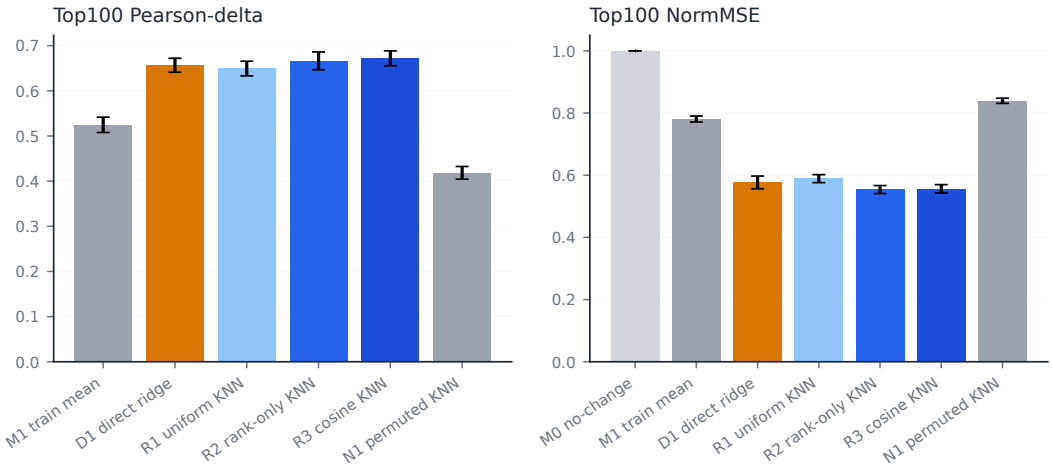


Figure 2: **Local embedding structure and mean-response prediction.** Replogle K562 Top100 results are reported as mean ± SD of condition-averaged scores across five perturbation splits. D1 fits a direct ridge decoder on GenePT embeddings. R1–R3 aggregate measured mean-response deltas from 16 GenePT neighbors using uniform, inverse-rank, and cosine-softmax weights, respectively. N1 permutes gene–embedding assignments and averages inverse-rank aggregation scores over 20 permutations within each split before the split-level summary. M0 and M1 are no-change and training-mean controls; M0 has undefined Pearson delta and is omitted from the left panel. Higher Pearson delta and lower NormMSE indicate better predictions.

Aggregation. For each metric, condition-level scores are averaged within each split and then equally across the five splits. Complete metric definitions and aggregation rules are provided in Appendix C.3.

4.2 Local embedding structure supports mean-response estimation

We examine whether local relationships in GenePT embedding space provide useful information for estimating mean responses to unseen perturbations. Across five Replogle K562 splits, we compare direct ridge regression on GenePT embeddings (Chen & Zou, 2024) with weighted aggregation of measured response deltas from the 16 nearest training perturbations. Inverse-rank and cosine-softmax weighting achieve Top100 Pearson delta values of 0.666 and 0.672, respectively, compared with 0.657 for ridge regression. The corresponding NormMSE values are 0.554 and 0.556, compared with 0.577 for ridge regression (Figure 2). Permuting gene–embedding assignments reduces the Pearson delta of inverse-rank aggregation to 0.418, supporting the predictive relevance of the original neighborhood relationships.

We further examine the association between embedding–response geometry and Scouter prediction accuracy across 215 held-out perturbations in K562 split 1. The Spearman association with Top100 Pearson delta is 0.679 for embedding–response neighborhood overlap among the 20 nearest neighbors, compared with 0.354 for global ranking agreement (Appendix D.1). This stronger local association is consistent with the value of neighborhood relationships for response estimation. Together, these findings motivate using local perturbation relationships to guide response-statistic aggregation when constructing query-specific priors.

4.3 Mean-response prediction across four datasets

We evaluate the mean-response accuracy of the complete PerturbRIFT framework across K562, RPE1, HepG2, and Jurkat (Table 1). Among the methods shown, PerturbRIFT achieves the highest Top100 Pearson delta and lowest NormMSE on each dataset. Top100 Pearson delta reaches 0.756, 0.788, 0.506, and 0.495, respectively, compared with 0.679, 0.749, 0.458, and 0.429 for RidgeSC, the strongest comparator on this metric.

The correlation gains persist when responses are centered on the training-perturbation mean, supporting improved prediction of perturbation-specific responses beyond the shared training-response

Table 1: **Mean-response prediction across four datasets.** Condition-level scores are averaged within each split and then equally across five splits. The first three metrics use query-specific Top100 DEG panels. PDS measures L1-based perturbation discrimination over all modeled genes, excluding the query target when present. ρ_Δ and $\hat{\rho}_\Delta$ center responses on the control and training-perturbation means, respectively; NMSE denotes NormMSE. Bold and underlining indicate the best and second-best distinct values, respectively, among the methods shown. NA denotes undefined correlation. PerturbRIFT is evaluated using sample means of cell populations generated with residual flow and Cube sampling.

Method	K562				RPE1			
	$\rho_\Delta\uparrow$	$\hat{\rho}_\Delta\uparrow$	NMSE↓	PDS↑	$\rho_\Delta\uparrow$	$\hat{\rho}_\Delta\uparrow$	NMSE↓	PDS↑
No change	NA	-0.279	1.000	0.501	NA	-0.317	1.000	0.500
Train mean	0.521	NA	0.783	0.501	0.726	NA	0.529	0.501
RidgeSC	<u>0.679</u>	<u>0.568</u>	<u>0.551</u>	0.710	<u>0.749</u>	<u>0.346</u>	<u>0.497</u>	0.638
SLIM	0.659	0.526	0.583	0.680	0.738	0.308	0.505	0.622
GEARS	0.510	0.234	0.732	0.624	0.699	0.156	0.598	0.580
TxPert-public	0.603	0.469	0.705	0.684	0.667	0.142	0.593	0.590
Scouter	0.595	0.452	0.649	0.710	0.674	0.249	0.606	0.658
CellFlow	0.545	0.257	0.731	0.612	0.702	0.122	0.604	0.564
scGPT	0.467	0.224	0.760	0.600	0.693	0.059	0.558	0.547
PerturbDiff	0.573	0.454	0.745	<u>0.745</u>	0.688	0.287	0.608	<u>0.680</u>
STATE	0.593	0.348	0.646	0.663	0.733	0.088	0.526	0.603
PerturbRIFT	**0.756**	**0.666**	**0.442**	**0.802**	**0.788**	**0.475**	**0.423**	**0.728**
Method	HepG2				Jurkat			
	$\rho_\Delta\uparrow$	$\hat{\rho}_\Delta\uparrow$	NMSE↓	PDS↑	$\rho_\Delta\uparrow$	$\hat{\rho}_\Delta\uparrow$	NMSE↓	PDS↑
No change	NA	-0.204	1.000	0.500	NA	-0.164	1.000	0.501
Train mean	0.409	NA	0.850	0.501	0.343	NA	0.907	0.501
RidgeSC	<u>0.458</u>	<u>0.344</u>	<u>0.759</u>	0.621	<u>0.429</u>	<u>0.342</u>	<u>0.797</u>	0.581
SLIM	0.439	0.314	0.772	0.607	0.412	0.315	0.810	0.571
GEARS	0.347	0.110	0.865	0.580	0.244	0.025	1.022	0.535
TxPert-public	0.391	0.212	0.946	0.579	0.419	0.246	0.826	0.578
Scouter	0.383	0.254	0.823	0.640	0.371	0.258	0.844	0.585
CellFlow	0.345	0.122	0.830	0.563	0.308	0.110	0.892	0.536
scGPT	0.384	0.151	0.839	0.559	0.286	0.095	0.940	0.530
PerturbDiff	0.409	0.265	0.849	<u>0.653</u>	0.363	0.266	0.871	<u>0.600</u>
STATE	0.414	0.153	0.791	0.604	0.371	0.197	0.839	0.566
PerturbRIFT	**0.506**	**0.377**	**0.714**	**0.686**	**0.495**	**0.414**	**0.738**	**0.644**

pattern. PerturbRIFT also achieves the highest PDS among the compared methods across all four datasets, indicating better discrimination between the matching observed response and other held-out perturbations. Complete results for all eight mean-response metrics are provided in Appendix D.2.

4.4 Single-cell population prediction across four datasets

We next evaluate the distributional fidelity of the predicted single-cell populations (Table 2). Among the methods shown, PerturbRIFT achieves the lowest expression-space MMD, gene-wise Wasserstein distance, and zero-rate error on each of the four datasets. PerturbRIFT also achieves the lowest PCA64MMD on all four datasets, indicating improved population-level agreement in a shared training-fitted representation.

We additionally assess agreement in perturbation-induced gene-correlation changes on the shared200 gene panel. PerturbRIFT ranks first among the compared methods on all four datasets on the shared200 panel. On RPE1, correlation-change agreement reaches 0.233, compared with 0.171 for SLIM, the strongest comparator on this metric. On Jurkat, it reaches 0.0684, compared with 0.0623 for STATE. Together, these results support the framework's ability to model multiple aspects of single-cell responses. Complete results for all eight distributional metrics, including PCA-space agreement and shared500 correlation changes, are provided in Appendix D.2.

Table 2: **Single-cell distributional fidelity across four datasets.** Expression-space MMD values are multiplied by 10^3 for display. W1 denotes GeneW1, and ZeroMAE denotes ZeroRateMAE. $r_{\Delta C}$ measures agreement between predicted and observed control-relative gene-correlation changes on a shared panel of 200 training-selected genes. Condition-level scores are averaged within each split and then equally across five splits. Bold and underlining indicate the best and second-best distinct values, respectively, among the methods shown. Missingness conventions follow Table 1. Control cells are sampled from training controls; GEARS uses paired-control cell inference.

Method	K562				RPE1			
	MMD↓	W1↓	ZeroMAE↓	$r_{\Delta C}$↑	MMD↓	W1↓	ZeroMAE↓	$r_{\Delta C}$↑
Control cells	7.880	0.0760	0.0593	0.0001	17.839	0.1001	0.0791	0.0006
SLIM	<u>5.117</u>	<u>0.0737</u>	<u>0.0573</u>	0.0897	<u>9.557</u>	<u>0.0870</u>	<u>0.0703</u>	<u>0.1712</u>
GEARS	6.894	0.0900	0.6104	0.0001	12.708	0.1235	0.6450	0.0006
Scouter	71.666	0.3080	0.6104	0.0211	75.107	0.2891	0.6450	0.0703
CellFlow	50.516	0.2510	0.6104	0.0974	46.229	0.2314	0.6450	0.1376
scGPT	73.044	0.3225	0.6104	0.0217	77.031	0.3087	0.6450	0.0235
PerturbDiff	9.018	0.1642	0.1083	0.0704	13.936	0.1319	0.1020	0.1277
STATE	5.855	0.1093	0.1291	<u>0.1157</u>	10.381	0.1113	0.1248	0.1664
PerturbRIFT	**4.627**	**0.0670**	**0.0504**	**0.1179**	**8.855**	**0.0810**	**0.0668**	**0.2330**
Method	HepG2				Jurkat			
	MMD↓	W1↓	ZeroMAE↓	$r_{\Delta C}$↑	MMD↓	W1↓	ZeroMAE↓	$r_{\Delta C}$↑
Control cells	10.443	0.0875	0.0664	-0.0002	5.634	0.0664	0.0557	-0.0012
SLIM	<u>7.726</u>	<u>0.0853</u>	<u>0.0648</u>	0.0896	<u>4.256</u>	<u>0.0652</u>	<u>0.0546</u>	0.0428
GEARS	8.807	0.0974	0.5920	-0.0002	6.026	0.0815	0.6374	-0.0012
Scouter	79.406	0.3089	0.5920	0.0425	65.134	0.2936	0.6374	0.0103
CellFlow	47.151	0.2398	0.5920	0.0854	42.475	0.2312	0.6374	0.0500
scGPT	81.759	0.3296	0.5920	0.0127	66.107	0.3038	0.6374	0.0088
PerturbDiff	11.359	0.1482	0.1116	0.0716	7.058	0.1305	0.1036	0.0310
STATE	8.258	0.1175	0.1261	<u>0.0953</u>	4.726	0.0986	0.1165	<u>0.0623</u>
PerturbRIFT	**7.477**	**0.0814**	**0.0618**	**0.1026**	**4.228**	**0.0630**	**0.0523**	**0.0684**

4.5 Ablation study

4.5.1 Biological relationships improve prior-mean estimation

We analyze the query-specific mean in Equation (3) using six nested configurations on K562, spanning control and random-anchor references, knowledge-guided neighborhood selection, learned weighting, and additional representation inputs. Each learned configuration is trained independently and evaluated using its analytic mean before source sampling or residual generation (Figure 5 in Appendix D.3).

Knowledge-guided neighborhood selection yields the largest gain. Replacing uniform averaging over 16 randomly selected eligible training perturbations with uniform averaging over 16 KG-ranked perturbations increases Top100 Pearson delta from 0.461 to 0.727 and reduces NormMSE from 0.809 to 0.495. Learning query-specific weights further improves these metrics to 0.757 and 0.446, respectively. Both changes improve Pearson delta and NormMSE in every split, on both the full modeled gene panel and Top100 non-dropout DEGs.

Additional representation inputs provide smaller, configuration-dependent gains. Relation features alone do not consistently improve Pearson delta, whereas incorporating raw query and anchor embeddings yields a Top100 Pearson delta of 0.760 and NormMSE of 0.438. These results identify biologically guided neighborhood selection and learned response weighting as the main contributors to prior-mean accuracy. Detailed configurations, estimates, and evaluation procedures are provided in Appendix D.3.

Table 3: **Effects of residual generation and sampling on K562 distributional fidelity.** Values are reported on their original scale and are equal-weight means of five split means, using seed 24 and conditions with at least 50 observed cells. Bold and underlining indicate the best and second-best distinct values, respectively, among the configurations shown, determined before rounding. P-MMD denotes PCA64MMD, SWD denotes SWD64, W1 denotes GeneW1, Zero denotes ZeroRateMAE, and Std denotes StdMAE. The final two columns measure agreement in control-relative gene-correlation changes on shared panels of 200 and 500 training-selected genes. Lower values are better for the first six metrics; higher values are better for both correlation metrics. R4 uses the R3 model with Cube sampling.

Configuration	MMD↓	P-MMD↓	SWD↓	W1↓	Zero↓	Std↓	$r^{200}_{\Delta C}$↑	$r^{500}_{\Delta C}$↑
$R0'$ Control shift	0.0052	0.0252	0.0819	0.0776	0.3869	0.0477	-0.0028	-0.0015
R0 Raw prior	0.0046	0.0734	0.0808	<u>0.0671</u>	<u>0.0511</u>	**0.0411**	-0.0001	0.0030
R1 No-flow roundtrip	0.0049	0.0678	0.0805	0.0699	0.0565	0.0483	0.0047	0.0068
R2 FM only	0.0046	0.0224	0.0812	0.0687	0.0538	0.0468	0.1113	0.0831
R3 FM + MMD	<u>0.0045</u>	<u>0.0213</u>	<u>0.0803</u>	0.0673	0.0525	0.0447	**0.1216**	**0.0896**
R4 Cube sampling	**0.0041**	**0.0212**	**0.0784**	**0.0646**	**0.0485**	<u>0.0435</u>	<u>0.1210</u>	<u>0.0890</u>

4.5.2 Contributions of residual generation and sampling

We evaluate the effects of residual generation and output sampling with the response-prior estimator and autoencoder held fixed (Table 3). The configurations include prior samples (R0), an autoencoder round trip without flow (R1), FM alone (R2), FM with latent MMD (R3), and R3 evaluated with Cube sampling (R4). A control-cell mean-shift baseline ($R0'$) provides an additional reference.

Relative to the no-flow round trip, FM reduces PCA64MMD from 0.0678 to 0.0224 and increases shared200 correlation-change agreement from 0.0047 to 0.1113. Adding latent MMD further reduces PCA64MMD to 0.0213 and increases shared200 correlation-change agreement to 0.1216. FM provides the main gains in structural recovery; adding MMD further improves average correlation-change agreement on both gene panels and lowers all six reported distance and error metrics.

Cube sampling further lowers all six distance and error metrics relative to R3, with small decreases in correlation-change agreement. The raw prior retains the lowest standard-deviation error, indicating the role of residual flow in more refining cross-gene dependence beyond the factorized prior marginal distribution. The control-shift reference has a PCA64MMD of 0.0252, compared with 0.0213 for R3, but near-zero correlation-change agreement. Complete configurations and evaluation scope are provided in Appendix D.3.

Figure 6 in Appendix D.3 illustrates predicted correlation-change patterns for RPL4, U2AF1, and MED10. In these examples, R2–R4 show closer agreement with the observed signed correlation changes than the prior and control-shift references. Table 3 provides the aggregate comparison across all eligible conditions.

5 Conclusion

We introduced PerturbRIFT to connect response-prior estimation with conditional single-cell generation. Across four within-dataset benchmarks, PerturbRIFT improves mean-response accuracy and multiple measures of distributional fidelity relative to the evaluated baselines. K562 ablations show that biological neighborhood selection accounts for the largest prior-mean gains, while residual flow improves recovery of perturbation-induced gene-correlation changes.

These findings highlight the value of biological knowledge graphs and gene embeddings for transferring measured response information from related experiments to unseen perturbations. They also suggest that constructing an informative prior distribution that closely matches the empirical measurements offers an additional avenue for improving prediction, alongside further refinement of the training method. Limitations and directions for future work are discussed in Appendix (Appendix E).

Reproducibility Statement

We provide detailed descriptions of the model architecture, retrieval procedure, training objectives, and evaluation protocols in the main text. Additional implementation details, hyperparameter settings, dataset preprocessing and splitting procedures, and metric definitions are provided in the appendices.

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. doi: 10.1038/s41592-025-02772-6. URL https://doi.org/10.1038/s41592-025-02772-6.

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. doi: 10.1038/s41592-023-01969-x. URL https://doi.org/10.1038/s41592-023-01969-x.

Yiqun Chen and James Zou. GenePert: Leveraging GenePT Embeddings for Gene Perturbation Prediction. *bioRxiv*, 2024. doi: 10.1101/2024.10.27.620513. URL https://doi.org/10.1101/2024.10.27.620513. Preprint.

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. doi: 10.1038/s41592-024-02201-0. URL https://doi.org/10.1038/s41592-024-02201-0.

Paul Datlinger, André F Rendeiro, Christian Schmidl, Thomas Krausgruber, Peter Traxler, Johanna Klughammer, Linda C Schuster, Amelie Kuchler, Donat Alpar, and Christoph Bock. Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods*, 14(3):297–301, 2017. doi: 10.1038/nmeth.4177. URL https://doi.org/10.1038/nmeth.4177.

J.-C. Deville and Y. Tillé. Efficient balanced sampling: The cube method. *Biometrika*, 91(4):893–912, 2004. doi: 10.1093/biomet/91.4.893. URL https://doi.org/10.1093/biomet/91.4.893.

Andrea Giuseppe Di Francesco, Andrea Rubbi, and Pietro Liò. Retrieval-Augmented Generation for Predicting Cellular Responses to Gene Perturbation. *arXiv preprint arXiv:2603.07233*, 2026. URL https://arxiv.org/abs/2603.07233.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M. Norman, Eric S. Lander, Jonathan S. Weissman, Nir Friedman, and Aviv Regev. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, 167(7):1853–1866.e17, 2016. doi: 10.1016/j.cell.2016.11.038. URL https://doi.org/10.1016/j.cell.2016.11.038.

Mingze Dong, Bao Wang, Jessica Wei, Antonio H. de O. Fonseca, Curtis J. Perry, Alexander Frey, Feriel Ouerghi, Ellen F. Foxman, Jeffrey J. Ishizuka, Rahul M. Dhodapkar, and David van Dijk. Causal identification of single-cell experimental perturbation effects with CINEMA-OT. *Nature Methods*, 20(11):1769–1779, 2023. doi: 10.1038/s41592-023-02040-5. URL https://doi.org/10.1038/s41592-023-02040-5.

Greg Finak, Andrew McDavid, Masanao Yajima, Jingyuan Deng, Vivian Gersuk, Alex K. Shalek, Chloe K. Slichter, Hannah W. Miller, M. Juliana McElrath, Martin Prlic, Peter S. Linsley, and Raphael Gottardo. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16:278, 2015. doi: 10.1186/s13059-015-0844-5.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 21(8):1481–1491, 2024. doi: 10.1038/s41592-024-02305-7. URL https://doi.org/10.1038/s41592-024-02305-7.

Siyu He, Yuefei Zhu, Daniel Naveed Tavakol, Haotian Ye, Yeh-Hsing Lao, Zixian Zhu, Cong Xu, Shradha Chauhan, Guy Garty, Raju Tomer, Gordana Vunjak-Novakovic, James Zou, Elham Azizi, and Kam W. Leong. Squidiff: predicting cellular development and responses to perturbations using a diffusion model. *Nature Methods*, 23(1):65–77, 2026. doi: 10.1038/s41592-025-02877-y. URL https://doi.org/10.1038/s41592-025-02877-y.

Dewei Hu, Marc Pielies Avellí, Lars Juhl Jensen, and Simon Rasmussen. SLIM: A small linear model with STRING embeddings for single-cell genetic perturbation prediction. *bioRxiv*, 2026. doi: 10.64898/2026.08.07.743481. URL https://doi.org/10.64898/2026.08.07.743481. Preprint.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, Nadezhda Azbukina, Fátima Sanchís-Calleja, Theo Uscidda, Artur Szalata, Manuel Gander, Aviv Regev, Barbara Treutlein, J. Gray Camp, and Fabian J. Theis. CellFlow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, 2025. doi: 10.1101/2025.04.11.648220. URL https://doi.org/10.1101/2025.04.11.648220. Preprint.

Quanquan Li, Yihe Chi, Liuyang Song, Hongbo Zhang, Jingyu Li, Xidong Xi, Conghua Wei, Yijie Sun, Yu Chen, Xin Liu, Qi Hu, Jing Ke, and Guitao Cao. Control-Anchored Residual Flow Matching Conditioned on Gene Geometry for Virtual Cell Perturbation Modeling. *arXiv preprint arXiv:2608.06824*, 2026. URL https://arxiv.org/abs/2608.06824.

Russell Littman, Jacob Levine, Sepideh Maleki, Yongju Lee, Vladimir Ermakov, Lin Qiu, Alexander Wu, Kexin Huang, Romain Lopez, Gabriele Scalia, Tommaso Biancalani, David Richmond, Aviv Regev, and Jan-Christian Hütter. Gene-embedding-based prediction and functional evaluation of perturbation expression responses with PRESAGE. *bioRxiv*, 2025. doi: 10.1101/2025.06.03.657653. URL https://doi.org/10.1101/2025.06.03.657653. Preprint.

Kaspar Märtens, Marc Boubnovski Martell, Cesar A. Prada-Medina, and Rory Donovan-Maiye. LangPert: LLM-Driven Contextual Synthesis for Unseen Perturbation Prediction. In *ICLR 2025 Workshop on Machine Learning for Genomics Explorations (MLGenX)*, 2025. URL https://openreview.net/forum?id=Tmx4o3Jg55.

Ajay Nadig, Joseph M. Replogle, Angela N. Pogson, Mukundh Murthy, Steven A. McCarroll, Jonathan S. Weissman, Elise B. Robinson, and Luke J. O'Connor. Transcriptome-wide analysis of differential expression in perturbation atlases. *Nature Genetics*, 57(5):1228–1237, 2025. doi: 10.1038/s41588-025-02169-3. URL https://doi.org/10.1038/s41588-025-02169-3.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, 2024. doi: 10.1038/s41587-023-01905-6. URL https://doi.org/10.1038/s41587-023-01905-6.

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, Reshma Mehta, Benjamin Hsu, Jeremy Sullivan, Chiara Ricci-Tam, Nianzhen Li, Julia Kazaks, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Hani Goodarzi, and Dave P. Burke. Virtual Cell Challenge: Toward a Turing test for the virtual cell. *Cell*, 188(13):3370–3374, 2025. doi: 10.1016/j.cell.2025.06.008. URL https://doi.org/10.1016/j.cell.2025.06.008.

Philipp S. L. Schäfer, Kendall A. Reid, Zach Boldyga, Ekin D. Aksu, Hugo Hakem, and Julio Saez-Rodriguez. Towards Principled Evaluation of Single-Cell Perturbation Prediction Models. *bioRxiv*, 2026. doi: 10.64898/2026.07.23.740433. URL https://doi.org/10.64898/2026.07.23.740433. Preprint.

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, and Patrick T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023. doi: 10.1038/s41586-023-06139-9. URL https://doi.org/10.1038/s41586-023-06139-9.

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*, 44:1050–1059, 2026. doi: 10.1038/s41587-025-02777-8.

Gefei Wang, Tianyu Liu, Jia Zhao, Youshu Cheng, and Hongyu Zhao. Modeling and predicting single-cell multi-gene perturbation responses with scLAMBDA. *bioRxiv*, 2024. doi: 10.1101/2024.12.04.626878. URL https://doi.org/10.1101/2024.12.04.626878. Preprint.

Frederik Wenkel, Wilson Tu, Cassandra Masschelein, Hamed Shirzad, Liam Hodgson, Ihab Bendidi, Cian Eastwood, Shawn T. Whitfield, Craig Russell, Yassir El Mesbahi, Jiarui Ding, Marta M. Fay, Berton Earnshaw, Emmanuel Noutahi, and Alisandra K. Denton. TxPert: using multiple knowledge graphs for prediction of transcriptomic perturbation effects. *Nature Biotechnology*, 2026. doi: 10.1038/s41587-026-03113-4. URL https://doi.org/10.1038/s41587-026-03113-4. Advance online publication.

Zichao Yan, Yan Wu, Mica Xu Ji, Chaitra Agrahar, Esther Wershof, Marcel Nassar, Mehrshad Sadria, Ridvan Eksi, Vladimir Trifonov, Ignacio Ibarra, Telmo Felgueira, Błażej Osiński, and Rory Stark. PRiMeFlow: Capturing Complex Expression Heterogeneity in Perturbation Response Modelling. *arXiv preprint arXiv:2604.13986*, 2026. URL https://arxiv.org/abs/2604.13986.

Chenglei Yu, Chuanrui Wang, Bangyan Liao, and Tailin Wu. scDFM: Distributional Flow Matching Model for Robust Single-Cell Perturbation Prediction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QSGanMEcUV.

Xian Zhang, Hui Wu, and Haikun Liu. DELPHAI predicts heterogeneous perturbation responses with learned cell fitness and gene-space retrieval. *bioRxiv*, 2026. doi: 10.64898/2026.07.01.735965. URL https://doi.org/10.64898/2026.07.01.735965. Preprint.

Ouyang Zhu and Jun Li. Scouter predicts transcriptional responses to genetic perturbations with large language model embeddings. *Nature Computational Science*, 6(1):21–28, 2026. doi: 10.1038/s43588-025-00912-8. URL https://doi.org/10.1038/s43588-025-00912-8.

A THE USE OF LARGE LANGUAGE MODELS (LLMS)

In this paper, we employ LLMs as assist tools for text refinement. All core research ideas and scientific conclusions presented in this paper are our own original contributions.

B METHOD DETAILS

B.1 ANCHOR SELECTION AND RELATION FEATURES

We use biological relationship resources to select training anchors and construct query–anchor features for response-prior estimation. Table 4 summarizes the resources and their roles. STRING provides functional association scores, CollecTRI supplies transcriptional regulatory relations, and Hallmark, KEGG, and Gene Ontology Biological Process collections provide gene-set memberships for computing pathway overlap. Gene representations are fixed 1536-dimensional GenePTV1 embeddings.

Candidate selection. For query q, the eligible candidate pool contains only training perturbations and excludes q and interventions on the same target gene. We compute three knowledge-based scores for each query–candidate pair: the STRING association score, a binary indicator of a CollecTRI regulatory relation, and the Jaccard overlap between their gene-set memberships pooled across Hallmark, KEGG, and Gene Ontology Biological Process. Each score is standardized within the candidate pool. The sum of the standardized scores determines the ranking, and the top A candidates form the anchor set $\mathcal{A}(q)$.

Table 4: **External resources for anchor selection and relation features.** All network and gene-set resources correspond to *Homo sapiens*. MSigDB gene sets use gene-symbol identifiers.

Resource	Version / collection	Role	Download
STRING	v12.0	Functional associations and protein-to-gene mapping	Links; Mapping
CollecTRI	Human network	Transcription factor–target relations	OmniPath
Hallmark	MSigDB v2026.1.Hs	Hallmark gene-set memberships	GMT
KEGG	MSigDB v2026.1.Hs	Curated pathway memberships	GMT
GO Biological Process	MSigDB v2026.1.Hs	Biological-process memberships	GMT
GenePT	V1; 1536 dimensions	Fixed gene embeddings	Zenodo

Query–anchor features. For relation-feature construction, the fixed gene embeddings are projected using a whitened PCA fitted exclusively on training-perturbation embeddings. The same transformation is applied to query and anchor embeddings, yielding $\tilde{e}_q, \tilde{e}_a \in \mathbb{R}^{d_e}$. For each selected anchor a, we construct

$$r_{qa} = [\tilde{e}_a,\ \tilde{e}_q \odot \tilde{e}_a,\ |\tilde{e}_q - \tilde{e}_a|,\ s_{qa}]\,, \tag{13}$$

where $\odot$ denotes elementwise multiplication and $s_{qa} \in \mathbb{R}^7$ contains the auxiliary features listed in Table 5. These features combine embedding relationships with knowledge-based scores, ranking information, and training-response support. The response-support feature uses only eligible training-condition responses and does not require the unseen query response. We use same configuration for all datasets, with $A = 16$, $d_e = 16$, and r_{qa} has $3d_e + 7 = 55$ dimensions.

Table 5: **Auxiliary query–anchor features.** The seven entries of s_{qa} supplement the embedding-derived components of r_{qa}.

Feature	Description
Combined knowledge score	Standardized combined score used to rank eligible anchors.
STRING score	Standardized STRING association score between the query and anchor.
CollecTRI score	Standardized binary indicator of a CollecTRI regulatory relation.
Pathway overlap score	Standardized Jaccard overlap of gene-set memberships pooled across Hallmark, KEGG, and Gene Ontology Biological Process.
Anchor rank	The anchor's position in the candidate ranking.
Seed-anchor indicator	Whether the anchor belongs to the seed-anchor set, defined as the top 4 selected anchors by combined knowledge score.
Training-response support	Directional agreement between the anchor's control-relative response and the seed-anchor responses, relative to the eligible training-pool mean.

Response-statistic estimation. Gene-wise zero probabilities and conditional nonzero-expression variances are estimated from complete training-condition response blocks. For a condition–gene pair with fewer than the configured minimum number of nonzero observations, the conditional nonzero mean and variance fall back to gene-specific estimates pooled across supported training conditions.

B.2 ZINormal representation training

The encoder processes expression values jointly with their observed nonzero masks. The probabilistic decoder represents each expression value using a point mass at zero and a Gaussian component for

nonzero observations. The autoencoder is trained by minimizing the average negative log-likelihood:

$$\mathcal{L}_{\text{rec}} = -\frac{1}{MG}\sum_{i,g}\begin{cases}\log\hat{\pi}_{i,g}, & x_{i,g}=0,\\ \log(1-\hat{\pi}_{i,g})+\log\mathcal{N}(x_{i,g};\hat{\mu}^{+}_{i,g},(\hat{\sigma}^{+}_{i,g})^2), & x_{i,g}\neq 0.\end{cases} \tag{14}$$

Each reconstruction example consists of a cell set sampled from a control or training-perturbation condition. The prescribed mean in Equation (11) is the empirical gene-wise mean of that same sampled set. The encoder and decoder checkpoint with the lowest validation reconstruction NLL is retained, frozen, and subsequently used for latent flow learning and prediction. Validation uses the empirical mean of each input cell set in the decoder, as in training, and does not update model parameters.

During residual training, source masks use hard forward draws with a straight-through relaxed-Bernoulli construction and an annealed temperature. At inference, source masks follow the Bernoulli sampling rule in Equation (6).

B.3 Sinkhorn pairing and latent distribution alignment

Stochastic source–target pairing. Since observed source–target cell correspondences are unavailable, we construct training pairs using entropic optimal transport in expression space. For source cell i and target cell j, the transport cost is

$$C_{ij} = \frac{1}{G}\|x_i^{(0)} - x_j^{(1)}\|_2^2. \tag{15}$$

Log-domain Sinkhorn iterations produce a balanced entropic transport plan with uniform empirical marginals. For each source cell, we normalize the corresponding plan row and independently sample one target index. Sampling is with replacement across rows, so target cells may be repeated and the selected indices need not form a permutation. The resulting pairs define the latent interpolation in Equation (8).

Latent endpoint alignment. We integrate the current velocity field to obtain transported latents $\widehat{Z}_1 = \{\hat{z}_i\}_{i=1}^M$ and align their distribution with the target latents $Z_1 = \{z_i^{(1)}\}_{i=1}^M$. For bandwidth s, the latent RBF kernel is

$$k_s(z,z') = \exp\left(-\frac{\|z-z'\|_2^2}{2Ds^2}\right), \tag{16}$$

where D is the latent dimension. The MMD objective averages a biased V-statistic over the bandwidth set $\mathcal{S}$:

$$\mathcal{L}_{\text{MMD}} = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\left[\frac{1}{M^2}\sum_{i,j}k_s(\hat{z}_i,\hat{z}_j) + \frac{1}{M^2}\sum_{i,j}k_s(z_i^{(1)},z_j^{(1)}) - \frac{2}{M^2}\sum_{i,j}k_s(\hat{z}_i,z_j^{(1)})\right]. \tag{17}$$

Both within-population sums include diagonal terms. Gradients propagate through the transported endpoint, while the target latents are treated as stop-gradient.

B.4 Cube sampling

At prediction time, we generate decoder masks using a per-gene adaptation of the cube method for balanced sampling (Deville & Tillé, 2004), without modifying the trained model or its objective. For each gene g, cell i has inclusion probability $p_{i,g} = 1 - \hat{\pi}_{i,g}$, with the decoder's nonzero mean $\hat{\mu}^{+}_{i,g}$ serving as the auxiliary balancing variable. The sampler couples inclusion decisions across cells through randomized pairwise updates that preserve the weighted probability total $\sum_i p_{i,g}\hat{\mu}^{+}_{i,g}$ until a final randomized rounding step resolves any remaining fractional probability. This procedure is designed to reduce finite-population imbalance in the mask-weighted expression totals.

Given the resulting binary masks, expression values are sampled from the same untruncated Gaussian components used in independent decoding and then clamped at zero. Cube sampling therefore changes only the generation of decoder masks; the Gaussian sampling and nonnegative output transformation remain unchanged.

B.5 TRAINING AND INFERENCE DETAILS

Hyperparameters. We use the same hyperparameters across all four datasets. The modeled gene count, cell-set size, and latent dimension are $G = 5000$, $M = 256$, and $D = 128$, respectively. Both the encoder and decoder use hidden dimension 512. The velocity network contains three Transformer blocks with eight attention heads and uses 64 fixed sinusoidal features to represent time. AdaLN-Zero modulation and a zero-initialized output projection initialize the latent flow near the identity transport.

The anchor-weight temperature is $\tau = 1$. Sinkhorn pairing uses entropic regularization 0.05 and 50 iterations. Flow integration uses a 20-step fourth-order Runge–Kutta solver. For latent distribution alignment, the kernel bandwidths are $\mathcal{S} = \{0.5, 1, 2, 4\}$ and $\lambda_{\mathrm{MMD}} = 0.5$.

Autoencoder training stops when validation reconstruction NLL plateaus; response-prior estimation and residual flow training use $T_w = 20{,}000$ and $T_{\mathrm{res}} = 20{,}000$ optimization steps, respectively. The corresponding AdamW learning rates are 3×10^{-4}, 2×10^{-4}, and 2×10^{-4}. All stages use weight decay 10^{-4} and gradient-norm clipping at 1. During residual training, the relaxed-Bernoulli temperature decreases linearly from 1.0 to 0.2.

Training and inference procedures. Algorithm 1 summarizes representation training, response-prior estimation, and residual flow learning. Algorithm 2 generates a cell population for an unseen query.

In both algorithms, flow integration solves

$$\frac{dZ_t}{dt} = v_\theta(Z_t, t, c_q, m_0), \qquad Z_{t=0} = Z_0, \tag{18}$$

using fixed-step fourth-order Runge–Kutta integration. The response-informed context c_q and prior mean m_0 remain fixed throughout the trajectory. The RK4 routine uses Z_0 as the initial state; subsequent velocity evaluations depend on the evolving latent population Z_t, time, c_q, and m_0.

Notation and sampling operations. The parameter sets Θ_{rec}, Θ_w, and Θ_c denote the autoencoder, anchor-weighting, and moment-to-context parameters, respectively; θ denotes the velocity-network parameters. The collection $\mathcal{S}_{\mathrm{tr}}$ stores training-condition response statistics and the control mean. The metadata collection $\mathcal{R}$ contains fixed target embeddings, eligible anchor pools, and query–anchor features.

The tuple $P_q = (m_0, \pi_0, (\sigma_0^+)^2, \mu_0^+)$ contains the query-specific prior parameters. The function $\mathrm{BuildPrior}$ computes these parameters from the learned aggregation weights using Equations (2)–(5) and constructs the context c_q. Both this function and the expanded inference procedure apply the nonzero-probability and variance lower bounds specified in Section 3.2.

The function $\mathrm{SampleCellSet}$ selects a single condition and samples all M cells from that condition. The training sampler $\mathrm{SamplePrior}_{\mathrm{ST}}$ uses the annealed straight-through construction described in Appendix B.3. The function $\mathrm{SampleTargets}$ independently draws one target index from each normalized transport-plan row, with replacement across rows.

For source generation at inference, $\mathrm{SampleZINormal}$ uses independent Bernoulli masks and untruncated Gaussian values. For final decoder sampling, $\mathrm{SampleZINormalCube}$ uses Cube-balanced Bernoulli masks with the same Gaussian sampling procedure, as described in Appendix B.4. The operation $\mathrm{AdamWStep}(\Theta, \mathcal{L})$ updates only the specified parameters Θ, using the stage-specific settings given above. The operation $\mathrm{UpdateAESelection}$ retains the parameters with the lowest validation NLL and updates the significant-improvement reference and patience counter. Its state is $S = (\Theta_{\mathrm{best}}, L_{\mathrm{best}}, L_{\mathrm{ref}}, b)$, where b counts consecutive evaluations without a significant improvement.

Algorithm 1 Training PerturbRIFT

Input: Training populations $\mathcal{D}_{\mathrm{tr}} = \{X_q : q \in \mathcal{C}_{\mathrm{tr}}\}$; controls X_{ctrl}; validation populations $\mathcal{D}_{\mathrm{val}}$; embeddings $\{e_c\}$; knowledge-graph relations $\mathcal{K}$; cell-set size M.
Output: Trained parameters $(\Theta_{\mathrm{rec}}, \Theta_w, \Theta_c, \theta)$; training statistics $\mathcal{S}_{\mathrm{tr}}$ and anchor metadata $\mathcal{R}$.

$\mathcal{S}_{\mathrm{tr}} \leftarrow \mathrm{EstimateMoments}(\mathcal{D}_{\mathrm{tr}}, X_{\mathrm{ctrl}})$
$\mathcal{R} \leftarrow \mathrm{BuildEligibleAnchors}(\mathcal{C}_{\mathrm{tr}}, \{e_c\}, \mathcal{K}, \mathcal{S}_{\mathrm{tr}})$
$(\Theta_{\mathrm{rec}}, \Theta_w, \Theta_c, \theta) \leftarrow \mathrm{Initialize}()$
Phase 1: representation training
$\mathcal{V} \leftarrow \mathrm{FixedValidationCellSets}(\mathcal{D}_{\mathrm{val}}, M)$; $k \leftarrow 0$
$L_0 \leftarrow \mathrm{ValidationNLL}(\Theta_{\mathrm{rec}}, \mathcal{V})$; $S \leftarrow (\Theta_{\mathrm{rec}}, L_0, L_0, 0)$
while $k < 50{,}000$ or $S.b < 5$ **do**
  $k \leftarrow k + 1$; $X \leftarrow \mathrm{SampleCellSet}(\mathcal{D}_{\mathrm{tr}} \cup \{X_{\mathrm{ctrl}}\}, M)$
  $B \leftarrow \mathbf{1}[X \neq 0]$; $m \leftarrow M^{-1} \sum_i x_i$; $Z \leftarrow E_\phi(X, B)$
  $(\hat{\pi}, \hat{\mu}^+, \hat{\sigma}^+) \leftarrow \mathrm{Decode}(Z, m; \Theta_{\mathrm{rec}})$
  $\mathcal{L}_{\mathrm{rec}} \leftarrow \mathrm{ZINormalNLL}(X; \hat{\pi}, \hat{\mu}^+, \hat{\sigma}^+)$
  $\Theta_{\mathrm{rec}} \leftarrow \mathrm{AdamWStep}(\Theta_{\mathrm{rec}}, \mathcal{L}_{\mathrm{rec}})$
  if $k \bmod 10{,}000 = 0$ **then**
    $L \leftarrow \mathrm{ValidationNLL}(\Theta_{\mathrm{rec}}, \mathcal{V})$; $S \leftarrow \mathrm{UpdateAESelection}(S, \Theta_{\mathrm{rec}}, L)$
  end if
end while
$\Theta_{\mathrm{rec}} \leftarrow S.\Theta_{\mathrm{best}}$; $\mathrm{Freeze}(\Theta_{\mathrm{rec}})$
Phase 2: response-prior estimation
for $k = 1, \ldots, T_w$ **do**
  $q \leftarrow \mathrm{SampleQuery}(\mathcal{C}_{\mathrm{tr}})$
  $w_q \leftarrow \mathrm{AnchorWeights}(q, \mathcal{R}; \Theta_w)$
  $\delta_0 \leftarrow \sum_{a \in \mathcal{A}(q)} w_{qa} \delta_a$; $m_0 \leftarrow m_{\mathrm{ctrl}} + \delta_0$
  $\mathcal{L}_{\mathrm{prior}} \leftarrow \mathrm{PriorLoss}(m_0, \delta_0; m_q, \delta_q)$
  $\Theta_w \leftarrow \mathrm{AdamWStep}(\Theta_w, \mathcal{L}_{\mathrm{prior}})$
  if checkpoint step **then**
    $\mathrm{SaveWeights}(\Theta_w)$
  end if
end for
$\Theta_w \leftarrow \mathrm{SelectSavedWeights}(\mathcal{D}_{\mathrm{val}}, \mathcal{R}, \mathcal{S}_{\mathrm{tr}})$
$\mathrm{Freeze}(\Theta_w)$
Phase 3: residual flow learning
for $k = 1, \ldots, T_{\mathrm{res}}$ **do**
  $q \leftarrow \mathrm{SampleQuery}(\mathcal{C}_{\mathrm{tr}})$
  $(P_q, c_q) \leftarrow \mathrm{BuildPrior}(q, \mathcal{R}, \mathcal{S}_{\mathrm{tr}}; \Theta_w, \Theta_c)$
  $(m_0, \pi_0, (\sigma_0^+)^2, \mu_0^+) \leftarrow P_q$
  $X_0 \leftarrow \mathrm{SamplePrior}_{\mathrm{ST}}(P_q, M, k)$; $X_1 \leftarrow \mathrm{SampleCells}(X_q, M)$
  $C_{ij} \leftarrow G^{-1} \|x_i^{(0)} - x_j^{(1)}\|_2^2$; $\Pi \leftarrow \mathrm{Sinkhorn}(C)$
  $J \leftarrow \mathrm{SampleTargets}(\Pi, \mathrm{replacement} = \mathrm{True})$; $X_1 \leftarrow X_1[J, :]$
  $B_j \leftarrow \mathbf{1}[X_j \neq 0]$; $Z_j \leftarrow E_\phi(X_j, B_j)$, $j \in \{0, 1\}$
  $t \sim \mathcal{U}(0, 1)$; $Z_t \leftarrow (1 - t)Z_0 + tZ_1$
  $\mathcal{L}_{\mathrm{FM}} \leftarrow \|v_\theta(Z_t, t, c_q, m_0) - (Z_1 - Z_0)\|_2^2$
  $\widehat{Z}_1 \leftarrow \mathrm{RK4}(v_\theta, Z_0, 0, 1; c_q, m_0)$
  $\mathcal{L}_{\mathrm{MMD}} \leftarrow \mathrm{MMD}(\widehat{Z}_1, \mathrm{StopGradient}(Z_1))$
  $\mathcal{L}_{\mathrm{res}} \leftarrow \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{MMD}} \mathcal{L}_{\mathrm{MMD}}$
  $(\theta, \Theta_c) \leftarrow \mathrm{AdamWStep}((\theta, \Theta_c), \mathcal{L}_{\mathrm{res}})$
end for
return $(\Theta_{\mathrm{rec}}, \Theta_w, \Theta_c, \theta), \mathcal{S}_{\mathrm{tr}}, \mathcal{R}$

Algorithm 2 Inference for an unseen perturbation

Input: Unseen query q; trained parameters $(\Theta_{\mathrm{rec}}, \Theta_w, \Theta_c, \theta)$; training statistics $\mathcal{S}_{\mathrm{tr}}$ and anchor metadata $\mathcal{R}$; requested population size M.

Output: Predicted single-cell population $\widehat{X}_q$.

$\mathcal{A}(q), r_q \leftarrow \mathrm{Retrieve}(q, \mathcal{R})$
$w_q \leftarrow \mathrm{AnchorWeights}(q, \mathcal{R}; \Theta_w)$
$m_0 \leftarrow m_{\mathrm{ctrl}} + \sum_{a \in \mathcal{A}(q)} w_{qa} \delta_a$
$\pi_0 \leftarrow \sum_{a \in \mathcal{A}(q)} w_{qa} \pi_a; \quad (\sigma_0^+)^2 \leftarrow \sum_{a \in \mathcal{A}(q)} w_{qa} (\sigma_a^+)^2$
Apply the nonzero-probability and variance lower bounds.
$\mu_0^+ \leftarrow m_0/(1-\pi_0); \quad P_q \leftarrow (m_0, \pi_0, (\sigma_0^+)^2, \mu_0^+)$
$c_q \leftarrow \mathrm{BuildContext}(q, \mathcal{R}, w_q, P_q; \Theta_w, \Theta_c)$
$X_0 \leftarrow \mathrm{SampleZINormal}(\pi_0, \mu_0^+, \sigma_0^+, M)$
$B_0 \leftarrow \mathbf{1}[X_0 \neq 0]; \quad Z_0 \leftarrow E_\phi(X_0, B_0)$
$\widehat{Z}_1 \leftarrow \mathrm{RK4}(v_\theta, Z_0, 0, 1; c_q, m_0)$
$(\hat{\pi}, \hat{\mu}^+, \hat{\sigma}^+) \leftarrow \mathrm{Decode}(\widehat{Z}_1, m_0; \Theta_{\mathrm{rec}})$
$\widehat{X}_q \leftarrow \mathrm{SampleZINormalCube}(\hat{\pi}, \hat{\mu}^+, \hat{\sigma}^+, M)$
$\widehat{X}_q \leftarrow \max(\widehat{X}_q, 0)$
return $\widehat{X}_q$

C Evaluation Details

C.1 Datasets, perturbation splits, and preprocessing

We evaluate PerturbRIFT on four perturbation datasets comprising 1081 K562, 1521 RPE1, 1775 HepG2, and 2079 Jurkat perturbation conditions after resource-coverage filtering. Each dataset uses a fixed panel of 5000 expression genes, referred to as allHVG throughout the evaluation.

For K562 and RPE1, we use the GEARS-preprocessed files, retaining their expression values and 5000-gene panels without additional normalization, log transformation, or highly variable gene selection. HepG2 and Jurkat are constructed from public raw UMI-count H5AD files for GSE264667, distributed through the Arc Institute Replogle–Nadig data resource (Nadig et al., 2025). For these two datasets, we normalize library sizes to the source-dataset median, apply $\log(1+x)$, select 5000 highly variable genes using the Seurat procedure, and remove perturbation conditions with fewer than 30 cells.

We generate all splits using a shared two-stage GEARS-like condition-shuffling procedure. We first retain perturbation targets covered by GenePT_V1, using the existing gene-alias mapping. For each seed in $\{1, 2, 3, 4, 5\}$, we assign approximately 20% of these targets to the test set and 10% of the remaining targets to the validation set. We then retain only targets covered by both the GEARS gene2go and essential-gene resources, preserving their assigned groups without rebalancing. This yields approximately 72%/8%/20% training/validation/test proportions. Models are trained and checkpoints selected separately for each dataset and split.

C.2 Baseline implementations

All methods use the perturbation splits and output gene panels described in Appendix C.1. The four datasets each include 12 methods in the mean-response comparison and 9 in the single-cell comparison, counting the corresponding no-change baselines and PerturbRIFT. Below we describe the implementations used in these experiments, including their input representations, training settings, model selection, and prediction procedures.

No change. The mean-response baseline assigns the training-control mean to every query. For distribution evaluation, we sample 256 training-control cells per query and return their expression unchanged. These are the same source cells used for the paired-control GEARS distribution predictions.

Training mean. We compute a pseudobulk expression mean for each training perturbation from its available training cells, then average these vectors with equal weight across conditions. Every query receives this same mean vector. This baseline requires no model fitting and provides mean predictions only.

RidgeSC. We use the RidgeSC implementation distributed with SLIM to regress training-perturbation pseudobulk expression on fixed 100,000-dimensional STRING embeddings. The model uses the mean across training perturbations as its bias and a closed-form ridge solution with regularization coefficient 0.1. It predicts expression directly without a PCA response basis and clamps negative predicted means to zero. Hyperparameters are fixed without validation-based selection, and only mean outputs are evaluated.

SLIM. We use the official bilinear model with 100,000-dimensional STRING embeddings, a 10-component PCA response basis, training-mean bias, and closed-form ridge regularization of 0.1. The response basis and regression are fitted to training-perturbation pseudobulk means, with fixed hyperparameters. For single-cell prediction, the default rescale procedure samples 256 cells uniformly from the combined training-control and training-perturbation cell pool for each query. Each gene is scaled by the ratio of its predicted mean to its sampled mean, with numerical floors of 10^{-6}, and expression is clipped to $[0, 14.99]$. Both distribution metrics and mean-response metrics use the resulting rescaled cells.

GEARS. We train the official GEARS model using its Gene Ontology and co-expression graphs, with hidden size 64, one layer for each graph branch, and decoder hidden size 16. Training runs for 20 epochs with batch size 32, learning rate 10^{-3}, weight decay 5×10^{-4}, and direction-loss weight 0.1. The checkpoint is selected using the official validation MSE on differentially expressed genes. Mean-response evaluation uses the original predictions averaged over 300 sampled training-control cells per query. For distribution evaluation, the same trained model transforms 256 sampled control cells per query; these controls are shared with the cell-level no-change baseline. The distribution predictions do not replace the original mean predictions in the mean-response comparison.

TxPert-public. We train the public TxPert implementation using a STRING graph with the top 20 neighbors per node. The configuration combines an MLP control-expression model with a four-layer GATv2 perturbation model using hidden size 128, two attention heads, and dropout 0.2. The overall predictor uses hidden size 512, latent size 64, and an MSE objective. Training uses batch size 64 for up to 60 epochs, and the checkpoint with the lowest validation loss is selected. Inference conditions on the training-control mean and produces a deterministic expression vector; only mean-response metrics are evaluated.

Scouter. We use the official Scouter model with fixed 1536-dimensional GenePT_V1 embeddings and the supplied gene-symbol alias mapping. The default cell encoder has hidden sizes 2048 and 512 with a 64-dimensional output, and the decoder has a 2048-dimensional hidden layer. Training uses batch size 256, learning rate 10^{-3}, an epoch-wise learning-rate decay factor of 0.9, and the official loss with $\gamma = 0$ and $\lambda = 0.5$. We train for up to 40 epochs with validation-based early stopping and patience 5, retaining the selected model weights. The official prediction routine generates 256 outputs from sampled training-control cells for each query, and their mean is used for mean-response evaluation.

CellFlow. We use the official optimal-transport flow-matching model with deterministic conditioning on 2560-dimensional ESM2 protein embeddings. A centered 100-component PCA is fitted using training perturbation cells and training controls, and flow matching is performed in this PCA space. The model is trained for $100{,}000$ iterations with batch size 64, without using validation responses for model selection. Inference transports a fixed set of 256 training-control cells, shared across queries within each split and seed. Predicted PCA coordinates are mapped back to the 5000-gene expression space by the linear inverse PCA transform, without nonnegative clipping. Mean predictions are computed from the reconstructed cells.

scGPT. We fine-tune the official scGPT perturbation model, initializing its gene encoder, value encoder, and Transformer encoder from the human pretrained checkpoint. Training uses randomly sampled gene sequences of length at most 1536 while retaining the perturbation target, with an MSE expression-reconstruction objective. We use batch size 16, gradient accumulation over four batches, Adam with learning rate 10^{-4}, an epoch-wise decay factor of 0.9, and gradient-norm clipping at 1. Fine-tuning runs for up to 15 epochs, with early-stopping patience 10 and checkpoint selection

by validation Pearson correlation between predicted and observed mean expression. Gene symbols are mapped through declared aliases; unresolved genes use the padding token. At inference, all 5000 panel genes are processed together, with an additional zero-expression target token when the perturbation target is outside the panel. The model predicts responses for 256 sampled training-control cells per query, clamps negative outputs to zero, and averages the resulting cells for mean-response evaluation.

PerturbDiff. We use the official Cross-DiT backbone and Gaussian diffusion implementation, adapted to the fixed 5000-gene expression panel and 1536-dimensional target embeddings. Processed expression is divided by 10 for modeling and rescaled after generation. The backbone has hidden size 512, eight layers, and eight attention heads; diffusion uses 1000 steps with a linear noise schedule. Training uses four cell sets of size 32 per step for 20,000 steps, with AdamW learning rate 0.002, weight decay 0.01, and gradient-norm clipping at 1. The MSE and energy-distance loss terms each have weight 1. We use the final exponential-moving-average weights, with decay 0.99, without validation-based checkpoint selection. Prediction uses 100 deterministic DDIM steps spanning the full trained noise schedule, with $\eta = 0$ and guidance strength 1, to generate eight sets of 32 cells per query. The resulting 256 cells provide both distribution predictions and their corresponding mean responses.

STATE. We train the official STATE cell-set model from scratch using 1536-dimensional GenePT target embeddings; no pretrained response model is used. The model receives 2000 highly variable genes selected from training perturbation cells and controls, and its gene decoder predicts the full 5000-gene panel. We use cell sets of size 512, residual prediction, a softplus output, and an energy-distance distributional loss. Training runs for up to 40,000 steps with microbatch size 1, gradient accumulation over eight batches, learning rate 10^{-4}, weight decay 5×10^{-4}, and gradient clipping at 10. Validation is performed every 1000 steps, and the checkpoint with the lowest validation loss is selected. For each query, inference samples 256 training-control cells with replacement and produces a cell-set response, which is clipped to $[0, 14]$ following the official inference procedure. Mean predictions are computed from these output cells.

PerturbRIFT. The architecture, training hyperparameters, and inference procedure are described in Appendix B.5. All four datasets use the complete predictor with Cube sampling for the final decoded population, as described in Appendix B.4. Mean-response predictions are computed from the 256 generated cells after nonnegative clamping.

C.3 Metric definitions and aggregation

Let $\hat{m}_q$ and m_q denote the predicted and observed mean-expression vectors for perturbation q, m_{ctrl} the training-control mean, and $\bar{m}_{\mathrm{tr}}$ the equal-condition mean across training perturbations. Mean-response metrics use either all modeled genes or the query-specific Top100 panel, denoted by S. Distribution metrics compare $M = 256$ predicted and observed cells over $G = 5000$ genes.

Mean-response metrics. We report MSE, MAE, and Spearman correlation between predicted and observed expression means. Pearson delta measures agreement between control-relative responses, while centroid-centered Pearson delta measures agreement beyond the average training-perturbation response:

$$\rho_{\Delta}(q) = \mathrm{corr}_{g \in S}(\hat{m}_q - m_{\mathrm{ctrl}}, m_q - m_{\mathrm{ctrl}}), \tag{19}$$

$$\hat{\rho}_{\Delta}(q) = \mathrm{corr}_{g \in S}(\hat{m}_q - \bar{m}_{\mathrm{tr}}, m_q - \bar{m}_{\mathrm{tr}}). \tag{20}$$

NormMSE normalizes prediction error by the error of the no-change baseline:

$$\mathrm{NormMSE}(q) = \frac{\sum_{g \in S}(\hat{m}_{q,g} - m_{q,g})^2}{\sum_{g \in S}(m_{q,g} - m_{\mathrm{ctrl},g})^2}. \tag{21}$$

Values below 1 indicate improvement over predicting the control mean.

Perturbation discrimination score. For each predicted response, L1 PDS ranks the observed means of all Q test perturbations in the same split by L1 distance, excluding the query target gene when present. If r_q is the ascending rank of the matching observed response, using average ranks for ties, then

$$\mathrm{PDS}(q) = 1 - \frac{r_q - 1}{Q - 1}. \tag{22}$$

Higher scores indicate better discrimination.

Maximum mean discrepancy. We compare joint expression distributions using multiscale inverse-multiquadratic kernels:

$$k_s^{(d)}(u,v) = \frac{s^2}{s^2 + \|u-v\|_2^2/d}, \qquad s \in \mathcal{S} = \{0.5, 1, 2, 4\}. \tag{23}$$

For populations $U = \{u_i\}_{i=1}^M$ and $V = \{v_i\}_{i=1}^M$ in d dimensions, the estimator is

$$\mathcal{D}_d(U,V) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \left[\frac{\sum_{i \neq j} \left[k_s^{(d)}(u_i, u_j) + k_s^{(d)}(v_i, v_j) \right]}{M(M-1)} - \frac{2}{M^2} \sum_{i,j} k_s^{(d)}(u_i, v_j) \right]. \tag{24}$$

Expression-space MMD applies $\mathcal{D}_G$ to the predicted and observed populations. This is an unbiased estimate of squared MMD, distinct from the biased latent RBF-MMD training objective.

PCA64MMD. We fit a centered, non-whitened 64-dimensional PCA to 4096 condition-balanced training cells. With loading matrix V_{64} and training mean μ_{PCA}, the normalized projection is

$$T(x) = \frac{V_{64}^\top (x - \mu_{\mathrm{PCA}})}{b}, \tag{25}$$

where b^2 is the median nonzero squared distance between projected training cells divided by 64, estimated from a fixed subset of 512 PCA-fitting cells. PCA64MMD applies $\mathcal{D}_{64}$ to the projected predicted and observed populations. The projection and scale are shared across methods and test conditions within each split.

Sliced Wasserstein distance. SWD64 averages one-dimensional Wasserstein-1 distances along 64 fixed random unit directions obtained by normalizing standard Gaussian vectors. For each direction, the distance is the mean absolute difference between sorted predicted and observed projections. The same directions are used across methods.

Gene-wise Wasserstein distance. GeneW1 averages the Wasserstein-1 distance between predicted and observed marginal expression distributions across all genes. With equal population sizes, each gene's distance is the mean absolute difference between its sorted predicted and observed expression values.

Zero-rate and standard-deviation metrics. ZeroRateMAE is the mean absolute difference between predicted and observed zero fractions across genes. StdMAE is the corresponding error in gene-wise expression standard deviations, computed with denominator M.

Correlation-change recovery. We select shared panels of $K \in \{200, 500\}$ genes ranked by their average absolute control-relative response across training perturbations. On each panel, let $\widehat{C}_q$, C_q, and C_{ctrl} denote the gene-correlation matrices of predicted, observed, and training-control cells. We quantify recovery of perturbation-induced correlation changes by

$$r_{\Delta C}^K(q) = \mathrm{corr}\left(\mathrm{vec}_\triangle(\widehat{C}_q - C_{\mathrm{ctrl}}), \mathrm{vec}_\triangle(C_q - C_{\mathrm{ctrl}}) \right), \tag{26}$$

where $\mathrm{vec}_\triangle$ extracts the strict upper triangle. Panels are fixed within each split and shared across conditions and methods, without excluding query targets. Gene-pair correlations with a numerically zero variance denominator are set to zero when constructing the matrices. The main-table notation $r_{\Delta C}$ denotes $r_{\Delta C}^{200}$.

Aggregation. Condition-level scores are averaged within each split and then equally across the five splits, with results reported separately for each dataset. Higher correlation and PDS values indicate better performance, whereas lower error and distance values are preferred.

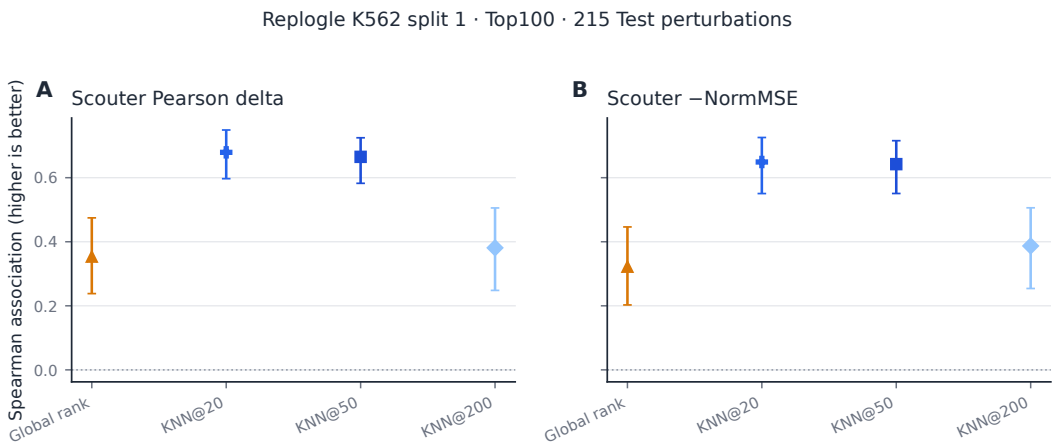


Figure 3: **Local embedding–response agreement and prediction accuracy.** Across 215 held-out perturbations in K562 split 1, points show Spearman associations between embedding–response geometry agreement and Scouter performance, measured by Top100 Pearson delta (A) or negative NormMSE (B). Error bars indicate 95% bootstrap confidence intervals from query resampling. Global rank compares the full training-candidate similarity rankings in embedding and response space. KNN@k reports chance-adjusted overlap between the top-k nearest neighbors in the two spaces. These neighborhood sizes are used only for diagnostic analysis and are independent of the 16 anchors used in Figure 2. Held-out responses are used only for diagnostic scoring.

D Additional Results

D.1 Embedding function: prediction and locality diagnostics

We further examine how GenePT embeddings relate to response prediction performance. Using the five K562 splits, we compare direct ridge regression on fixed, L2-normalized GenePT embeddings with weighted aggregation of measured response deltas from the 16 nearest training perturbations ranked by embedding cosine similarity. Retrieval uses uniform, inverse-rank, or cosine-softmax weighting, with ridge regularization and softmax temperature selected on the validation set using Top100 Pearson delta. To assess the contribution of meaningful embedding structure, we additionally repeat inverse-rank aggregation after permuting gene–embedding assignments across 20 random seeds. Performance is evaluated using Pearson delta and NormMSE on each query's predefined Top100 non-dropout DEG panel, averaging first across perturbations within each split and then across the five splits (Figure 2).

We next investigate whether local or global agreement between embedding geometry and response geometry better explains predictive performance. For each of the 215 held-out perturbations in K562 split 1, we rank training perturbations by cosine similarity in embedding space and by cosine similarity of control-relative mean responses, evaluated on the query's Top100 panel. Global agreement is measured by Spearman rank correlation between the two full rankings. Local agreement is measured by chance-adjusted overlap among the nearest 20, 50, or 200 neighbors.

Local agreement at $k = 20$ and $k = 50$ is more strongly associated with higher Pearson delta and lower NormMSE than either global rank agreement or overlap at $k = 200$ (Figures 3 and 4). Uncertainty is estimated using 1000 query-bootstrap resamples. These results support the relevance of local embedding neighborhoods for identifying training perturbations whose measured responses are informative for unseen targets.

D.2 Extended benchmark tables

Tables 6–13 present the complete results for eight mean-response and eight single-cell distribution metrics on K562, RPE1, HepG2, and Jurkat. Values are averaged across five perturbation splits, with bold and underline indicating the best and second-best distinct scores for each metric, respectively;

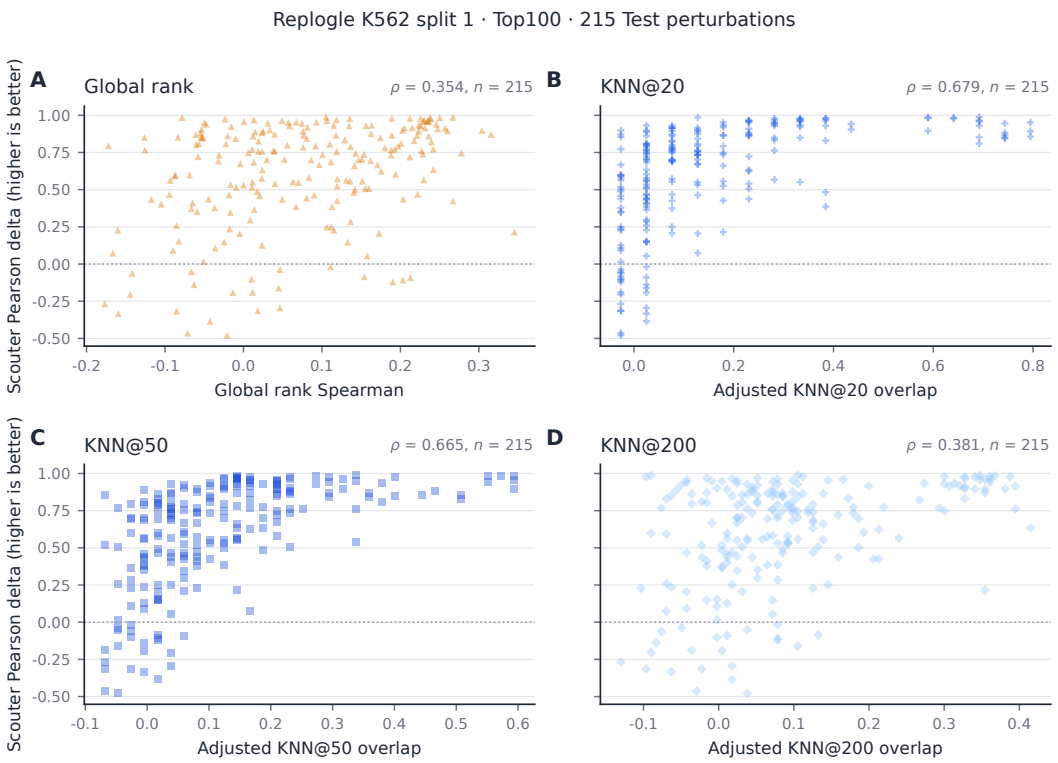


Figure 4: **Query-level embedding geometry and Scouter accuracy.** Each point corresponds to one of 215 held-out perturbations in Replogle K562 split 1. The vertical axis shows Top100 Pearson delta. The horizontal axes report global embedding–response rank agreement (A) and chance-adjusted nearest-neighbor overlap at $k = 20$, 50, and 200 (B–D). Annotations report Spearman correlations across queries. Prediction accuracy is more strongly associated with local neighbor overlap at $k = 20$ and $k = 50$ than with global rank agreement or broader overlap at $k = 200$.

rankings are determined before rounding. Metric definitions and aggregation procedures are provided in Appendix C.3.

Table 6: **K562: all eight selected mean-response metrics.** Values are equal averages of five split means; bold and underline denote the best and second distinct values before rounding. ρ_Δ, $\hat{\rho}_\Delta$, and NMSE follow Table 1; PDS is L1 perturbation discrimination over all genes excluding the query target when present. NA denotes undefined correlation. PerturbRIFT uses published residual Cube cell means.

Method	Top100				allHVG			All genes
	$\rho_\Delta \uparrow$	$\hat{\rho}_\Delta \uparrow$	NMSE$\downarrow$	MAE$\downarrow$	$\rho_\Delta \uparrow$	$\hat{\rho}_\Delta \uparrow$	NMSE$\downarrow$	PDS$\uparrow$
No change	NA	-0.279	1.000	0.282	NA	0.077	1.000	0.501
Train mean	0.521	NA	0.783	0.240	0.362	NA	0.941	0.501
RidgeSC	<u>0.679</u>	<u>0.568</u>	<u>0.551</u>	<u>0.178</u>	0.481	<u>0.378</u>	0.922	0.710
SLIM	0.659	0.526	0.583	0.184	<u>0.482</u>	0.372	<u>0.878</u>	0.680
GEARS	0.510	0.234	0.732	0.215	0.350	0.213	1.068	0.624
TxPert-public	0.603	0.469	0.705	0.197	0.429	0.361	1.091	0.684
Scouter	0.595	0.452	0.649	0.185	0.399	0.309	1.136	0.710
CellFlow	0.545	0.257	0.731	0.217	0.386	0.258	0.941	0.612
scGPT	0.467	0.224	0.760	0.218	0.335	0.171	1.091	0.600
PerturbDiff	0.573	0.454	0.745	0.188	0.399	0.355	1.307	<u>0.745</u>
STATE	0.593	0.348	0.646	0.194	0.416	0.277	0.968	0.663
PerturbRIFT	**0.756**	**0.666**	**0.442**	**0.147**	**0.548**	**0.457**	**0.863**	**0.802**

Table 7: **K562: all eight selected single-cell metrics.** Five split means are weighted equally; bold and underline denote best and second distinct values before rounding. P-MMD is PCA64MMD, SWD is SWD64, W1 is GeneW1, Zero is ZeroRateMAE, and Std is StdMAE. All values use their original scale. MMD, SWD, W1, Zero, and Std use all modeled genes; P-MMD uses training-fitted PCA. $r^{200}_{\Delta C}$ and $r^{500}_{\Delta C}$ use 200 and 500 training-selected response genes, respectively, with each panel shared across conditions and methods within a split.

Method	MMD↓	P-MMD↓	SWD↓	W1↓	Zero↓	Std↓	$r^{200}_{\Delta C}$↑	$r^{500}_{\Delta C}$↑
Control cells	0.0079	0.0478	0.0937	0.0760	0.0593	0.0477	0.0001	0.0003
SLIM	<u>0.0051</u>	<u>0.0265</u>	<u>0.0858</u>	<u>0.0737</u>	<u>0.0573</u>	0.0542	0.0897	0.0678
GEARS	0.0069	0.0373	0.0908	0.0900	0.6104	<u>0.0477</u>	0.0001	0.0003
Scouter	0.0717	0.1700	0.3086	0.3080	0.6104	0.3473	0.0211	0.0155
CellFlow	0.0505	0.0330	0.2238	0.2510	0.6104	0.2832	0.0974	0.0799
scGPT	0.0730	0.1727	0.3050	0.3225	0.6104	0.3824	0.0217	0.0214
PerturbDiff	0.0090	0.0621	0.1014	0.1642	0.1083	0.1817	0.0704	0.0588
STATE	0.0059	0.0288	0.0871	0.1093	0.1291	0.1097	<u>0.1157</u>	**0.0942**
PerturbRIFT	**0.0046**	**0.0220**	**0.0816**	**0.0670**	**0.0504**	**0.0450**	**0.1179**	<u>0.0857</u>

Table 8: **RPE1: complete mean-response results.** Notation and aggregation follow Table 6. All values are reported on their original scale.

Method	Top100				allHVG			All genes
	ρ_Δ↑	$\hat{\rho}_\Delta$↑	NMSE↓	MAE↓	ρ_Δ↑	$\hat{\rho}_\Delta$↑	NMSE↓	PDS↑
No change	NA	-0.317	1.000	0.383	NA	0.051	1.000	0.500
Train mean	0.726	NA	0.529	0.222	0.616	NA	1.048	0.501
RidgeSC	<u>0.749</u>	<u>0.346</u>	<u>0.497</u>	<u>0.196</u>	0.617	<u>0.304</u>	1.000	0.638
SLIM	0.738	0.308	0.505	0.199	<u>0.621</u>	0.300	0.969	0.622
GEARS	0.699	0.156	0.598	0.208	0.558	0.108	1.399	0.580
TxPert-public	0.667	0.142	0.593	0.225	0.554	0.209	1.073	0.590
Scouter	0.674	0.249	0.606	0.213	0.541	0.252	1.188	0.658
CellFlow	0.702	0.122	0.604	0.223	0.578	0.154	1.141	0.564
scGPT	0.693	0.059	0.558	0.227	0.571	0.161	0.994	0.547
PerturbDiff	0.688	0.287	0.608	0.212	0.547	0.280	1.189	<u>0.680</u>
STATE	0.733	0.088	0.526	0.219	0.607	0.230	**0.893**	0.603
PerturbRIFT	**0.788**	**0.475**	**0.423**	**0.168**	**0.653**	**0.373**	<u>0.958</u>	**0.728**

Table 9: **RPE1: complete single-cell distributional results.** Notation and aggregation follow Table 7. All values are reported on their original scale.

Method	MMD↓	P-MMD↓	SWD↓	W1↓	Zero↓	Std↓	$r^{200}_{\Delta C}$↑	$r^{500}_{\Delta C}$↑
Control cells	0.0178	0.0926	0.1255	0.1001	0.0791	<u>0.0605</u>	0.0006	-0.0001
SLIM	<u>0.0096</u>	<u>0.0414</u>	<u>0.1042</u>	<u>0.0870</u>	<u>0.0703</u>	0.0643	<u>0.1712</u>	0.1255
GEARS	0.0127	0.0587	0.1158	0.1235	0.6450	<u>0.0605</u>	0.0006	-0.0001
Scouter	0.0751	0.1962	0.2988	0.2891	0.6450	0.3263	0.0703	0.0450
CellFlow	0.0462	0.0528	0.2021	0.2314	0.6450	0.2461	0.1376	0.1071
scGPT	0.0770	0.2088	0.3107	0.3087	0.6450	0.3646	0.0235	0.0284
PerturbDiff	0.0139	0.0602	0.1199	0.1319	0.1020	0.1333	0.1277	0.0991
STATE	0.0104	0.0442	0.1049	0.1113	0.1248	0.1077	0.1664	<u>0.1312</u>
PerturbRIFT	**0.0089**	**0.0354**	**0.1003**	**0.0810**	**0.0668**	**0.0563**	**0.2330**	**0.1686**

Table 10: **HepG2: complete mean-response results.** Notation and aggregation follow Table 6. All values are reported on their original scale.

Method	Top100				allHVG			All genes
	$\rho_\Delta\uparrow$	$\hat{\rho}_\Delta\uparrow$	NMSE↓	MAE↓	$\rho_\Delta\uparrow$	$\hat{\rho}_\Delta\uparrow$	NMSE↓	PDS↑
No change	NA	-0.204	1.000	0.254	NA	0.111	**1.000**	0.500
Train mean	0.409	NA	0.850	0.217	0.325	NA	1.038	0.501
RidgeSC	0.458	0.344	0.759	0.184	0.340	0.268	1.061	0.621
SLIM	0.439	0.314	0.772	0.187	0.340	0.268	1.028	0.607
GEARS	0.347	0.110	0.865	0.202	0.256	0.180	1.118	0.580
TxPert-public	0.391	0.212	0.946	0.190	0.298	0.124	1.834	0.579
Scouter	0.383	0.254	0.823	0.183	0.285	0.224	1.301	0.640
CellFlow	0.345	0.122	0.830	0.199	0.261	0.163	1.126	0.563
scGPT	0.384	0.151	0.839	0.202	0.276	0.160	1.114	0.559
PerturbDiff	0.409	0.265	0.849	0.187	0.292	0.262	1.318	0.653
STATE	0.414	0.153	0.791	0.188	0.308	0.209	1.076	0.604
PerturbRIFT	**0.506**	**0.377**	**0.714**	**0.166**	**0.369**	**0.285**	1.094	**0.686**

Table 11: **HepG2: complete single-cell distributional results.** Notation and aggregation follow Table 7. All values are reported on their original scale.

Method	MMD↓	P-MMD↓	SWD↓	W1↓	Zero↓	Std↓	$r^{200}_{\Delta C}\uparrow$	$r^{500}_{\Delta C}\uparrow$
Control cells	0.0104	0.0372	0.1103	0.0875	0.0664	0.0533	-0.0002	0.0003
SLIM	0.0077	0.0251	0.1045	0.0853	0.0648	0.0604	0.0896	0.0654
GEARS	0.0088	0.0285	0.1065	0.0974	0.5920	**0.0533**	-0.0002	0.0003
Scouter	0.0794	0.1774	0.3168	0.3089	0.5920	0.3517	0.0425	0.0260
CellFlow	0.0472	0.0327	0.2086	0.2398	0.5920	0.2621	0.0854	0.0656
scGPT	0.0818	0.1883	0.3290	0.3296	0.5920	0.3914	0.0127	0.0122
PerturbDiff	0.0114	0.0399	0.1165	0.1482	0.1116	0.1580	0.0716	0.0565
STATE	0.0083	0.0261	0.1046	0.1175	0.1261	0.1242	0.0953	0.0779
PerturbRIFT	**0.0075**	**0.0235**	**0.1019**	**0.0814**	**0.0618**	0.0542	**0.1026**	**0.0800**

Table 12: **Jurkat: complete mean-response results.** Notation and aggregation follow Table 6. All values are reported on their original scale.

Method	Top100				allHVG			All genes
	$\rho_\Delta\uparrow$	$\hat{\rho}_\Delta\uparrow$	NMSE↓	MAE↓	$\rho_\Delta\uparrow$	$\hat{\rho}_\Delta\uparrow$	NMSE↓	PDS↑
No change	NA	-0.164	1.000	0.194	NA	0.065	1.000	0.501
Train mean	0.343	NA	0.907	0.178	0.260	NA	0.994	0.501
RidgeSC	0.429	0.342	0.797	0.155	0.296	0.220	1.004	0.581
SLIM	0.412	0.315	0.810	0.157	0.299	0.222	0.971	0.571
GEARS	0.244	0.025	1.022	0.177	0.182	0.039	1.369	0.535
TxPert-public	0.419	0.246	0.826	0.157	0.291	0.219	**0.969**	0.578
Scouter	0.371	0.258	0.844	0.156	0.236	0.169	1.149	0.585
CellFlow	0.308	0.110	0.892	0.169	0.210	0.117	1.056	0.536
scGPT	0.286	0.095	0.940	0.173	0.184	0.065	1.240	0.530
PerturbDiff	0.363	0.266	0.871	0.159	0.218	0.205	1.171	0.600
STATE	0.371	0.197	0.839	0.157	0.256	0.109	1.115	0.566
PerturbRIFT	**0.495**	**0.414**	**0.738**	**0.138**	**0.326**	**0.249**	1.039	**0.644**

Table 13: **Jurkat: complete single-cell distributional results.** Notation and aggregation follow Table 7. All values are reported on their original scale.

Method	MMD↓	P-MMD↓	SWD↓	W1↓	Zero↓	Std↓	$r^{200}_{\Delta C}$↑	$r^{500}_{\Delta C}$↑
Control cells	0.0056	0.0230	0.0841	0.0664	0.0557	0.0465	-0.0012	-0.0005
SLIM	0.0043	0.0161	0.0796	0.0652	0.0546	0.0495	0.0428	0.0368
GEARS	0.0060	0.0252	0.0859	0.0815	0.6374	0.0465	-0.0012	-0.0005
Scouter	0.0651	0.1685	0.3019	0.2936	0.6374	0.3408	0.0103	0.0087
CellFlow	0.0425	0.0208	0.2045	0.2312	0.6374	0.2676	0.0500	0.0400
scGPT	0.0661	0.1692	0.2931	0.3038	0.6374	0.3671	0.0088	0.0102
PerturbDiff	0.0071	0.0290	0.0915	0.1305	0.1036	0.1406	0.0310	0.0260
STATE	0.0047	0.0183	0.0824	0.0986	0.1165	0.1105	0.0623	**0.0530**
PerturbRIFT	**0.0042**	**0.0158**	**0.0789**	**0.0630**	**0.0523**	**0.0463**	**0.0684**	0.0496

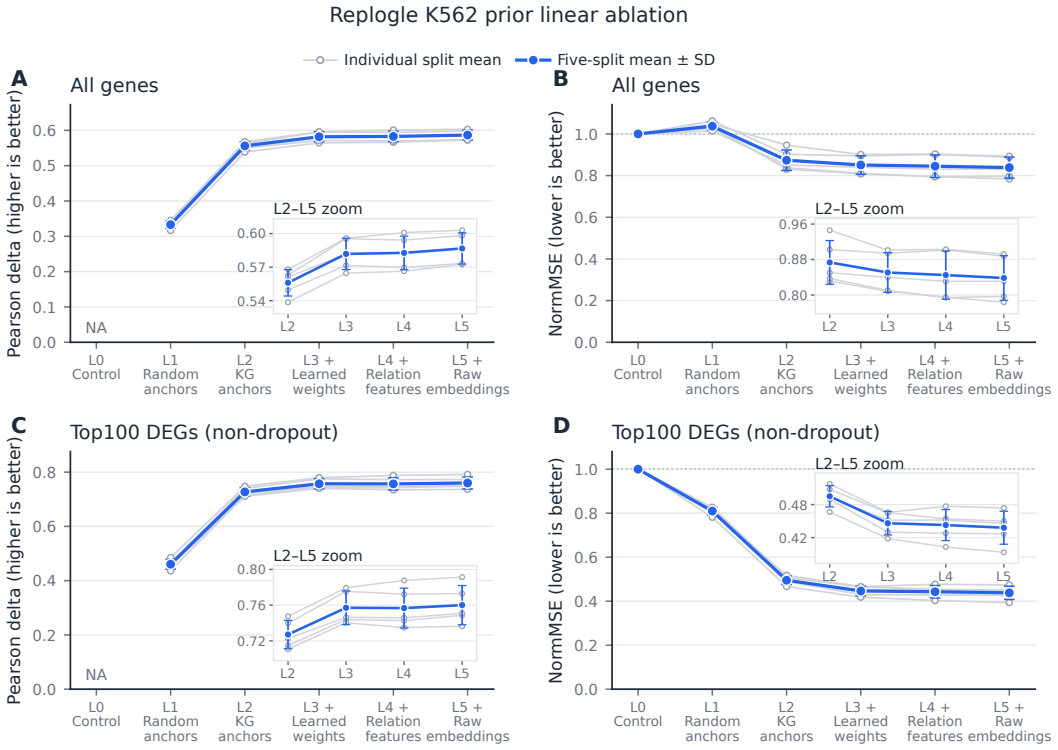


Figure 5: **Sequential ablation of the retrieved analytic mean on K562.** Panels A–B evaluate all genes and C–D evaluate Top100 non-dropout DEGs. Grey trajectories show individual split means; blue points and error bars show the equal-weight mean and SD across five correlated splits, after averaging learned-arm seed scores within each query. L0 Pearson delta is undefined because its predicted change is zero. Insets expand the L2–L5 scale.

D.3 Extended prior and residual ablations

Prior-mean estimation.

We compare six nested configurations on K562 to examine how biological relationships, learned weighting, and representation inputs contribute to prior-mean accuracy (Figure 5). Starting from the control mean (L0), the configurations introduce uniform aggregation over random training anchors (L1), knowledge-guided neighborhood selection (L2), learned weighting (L3), relation features (L4), and additional query and anchor embeddings (L5). Anchor-based configurations use 16 eligible training perturbations per query, excluding the query itself and interventions targeting the same gene.

Table 14: **Prior analytic-mean ablation.** Values are equal-weight five-split means ± SD. L0 correlation is undefined. Learned-arm scores average three seeds within each query before split aggregation.

Configuration	All genes		Top100 non-dropout DEGs	
	$\rho_\Delta \uparrow$	NormMSE$\downarrow$	$\rho_\Delta \uparrow$	NormMSE$\downarrow$
L0 Control	NA	1.0000 ± 0.0000	NA	1.0000 ± 0.0000
L1 Random anchors	0.3330 ± 0.0107	1.0370 ± 0.0189	0.4605 ± 0.0179	0.8094 ± 0.0177
L2 KG anchors	0.5560 ± 0.0118	0.8736 ± 0.0493	0.7271 ± 0.0160	0.4952 ± 0.0194
L3 + Weighting	0.5818 ± 0.0140	0.8509 ± 0.0446	0.7571 ± 0.0188	0.4462 ± 0.0213
L4 + Relations	0.5828 ± 0.0150	0.8450 ± 0.0543	0.7567 ± 0.0223	0.4428 ± 0.0282
L5 + Embeddings	0.5867 ± 0.0139	0.8384 ± 0.0503	0.7600 ± 0.0219	0.4380 ± 0.0300

Table 15: **Adjacent gains in the prior ablation.** Positive values indicate improvement: current minus previous for Pearson delta and previous minus current for NormMSE. Entries show the mean ± SD of five paired split-level differences.

Transition	All genes		Top100 non-dropout DEGs	
	$\Delta\rho_\Delta \uparrow$	NormMSE gain$\uparrow$	$\Delta\rho_\Delta \uparrow$	NormMSE gain$\uparrow$
$L0 \to L1$	NA	-0.0370 ± 0.0189	NA	$+0.1906 \pm 0.0177$
$L1 \to L2$	$+0.2230 \pm 0.0026$	$+0.1634 \pm 0.0394$	$+0.2666 \pm 0.0094$	$+0.3143 \pm 0.0165$
$L2 \to L3$	$+0.0258 \pm 0.0047$	$+0.0227 \pm 0.0146$	$+0.0300 \pm 0.0043$	$+0.0490 \pm 0.0072$
$L3 \to L4$	$+0.0009 \pm 0.0027$	$+0.0059 \pm 0.0101$	-0.0004 ± 0.0052	$+0.0033 \pm 0.0103$
$L4 \to L5$	$+0.0039 \pm 0.0014$	$+0.0066 \pm 0.0073$	$+0.0033 \pm 0.0023$	$+0.0048 \pm 0.0033$

L0–L2 are evaluated once per split, whereas L3–L5 are trained independently using seeds 24, 41, and 73. For learned configurations, scores are first averaged across seeds within each query, then across queries within each split, and finally equally across the five splits. Tables 14 and 15 report mean performance and differences between adjacent configurations, respectively, together with standard deviations across splits. All configurations are evaluated using the analytic prior mean before source sampling or residual generation.

Residual generation and sampling.

We evaluate residual generation on five K562 splits using seed 24, retaining conditions with at least 50 observed cells. The evaluation includes 968 split–condition pairs covering 649 distinct perturbations, with 256 generated cells over 5000 genes per condition.

With the response-prior estimator and autoencoder held fixed, we compare direct samples from the factorized prior (R0), an encoder–decoder round trip without flow (R1), FM alone (R2), and FM with latent RBF-MMD weighted by 0.5 (R3). R4 uses the same R3 model and transported endpoint, replacing independent decoder-mask sampling with Cube sampling (Appendix B.4). An additional control-shift reference ($R0'$) generates cells as

$$\hat{x} = \max(0, x_{\text{ctrl}} + m_0 - m_{\text{ctrl}}),$$

using sampled training-control cells and the query-specific prior mean. All generated populations undergo nonnegative clamping.

Condition-level scores are averaged within each split and then equally across the five splits (Table 3). The evaluated condition set differs from that of the main benchmark. Figure 6 provides condition-level examples of predicted gene-correlation changes.

E Limitations and future directions

Source distribution design for sparse biological data. A promising direction is to explore source distribution design as a broader principle for flow matching in high-dimensional sparse biological data. Source distributions that encode sparsity, coordinated gene programs, and cell-state variation

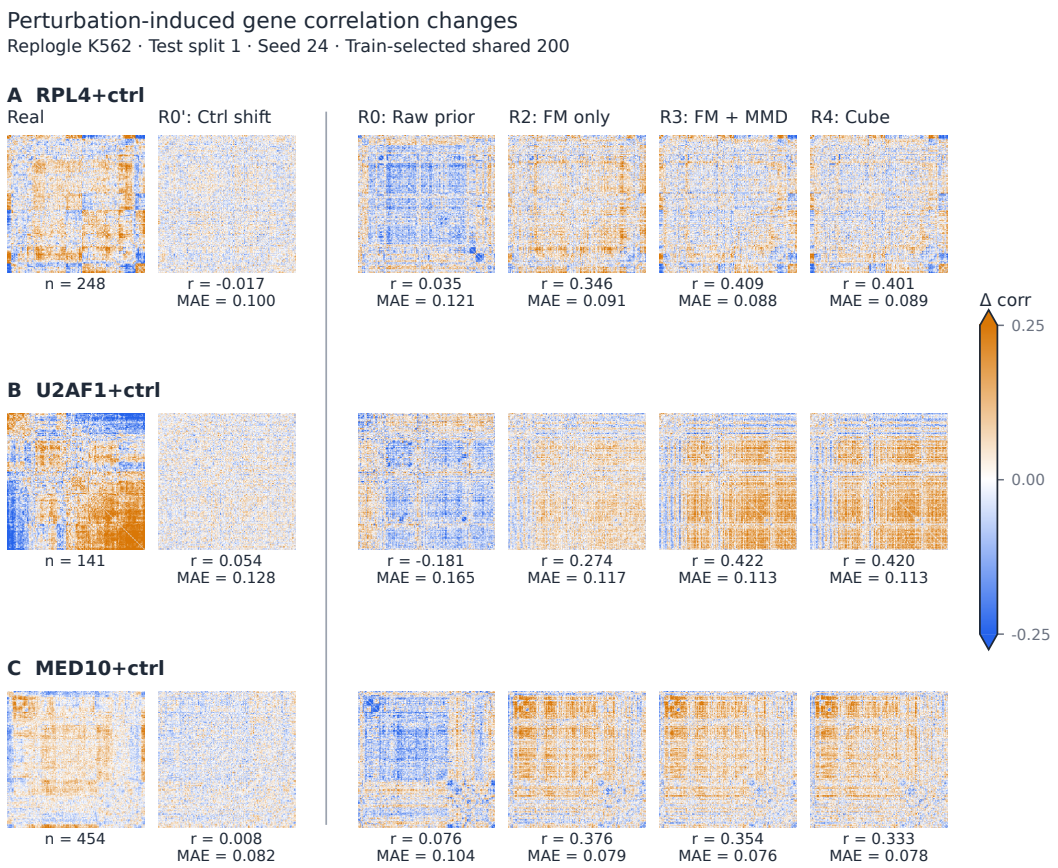


Figure 6: **Examples of predicted gene-correlation changes on K562.** Rows show RPL4, U2AF1, and MED10 in split 1 with seed 24. Columns compare observed cells with the control-shift reference, raw prior, FM alone, FM with latent MMD, and Cube sampling. Colors represent $\Delta C = C_{\text{population}} - C_{\text{control}}$ on the same 200 training-selected genes. Annotations report off-diagonal correlation agreement r and MAE relative to the observed ΔC. Examples use all retained observed cells (248, 141, and 454, respectively).

could provide biologically informed starting points for generation. Extending this principle to other single-cell modalities and molecular profiles may open new opportunities for efficient generative modeling guided by domain-specific structure.

Extending response-informed generation across contexts. Building on the within-dataset single-gene setting studied here, future work could develop context-aware response priors that integrate reference experiments across cellular backgrounds. A central question is how to combine transferable response statistics with context-specific information as reference coverage varies. Combinatorial perturbations offer a complementary direction: measured single-gene responses could inform priors for joint interventions, while both prior construction and conditional generation could be adapted to model non-additive effects.

Adaptive prior construction and conditional generation. Another direction is to make prior estimation and generative refinement responsive to the evidence available for each query. Reference-neighborhood coverage and agreement among measured training responses could guide how strongly prior statistics are retained and how much flexibility is introduced during generation. For example, softly constrained mean calibration could allow learned adjustments while retaining guidance from the estimated response prior. Such designs would provide a way to adapt the balance between prior information and conditional refinement to individual queries.

Distribution parameterization and population-level evaluation. Future evaluation could connect distributional fidelity more directly to biologically interpretable response patterns. Higher-order expression structure, coordinated gene programs, and rare response states provide complementary targets beyond pairwise gene-correlation changes. Assessing these properties could help characterize how well generated populations reproduce the organization and heterogeneity of perturbation responses.