DRIFT: DISENTANGLED RESPONSIVE-INVARIANT FLOW TRANSPORT FOR SINGLE-CELL PERTURBATION PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting cellular responses to perturbations is a central problem in cellular biology, with broad applications in systems biology and drug discovery. This task is challenging because cellular responses can be complex and cell-state dependent, intrinsic cell-to-cell variability can be confounded with perturbation effects, and destructive single-cell RNA sequencing precludes paired measurements of the same cell before and after treatment. Flow matching transports control cells to perturbed states flexibly, but acting on the full cell state can confound perturbation effects with pre-existing cell-to-cell variability. Disentangled approaches separate responsive from invariant components, but model perturbations through prescribed mechanisms, such as latent shifts or graph edits, limiting their flexibility. We address both limitations in a unified framework. A variational encoder disentangles each cell into an invariant block, capturing state unaffected by the perturbation, and a responsive block, capturing state it changes, through conditional priors and an information-theoretic invariance constraint. Conditional flow matching transports only the responsive block, conditioned on the perturbation and invariant state, yielding a flexible, data-driven model of perturbation effects without confounding pre-existing variability. Across several benchmarks, our method outperforms the strongest published method in settings involving combinatorial and unseen perturbation prediction.

1 INTRODUCTION

A perturbation experiment measures what a cell does when a gene is activated or silenced, or when a drug is applied. In a pooled screen (Dixit et al., 2016; Replogle et al., 2020; 2022; Srivatsan et al., 2020), a large population of cells is treated at once, each cell receives a single perturbation identified by a sequencing barcode, and the transcriptome of every cell is measured afterwards. The result is two kinds of data: untreated control cells, and perturbed cells grouped by which perturbation treatment they received. The prediction task is to predict the distribution of transcriptomes that control cells would exhibit under unseen or combinatorial perturbations.

Such a model would let researchers predict, in silico, how cells respond to perturbations they have never been exposed to (Bunne et al., 2023; Lotfollahi et al., 2023). This would allow researchers to screen perturbations computationally before applying them in vitro, prioritize combinations in spaces too large to explore experimentally, and estimate the effect of treatments for which no experiment exists. These capabilities are important for drug discovery, precision medicine, and cell engineering (Bunne et al., 2023; Lotfollahi et al., 2023).

These screens have made training data abundant, but the measurement itself withholds the very quantity a per-cell model requires. Sequencing destroys the cell it measures (Kester & Van Oudenaarden, 2018; Chen et al., 2019; Schiebinger et al., 2019; Demetci et al., 2022) which results in unpaired control and treatment populations which prohibit the measurement of a per-cell causal effect and the outcome of cell under treatment is never measured. Therefore, every approach in the field relies on assumptions to bridge the gap between the observed data and the quantity of interest.

Existing approaches close this gap with different modeling choices. Statistical methods predict the response with simple estimators, such as the control mean or a linear map, and carefully tuned

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.

linear baselines are competitive with respect to far more elaborate models (Ahlmann-Eltze et al., 2025). Foundation models pretrained on large unperturbed corpora (Cui et al., 2024; Theodoris et al., 2023; Rosen et al., 2026; Yang et al., 2022; Hao et al., 2024; Yang et al., 2024; Zeng et al., 2025; Khan et al., 2023) supply broad representations but are repeatedly found to miss perturbation-specific effects (Ahlmann-Eltze et al., 2025; Csendes et al., 2025; Kedzierska et al., 2023), which has prompted more discriminating evaluation protocols (Viñas Torné et al., 2025; Roohani et al., 2025) that we adopt. Graph-based models reach unseen perturbations by placing genes in a structure taken from external biology: a gene co-expression or Gene Ontology graph (Roohani et al., 2023; Kamimoto et al., 2023; Wu et al., 2023), or gene modules shared across perturbations (Ruan et al., 2026). Generative models learn the distribution of the whole perturbed cell state, in expression space or in a latent space (Adduri et al., 2025; Yu et al., 2026). Transport methods (Schiebinger et al., 2019; Bunne et al., 2023; Klein et al., 2023; 2025; Ruan et al., 2026) model the shift of an entire distribution from control to perturbed. Disentangled and causal latent-variable models (Lotfollahi et al., 2023; 2019; Weinberger et al., 2023; Yu et al., 2025; Xing & Yau, 2025; Bereket & Karaletsos, 2023; Lopez et al., 2022; Jiang et al., 2026) instead isolate a latent axis along which a perturbation acts. This factorization is intended to enable extrapolation to unobserved perturbation combinations (Lotfollahi et al., 2023; Bereket & Karaletsos, 2023), generalization to held-out cell states (Aliee et al., 2024; Zhang et al., 2024), and a causal interpretation of the perturbation factor (Lopez et al., 2022; Gao et al., 2025). Taken together, these methods either model the full cell state or impose a disentanglement scheme with a preselected, potentially inflexible perturbation mechanism.

We focus on two complementary approaches. Transport methods assume minimal displacement between control and perturbed cells and learn the transport map from data, but move the whole cell representation, either in a latent space or in the original expression space (Bunne et al., 2023; Klein et al., 2025; Yu et al., 2026). The invariant part of the cell then moves together with the responsive part, and the perturbation is confounded with the cell characteristics that are invariant to treatment. Disentanglement based models separate these invariant characteristics from the responsive state, but they generally assume a causal mechanism for how the perturbation acts: an additive vector (Lotfollahi et al., 2023), a sparse mechanism shift (Lachapelle et al., 2022; Bereket & Karaletsos, 2023), or an edit to a latent causal graph (An et al., 2025). These approaches commit to an inductive prior about perturbation dynamics and thus do not generalize to incompatible biological programs.

Our proposed method, which we call DRIFT (Disentangled Responsive-Invariant Flow Transport), separates the part of a cell that is responsive to the perturbation from the part that is invariant (Higgins et al., 2018; Winter et al., 2022; Garrido et al., 2023), and then models the perturbation mechanism as a per-cell transport that moves only the responsive part, leaving fixed the part carrying cell identity and the other perturbation-invariant factors. The disentanglement is learned following the auxiliary-variable method (Khemakhem et al., 2020a) and related lines of work (Khemakhem et al., 2020b; Sorrenson et al., 2020; Hyvarinen & Morioka, 2016; Lu et al., 2022; Aliee et al., 2024; Von Kügelgen et al., 2021), augmented with an information-theoretic constraint of invariance. The dynamics of the perturbation is modeled through a conditional flow (Lipman et al., 2022; Tong et al., 2023; Klein et al., 2025) that moves the variant block while keeping the invariant part fixed.

We summarize our contributions as follows. 1) We propose an information-theoretic disentanglement scheme that uses auxiliary-variable conditional priors together with a mutual-information-based invariance constraint to split the latent space of a cell into an invariant block, insensitive to the perturbation, and a variant block, that captures the perturbation-responsive part. 2) We use a conditional flow matching model, conditioned jointly on the perturbation and on the invariant latent of the cell, that transports the responsive (variant) part of the latent representation, leaving the invariant part fixed. 3) We obtain state-of-the-art results across multiple benchmarks: our method outperforms the strongest published method on every dataset we experimented with.

2 METHOD

DRIFT models the perturbation effect as a mechanism that changes part of a cell's state and leaves the remaining part unchanged. A cell is organized into an invariant block, holding the state a perturbation leaves untouched, and a variant (responsive) block, holding the state it changes, and the perturbation acts as a transport of the variant block alone. Our method unfolds in two stages (Figure 1). Stage 1 (§2.1 and §2.2) learns a disentangled representation: an encoder maps a cell to

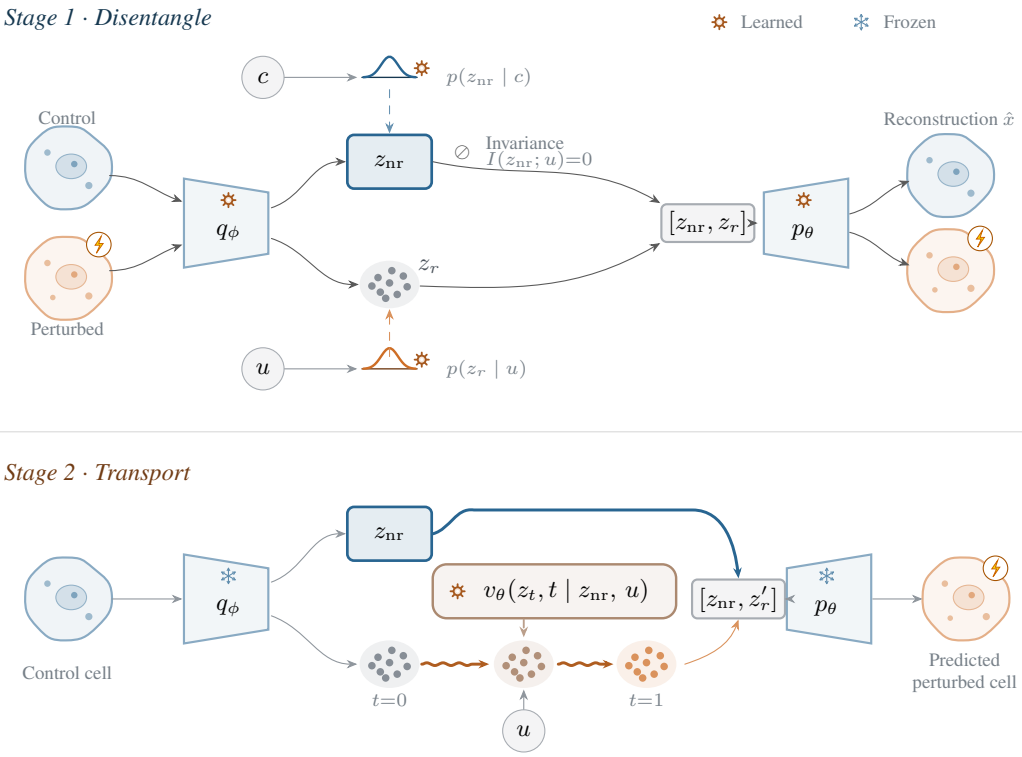


Figure 1: **Overview of DRIFT.** The two stages share one architecture. *Stage 1* trains a variational autoencoder on control and perturbed cells, encoding each into an *invariant* code z_{nr} (driven by the covariates c: cell line, batch, sequencing depth) and a *responsive* latent z_r (driven by the perturbation u). Each block is associated to a *learned* conditional prior, $p(z_{nr} \mid c)$ and $p(z_r \mid u)$, and z_{nr} is optimized to carry no perturbation information, $I(z_{nr}; u)=0$; the concatenation $[z_{nr}, z_r]$ is decoded to reconstruct the input, control or perturbed. *Stage 2* freezes the encoder and decoder and learns the mechanism $p(z_r \mid z_{nr}, u)$ as a transport: a conditional flow $v_\theta(z_t, t \mid z_{nr}, u)$ generates only the responsive latent z'_r from its control to its perturbed state along the interpolant z_t ($t: 0 \to 1$), while z_{nr} is copied from the input cell. Decoding $[z_{nr}, z'_r]$ yields the predicted perturbed cell.

the two blocks, each block is tied to a prior that depends on a different observed variable, and an information-theoretic constraint prevents perturbation information to leak into the invariant block. Stage 2 (§2.3) learns how a perturbation moves the responsive block, as a conditional flow. Inference (§2.4) combines the invariant state of an observed control cell with a response generated by the flow.

2.1 PROBLEM STATEMENT AND GENERATIVE MODEL

We observe single-cell RNA sequencing counts and write $x \in \mathbb{R}^G$ for the expression profile of one cell over G genes. A perturbation is the experimental intervention applied to that cell: which gene is activated or silenced by CRISPR (Jinek et al., 2012; Gilbert et al., 2014), or which chemical compound is added. We label a perturbation by u, with $u = \varnothing$ for an untreated control. The perturbation acts on the cell by changing its transcriptome, so it changes the distribution of x.

We also observe $c \in \mathbb{R}^K$, the factors that vary across cells and are not caused by the perturbation, such as the cell line, and technical properties of the measurement.

We are given unpaired populations of control cells and of cells under various u; c is observed per cell and the two populations are not assumed to share it. The task is to predict, for a target perturbation $u^\star$ that may be absent from training, the distribution of profiles that observed control cells from the same dataset would take under $u^\star$.

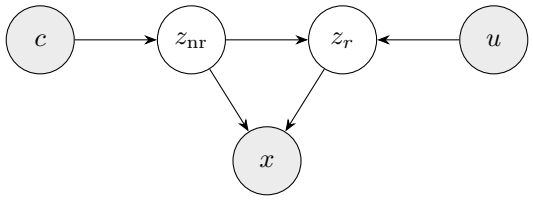


Figure 2: The generative model of DRIFT; shaded nodes are observed. The non-responsive block z_{nr} is generated from the assay covariates c; the responsive block z_r from the perturbation u and z_{nr} (the mechanism); the profile x from both. The non-responsive block is invariant to the perturbation, $z_{\mathrm{nr}} \perp u$ (Eq. 3), the property that defines it, while the mechanism $z_{\mathrm{nr}} \to z_r$ lets the response depend on the basal state.

We consider a latent state that splits into two blocks,

$$z = (z_{\mathrm{nr}}, z_r) \in \mathbb{R}^{d_{\mathrm{nr}}} \times \mathbb{R}^{d_r}, \tag{1}$$

where z_{nr} is the *invariant* (non-responsive) block, holding the state a perturbation leaves untouched, and z_r is the *variant* (responsive) block, holding the state a perturbation moves.

A decoder $p_\theta(x \mid z_{\mathrm{nr}}, z_r)$ maps the two blocks back to an expression profile. The generative model is the chain of Fig. 2,

$$p(x, z_{\mathrm{nr}}, z_r \mid u, c) = \underbrace{p_\theta(x \mid z_{\mathrm{nr}}, z_r)}_{\text{decoder}}\ \underbrace{p(z_r \mid z_{\mathrm{nr}}, u)}_{\text{mechanism}}\ \underbrace{p(z_{\mathrm{nr}} \mid c)}_{\text{invariant state}}, \tag{2}$$

with edges $\{c \to z_{\mathrm{nr}},\ z_{\mathrm{nr}} \to z_r,\ u \to z_r,\ z_{\mathrm{nr}} \to x,\ z_r \to x\}$ and $u \perp c$. The invariant block is generated from the perturbation-independent factors c; the responsive block from the perturbation and from the invariant block of the same cell; and the expression profile from both blocks. In our model, we encourage the invariant block to be independent of the perturbation,

$$z_{\mathrm{nr}} \perp u, \tag{3}$$

through a mutual-information penalty (§2.2). The mechanism $p(z_r \mid z_{\mathrm{nr}}, u)$ keeps z_r dependent on z_{nr}, which lets the response vary with the basal state. Stage 1 (§2.2) learns the two blocks and enforces (3); stage 2 (§2.3) learns the mechanism as a transport.

2.2 STAGE 1: DISENTANGLING THE VARIANT AND INVARIANT BLOCKS

Disentanglement learned without supervision or inductive bias does not reliably recover meaningful factors (Locatello et al., 2019; Hyvärinen & Pajunen, 1999), which motivates tying each block to an observed variable. In the causal representation learning view (Schölkopf et al., 2021), a perturbation is a soft intervention: it changes the mechanism that generates a latent variable without fixing its value (Zhang et al., 2023). Following the auxiliary-variable construction (Khemakhem et al., 2020a; Aliee et al., 2024), each block is given a prior that depends on one observed variable only, which we assume to be Gaussian whose mean and (diagonal) covariance are functions of that variable,

$$p(z_{\mathrm{nr}} \mid c) = \mathcal{N}\big(\mu_{\mathrm{nr}}(c),\ \mathrm{diag}\,\sigma^2_{\mathrm{nr}}(c)\big), \qquad p(z_r \mid u) = \mathcal{N}\big(\mu_r(u),\ \mathrm{diag}\,\sigma^2_r(u)\big), \tag{4}$$

that is, they are members of the conditional exponential family of Khemakhem et al. (2020a) with Gaussian sufficient statistics. The priors are learned jointly with the encoder and decoder. The encoder is also fed with the auxiliary variables u and c.

The encoder maps a cell to Gaussian posteriors over the two blocks conditioned on the cell and its auxiliaries, $q_\phi(z_{\mathrm{nr}} \mid x, u, c)$ and $q_\phi(z_r \mid x, u, c)$, following Aliee et al. (2024), and the decoder reconstructs x from both blocks. Writing q_ϕ for the joint posterior, we maximize the ELBO

$$\begin{aligned}\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi}\big[\log p_\theta(x \mid z_{\mathrm{nr}}, z_r)\big] - \beta_t \Big(&\mathrm{KL}\big[q_\phi(z_{\mathrm{nr}} \mid x, u, c) \,\|\, p(z_{\mathrm{nr}} \mid c)\big] \\ &+ \mathrm{KL}\big[q_\phi(z_r \mid x, u, c) \,\|\, p(z_r \mid u)\big]\Big),\end{aligned} \tag{5}$$

subject to the invariance constraint below. Stage 1 regularizes z_r toward the marginal prior $p(z_r \mid u)$ instead of the mechanism $p(z_r \mid z_{\mathrm{nr}}, u)$ of (2). The dependence of the response on the basal state is deferred to the stage-2 flow.

Enforcing invariance. The conditional prior anchors each block to its auxiliary, but reconstruction does not prevent the invariant block from also carrying the perturbation. We therefore encourage (3) by penalizing mutual information between z_{nr} and u (Moyer et al., 2018),

$$\min_{\phi} \; I(z_{\mathrm{nr}}; u), \tag{6}$$

using the contrastive log-ratio upper bound (CLUB) of Cheng et al. (2020),

$$\mathcal{L} \;\leftarrow\; \mathcal{L} \;-\; \lambda\, I_{\mathrm{CLUB}}(z_{\mathrm{nr}}; u). \tag{7}$$

This penalty encourages perturbation-dependent factors to concentrate in the responsive block, helping make z_r the coordinate transported in stage 2.

Regularizing the perturbation encoder.

The perturbation identity u is fed to the model through an embedded representation e_u, produced by a network that maps pretrained features of the targeted gene to a vector. Observed perturbations, available at training time, are paired with their corresponding perturbed cell population, whereas unseen perturbations have no example of the resulting outcome. Since the model must generalize to unseen genes, e_u alone determines its behavior in the out-of-distribution regime, which requires the representation to be well structured so that unseen perturbations are meaningfully embedded and predictive of their effects. To this end, we regularize e_u with three complementary terms. A linear head predicts the mean measured response of each training condition from its code, tying the embedding to the response it must explain. A second term constrains pairwise distances between codes to follow distances between the corresponding mean measured responses, so that perturbations with similar effects receive similar codes. Finally, Gaussian noise injected into the embedding during training prevents the network from collapsing into a lookup table of training conditions. These terms are auxiliary to the main objective in (5) and are formally defined in Appendix C.

Training. The CLUB bound is evaluated with an auxiliary predictor $q_\psi(u \mid z_{\mathrm{nr}})$, trained to keep the bound tight while the encoder minimizes the bound $I_{\mathrm{CLUB}}(z_{\mathrm{nr}}; u)$. Two collapse modes must be guarded against. First, the invariant block's KL term can collapse the posterior onto the prior; we prevent this with a KL-annealing schedule, $\beta_t = \min(1, t/T_{\mathrm{w}})\,\beta$ (Higgins et al., 2017), which lets reconstruction dominate early so the block learns a useful representation before its KL cost is applied. Second, CLUB (Cheng et al., 2020) is a valid upper bound only once q_ψ has fit the current encoder; applying it from initialization drives z_{nr} to a trivial solution instead of an invariant one. We therefore ramp the weight of (7) up from zero over the same warm-up, while q_ψ takes several gradient steps per encoder step. Encoder, decoder, both priors, and q_ψ are then trained jointly by stochastic gradient descent. Stage 1 returns a disentangled variational autoencoder with an invariant latent kept fixed and a responsive block, which the stage 2 velocity field is trained to transport.

2.3 Stage 2: Transporting the responsive block

With the encoder frozen, stage 2 learns the conditional mechanism $p(z_r \mid z_{\mathrm{nr}}, u)$ in (2). Given the invariant block z_{nr} of a control cell and a perturbation u, it generates the responsive block the cell would exhibit under u. For each u, the responsive blocks of control cells and cells perturbed by u define two distributions, and we learn a conditional flow that transports the former to the latter while keeping z_{nr} fixed. This mirrors inference: the invariant block of an observed control cell is retained, and only its responsive block is generated. The invariance penalty (6) supports this construction by encouraging z_{nr} to have the same distribution in control and perturbed cells.

Conditional flow. Let e_u be the perturbation code. A velocity field $v_\theta(z, t \mid z_{\mathrm{nr}}, e_u)$ defines a transport through the ordinary differential equation $\dot{z} = v_\theta(z, t \mid z_{\mathrm{nr}}, e_u)$, integrated from $t{=}0$ to $t{=}1$. We fit the field with conditional flow matching (Lipman et al., 2022; Tong et al., 2023), a regression on the straight-line velocity between a source sample and a target sample that requires no simulation of the ordinary differential equation during training:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\,(z_0,z_1)\sim\pi}\big\| \, v_\theta(z_t, t \mid z_{\mathrm{nr}}, e_u) - (z_1 - z_0) \, \big\|^2, \qquad z_t = (1-t)z_0 + t z_1. \tag{8}$$

The source $z_0 = z_r^{\text{ctrl}}$ is the responsive block of a control cell, the target $z_1 = z_r^{\text{pert}}$ is the responsive block of a cell under u, and z_{nr} is the invariant block of the control cell. The field therefore displaces the responsive block of an observed cell. The expectation in (8) is taken over pairs drawn from a coupling π of the two populations. Any such coupling yields a flow that carries the control distribution onto the perturbed distribution (Tong et al., 2023); the coupling determines which control cell is carried onto which perturbed cell.

Pairing by minibatch entropic optimal transport. A perturbation experiment cannot record which perturbed cell originates from which control cell, so the pairs in (8) are unobserved. We approximate such pairing from a minibatch entropic optimal transport plan, as in OT-CFM (Tong et al., 2023). For each sampled perturbation u, we draw n control and n perturbed cells and solve

$$\pi^{\varepsilon} = \arg\min_{\pi \in \Pi_n} \sum_{i,j} \pi_{ij}\, C_{ij} - \varepsilon H(\pi), \qquad C_{ij} = \alpha_{\text{nr}} \left\| z_{\text{nr}}^{(i)} - z_{\text{nr}}^{(j)} \right\|^2 + \alpha_r \left\| z_r^{(i)} - z_r^{(j)} \right\|^2, \quad (9)$$

where i indexes control cells, j indexes perturbed cells, Π_n is the set of couplings with uniform marginals, H is the entropy, and $\varepsilon > 0$ is the regularization strength. We solve (9) with Sinkhorn iterations (Cuturi, 2013) and pair each perturbed cell with a control cell sampled from its column of π^{ε}. The α_{nr} term favors control cells with a similar invariant block, following the conditional optimal transport construction of Kerrigan et al. (2024), so the difference within a pair is ascribed to the perturbation. The α_r term favors pairs with a small displacement of the responsive block. The plan is solved again on fresh cells at every minibatch round, and its only role is to supply the pairs that (8) regresses on. The finite minibatch and the entropic regularization make π^{ε} an approximation of the optimal transport plan between the two populations. We stress that two distinct plans are at play: π^{ε}, approximated via minibatch entropic OT, only selects pairs for training the velocity field, whereas the velocity field itself learns the transport of the responsive block of control cells onto the responsive block of perturbed cells. At inference, this learned velocity field is used to transport the responsive block alone, via the scheme we detail next.

2.4 Inference

Predicting the response to a target perturbation $u^{\star}$ proceeds by drawing an observed control cell, encoding its invariant block z_{nr}, integrating the flow conditioned on $(z_{\text{nr}}, e_{u^{\star}})$ to generate a responsive block z_r, and decoding the pair. The invariant state is copied from an observed control cell and only the response is generated. The prediction is well posed provided the control invariant state lies in the support the field has seen under $u^{\star}$, which the invariance (6) encourages. Each stage-1 term shapes the responsive block for the stage-2 transport. The conditional prior organizes z_r by perturbation, which lets the flow interpolate to unseen u. The invariance term (6) concentrates the response in z_r and makes the α_{nr} term of the coupling a closer measure of cell identity.

3 Experiments

We evaluate DRIFT on perturbation benchmarks that test combinatorial and zero-shot generalization to unseen perturbations. Norman (Norman et al., 2019) is a combinatorial benchmark with both single-gene and two-gene perturbations, and Replogle (Replogle et al., 2022) is a zero-shot benchmark with single-gene perturbations. We compare DRIFT with multiple published methods from four paradigms. 1) Foundation models: scGPT (Cui et al., 2024), scFoundation (Hao et al., 2024), GeneCompass (Yang et al., 2024), and CellFM (Zeng et al., 2025). 2) Graph-based models: GEARS (Roohani et al., 2023), CellOracle (Kamimoto et al., 2023), and GraphVCI (Wu et al., 2023). 3) Generative models: CPA (Lotfollahi et al., 2023), samsVAE (Bereket & Karaletsos, 2023), STATE (Adduri et al., 2025), CellFlow (Klein et al., 2025), scDFM (Yu et al., 2026), and scBIG (Ruan et al., 2026). 4) Statistical methods: Control, Linear, and Linear-scGPT (Ahlmann-Eltze et al., 2025).

We measure the *shift* induced by a perturbation relative to control (Ahlmann-Eltze et al., 2025). Following Ruan et al. (2026), we report nine metrics in three groups: *correlation* between predicted and true shifts over all genes ($\rho\Delta$) and the 20 most differentially expressed genes ($\rho\Delta^{\text{D}}$); *directional agreement*, the fraction of genes with correctly signed shifts ($\text{ACC}\Delta$, $\text{ACC}\Delta^{\text{D}}$), together with DES, the rank correlation between predicted and true log fold changes over the genes a perturbation significantly moves; and *discriminability*, measured by PDS (Roohani et al., 2025). PDS ranks the true condition among candidates by their distance to the prediction, thereby penalizing responses

Table 1: Norman **additive** split. **Bold**: best; underline: second best.

Type	Method	$\rho\Delta\uparrow$	$\rho\Delta^{\mathrm{D}}\uparrow$	ACC$\Delta\uparrow$	ACC$\Delta^{\mathrm{D}}\uparrow$	DES$\uparrow$	PDS$\uparrow$	L2$\downarrow$	MSE$\downarrow$	MAE$\downarrow$
Sta.	Control	–	–	–	–	–	0.5000	9.25	0.0487	0.1720
Found.	Linear	0.6211	0.7075	0.8007	0.8906	0.3192	0.5363	6.76	0.0265	0.1233
	Linear-scGPT	0.6925	0.7514	0.8424	0.9688	0.5023	0.7913	5.31	0.0155	0.0942
	scGPT	0.4408	0.4416	0.8125	0.4839	0.2504	0.5022	7.35	0.0296	0.1285
	scFoundation	0.6813	0.4778	0.8768	0.7453	0.5409	0.7994	4.91	0.0138	0.0852
	GeneCompass	0.6897	0.4810	0.8916	0.7484	0.5948	0.8024	4.77	0.0124	0.0808
	CellFM	0.5947	0.5209	0.8819	0.8459	0.7550	0.7419	5.41	0.0161	0.0889
Gra.	GEARS	0.7134	0.5065	0.8916	0.7641	0.6220	0.8155	4.59	0.0117	0.0788
	CellOracle	0.0437	0.2567	0.4942	0.4536	0.0319	0.5277	11.10	0.0691	0.2001
	GraphVCI	0.5468	0.5722	0.8034	0.7905	0.6495	0.5151	6.68	0.0258	0.1177
Gen.	samsVAE	0.6497	0.7750	0.8470	0.9658	0.6138	0.6689	6.87	0.0329	0.1067
	STATE	0.2439	0.2628	0.8023	0.4141	0.4891	0.5171	9.02	0.0421	0.1343
	CellFlow	0.7892	0.7275	0.9151	0.9391	0.8113	0.7581	4.54	0.0114	0.0775
	scBIG	0.8496	0.8230	0.9197	0.9906	0.8593	0.8548	3.92	0.0091	0.0689
	DRIFT (ours)	**0.8871**	**0.8914**	**0.9222**	**0.9977**	**0.9017**	**0.9167**	**3.12**	**0.0057**	**0.0549**

that do not distinguish conditions despite high per-gene correlation. We also report pointwise errors (L_2, MSE, and MAE). All results are averaged over five seeds; standard deviations are reported in Appendix G. Definitions appear in Appendix A, implementation details in Appendix B.

3.1 Results: Norman

Norman (Norman et al., 2019) is a CRISPRa (Gilbert et al., 2014) screen in K562 cells, where one or two genes are targeted and the transcriptome is measured in each cell. Norman is a standard benchmark focusing on *combinatorial* perturbation prediction: it includes both single-gene activations and gene pairs, allowing evaluation of whether a model can predict a joint effect after observing its constituent genes separately. The *additive* split holds out 32 two-gene combinations whose constituent genes are each observed individually during training, isolating combinatorial generalization without novel genes. The *holdout* split holds out 21 single-gene and 5 two-gene conditions.

Additive. Table 1 compares DRIFT with the baselines on the additive split. DRIFT achieves the best performance on all nine metrics and improves on the strongest baseline, scBIG, globally on all metrics. The largest gains appear where combinatorial generalization is more challenging. ρ^{D}_{Δ}, which considers the twenty genes most affected by each perturbation, improves by 8.3%; thus, the gain is not driven by the many genes with negligible responses. PDS improves by 7.2% over scBIG, indicating that the predicted shifts more clearly identify the condition that produced them. Pointwise errors also improve substantially: MSE by 37.2% and L_2 by 20.4%. Together, these results show that DRIFT captures both the magnitude and the condition-specific structure of combinatorial perturbation responses.

Holdout split. The holdout split is even more challenging: nine of its 26 test conditions involve a gene absent from training, requiring zero-shot generalization; the double perturbations additionally test combinatorial generalization. Table 2 reports results separately for single and double perturbations. On *singles*, DRIFT outperforms scBIG on all nine metrics: $\rho^{\mathrm{D}}_{\Delta}$ rises by 7.8%, while MSE improves by 4.6%. These gains hold despite the need to extrapolate to gene perturbations unseen during training. On *doubles*, the additive-split pattern reappears more strongly: DRIFT outperforms scBIG on every metric by a wider margin than on singles. $\rho^{\mathrm{D}}_{\Delta}$ improves 6.9% and ACCΔ^{D} nearly saturates at 0.9873. PDS improves 8.5%, and the pointwise errors improve furthest of all, MSE by 28.9% and L_2 by 15.8%. These results show that DRIFT superiority carries over when the constituent genes are observed, even within the more challenging holdout regime.

3.2 Results: Replogle RPE1

Replogle2022-RPE1 (Replogle et al., 2022) is a CRISPRi knockdown screen in RPE1 cells. Compared with Norman, it differs in cell type, perturbation modality (knockdown rather than activation), and scale (1,037 training conditions versus 190). All 384 test conditions are single-gene perturba-

Table 2: Norman **holdout** split. **Bold**: best; underline: second best.

Type	Method	$\rho\Delta\uparrow$	$\rho\Delta^{\mathrm{D}}\uparrow$	ACC$\Delta\uparrow$	ACC$\Delta^{\mathrm{D}}\uparrow$	DES↑	PDS↑	L2↓	MSE↓	MAE↓
Single gene perturbation										
Sta.	Control	–	–	–	–	–	0.5000	4.33	0.0108	0.0685
Sta.	Linear	0.5924	0.6021	0.7695	0.8476	0.4548	0.5119	3.46	0.0067	0.0516
Found.	Linear-scGPT	0.6072	0.6352	0.7620	0.8095	0.4584	0.6571	3.60	0.0072	0.0575
Found.	scGPT	0.5172	0.5358	0.7509	0.8286	0.5987	0.5048	3.58	0.0071	0.0530
Found.	scFoundation	0.4453	0.3144	0.6882	0.7429	0.3897	0.6619	3.65	0.0079	0.0568
Found.	GeneCompass	0.5307	0.3423	0.7006	0.7429	0.4543	0.5857	3.44	0.0066	0.0546
Found.	CellFM	0.4331	0.5226	0.6708	0.7119	0.5318	0.5524	3.83	0.0086	0.0600
Gra.	GEARS	0.4539	0.5345	0.7059	0.8024	0.3968	0.4905	3.77	0.0078	0.0574
Gra.	GraphVCI	0.3902	0.1865	0.7054	0.5583	0.1395	0.5000	3.28	0.0066	0.0500
Gra.	samsVAE	0.4778	0.4982	0.7050	0.8310	0.3366	0.3452	3.88	0.0081	0.0614
Gen.	STATE	0.3085	0.5207	0.7600	0.5763	0.5712	0.5048	5.94	0.0177	0.0737
Gen.	CellFlow	0.5930	0.4736	0.7778	0.8119	0.6280	0.6643	3.40	0.0076	0.0535
Gen.	scBIG	0.6507	0.6537	0.7876	0.8548	0.6333	0.7467	3.26	0.0065	0.0491
	DRIFT (ours)	**0.6690**	**0.7046**	**0.7890**	**0.8783**	**0.6904**	**0.7596**	**3.09**	**0.0062**	**0.0480**
Double gene perturbations										
Sta.	Control	–	–	–	–	–	0.5000	6.02	0.0200	0.0984
Sta.	Linear	0.7179	0.8277	0.8240	0.9200	0.5637	0.5000	4.31	0.0105	0.0656
Found.	Linear-scGPT	0.7351	0.7230	0.8282	0.9100	0.6118	0.7000	3.91	0.0080	0.0626
Found.	scGPT	0.6350	0.4719	0.8068	0.8333	0.4456	0.5000	4.70	0.0124	0.0726
Found.	scFoundation	0.6605	0.4445	0.7964	0.9200	0.6538	0.6500	3.91	0.0083	0.0589
Found.	GeneCompass	0.6437	0.6866	0.7471	0.7400	0.5905	0.6000	4.37	0.0108	0.0723
Found.	CellFM	0.5615	0.8151	0.7072	0.7600	0.6746	0.5000	4.89	0.0132	0.0804
Gra.	GEARS	0.5087	0.6578	0.7130	0.8300	0.4156	0.5000	5.00	0.0138	0.0788
Gra.	GraphVCI	0.4290	0.5429	0.6889	0.6533	0.4003	0.5048	4.61	0.0114	0.0725
Gra.	samsVAE	0.6267	0.6402	0.7539	0.8800	0.5183	0.6500	4.66	0.0118	0.0742
Gen.	STATE	0.3786	0.6379	0.8126	0.5800	0.7097	0.5000	5.94	0.0177	0.0737
Gen.	CellFlow	0.7404	0.6774	0.8388	0.8400	0.7636	0.4500	4.12	0.0097	0.0621
Gen.	scBIG	0.8026	0.8719	0.8552	0.9700	0.8313	0.7440	3.68	0.0083	0.0554
	DRIFT (ours)	**0.8562**	**0.9319**	**0.8724**	**0.9873**	**0.8734**	**0.8075**	**3.10**	**0.0059**	**0.0469**

Table 3: Replogle RPE1. **Bold**: best; underline: second best.

Type	Method	$\rho\Delta\uparrow$	$\rho\Delta^{\mathrm{D}}\uparrow$	ACC$\Delta\uparrow$	ACC$\Delta^{\mathrm{D}}\uparrow$	DES↑	PDS↑	L2↓	MSE↓	MAE↓
Sta.	Control	–	–	–	–	–	0.5000	7.03	0.0171	0.0768
Sta.	Linear	0.2720	0.3630	0.5816	0.6919	0.2149	0.5318	14.13	0.0916	0.1491
Found.	Linear-scGPT	0.3404	0.5438	0.5852	0.7681	0.2497	0.5338	7.60	0.0172	0.0890
Found.	scGPT	0.2840	0.5691	0.5882	0.7759	0.2980	0.5001	9.49	0.0246	0.1037
Found.	scFoundation	0.2037	0.3632	0.5489	0.6908	0.2331	0.5098	7.07	0.0155	0.0812
Found.	GeneCompass	0.3113	0.4299	0.5758	0.7148	0.2873	0.5038	7.18	0.0162	0.0822
Found.	CellFM	0.3060	0.4299	0.5733	0.7086	0.2819	0.5039	7.20	0.0163	0.0825
Gra.	GEARS	0.3083	0.4289	0.5752	0.7096	0.2820	0.5004	7.18	0.0162	0.0822
Gra.	GraphVCI	0.4294	0.5302	0.5589	0.6436	0.1863	0.5000	7.41	0.0157	0.0913
Gra.	samsVAE	0.4083	0.5204	0.5998	0.7565	0.3280	0.4984	6.86	0.0140	0.0780
Gen.	STATE	0.3638	0.5298	0.6463	0.7891	0.3711	0.5011	8.38	0.0194	0.0759
Gen.	CellFlow	0.3878	0.4391	0.6226	0.7194	0.3114	0.5441	6.48	0.0135	0.0711
Gen.	scBIG	0.4875	0.5925	0.6471	0.8089	0.5145	0.5520	6.12	0.0118	0.0676
	DRIFT (ours)	**0.5085**	**0.6160**	**0.6537**	**0.8149**	**0.7021**	**0.6044**	**5.68**	**0.0101**	**0.0633**

tions targeting genes absent from training, making this a fully *zero-shot* setting. We use the split of Ruan et al. (2026). DRIFT leads on all nine metrics (Table 3). It surpasses scBIG on both correlation metrics ρ_Δ and ρ_Δ^{D} and PDS. Its largest gain is in DES ($+36.5\%$), the rank agreement between predicted and true log fold changes over the genes a perturbation significantly moves. These results provide strong evidence that the disentangled formulation captures generalizable structure in perturbation responses across several datasets.

Table 4: ComboSciPlex. **Bold**: best; underline: second best.

Method	$\rho\Delta \uparrow$	DE-Spearman $\rho \uparrow$	DS↑	L2↓	MSE↓	MAE↓
Control	N.A.	N.A.	0.5714	5.3716	0.0324	0.0698
scGPT	0.8322	−0.1261	0.8571	1.6934	0.0031	0.0251
CPA	0.8150	0.7906	**0.8980**	1.6592	0.0029	0.0240
scDFM	0.8933	0.8289	0.8776	1.6567	0.0028	0.0220
DRIFT (ours)	**0.9467**	**0.8901**	0.8816	**1.2804**	**0.0019**	**0.0150**

3.3 Results: ComboSciPlex

ComboSciPlex (Srivatsan et al., 2020) is a drug molecule perturbation benchmark with 63,378 cells exposed to 17 compounds. Of its 31 perturbed conditions, 25 are compound pairs; 24 conditions are used for training and 7 are held out. We follow the protocol of Yu et al. (2026) and report the two metrics it uses in place of DES and PDS: DE-Spearman ρ, the rank correlation between predicted and measured log fold changes on the statistically significant differentially expressed genes, and the discrimination score (DS) of Roohani et al. (2025), which measures whether the predicted populations of different perturbations remain separated from one another. DRIFT leads on five of the six metrics. Against scDFM, the strongest baseline, it reduces pointwise errors by 23 to 33% and raises DE-Spearman ρ from 0.829 to 0.890 and $\rho\Delta$ from 0.893 to 0.947. In DS, DRIFT ranks ahead of scDFM and slightly below CPA, which performs worse on all remaining metrics. These results suggest that our formulation extends beyond genetic perturbations to combinations of drugs.

3.4 Ablations

We assess the contribution of each DRIFT component through ablation studies (Appendix F). Disentanglement drives most of the performance gains, improving every metric across datasets and making the largest contribution overall. The invariance penalty provides consistent complementary gains. Conditioning regularization is especially valuable in zero-shot settings, where test perturbations are entirely unseen, but is less consequential for combinatorial generalization, where test perturbations combine genes observed during training (details in Table 8).

4 Conclusion

We presented DRIFT, a two-stage model that disentangles a cell's invariant state from its perturbation-responsive component, then transports the latter while preserving the former. In the first stage, a variational autoencoder learns this disentangled representation through conditional priors informed by distinct auxiliary variables and an information-theoretic invariance constraint, thereby partitioning the latent state into invariant and responsive blocks. In the second stage, a conditional flow transports only the responsive component by learning from data a velocity field that captures perturbation-induced dynamics in the responsive latent block.

Across the Norman additive and holdout splits, the Replogle RPE1 screen, and the ComboSciPlex drug perturbation dataset, DRIFT outperforms the strongest published baseline. These improvements are consistent across tasks, on both genetic and chemical perturbation, from learning combinatorial effects from single gene perturbations to generalizing to unseen perturbations, and across correlation, directional agreement, and discriminability metrics. Together, these results suggest that the disentangled formulation captures a central feature of perturbation response: separating invariant and responsive components reduces confounding from cell-level variation and yields a more faithful representation of perturbation-induced dynamics.

Limitations. Although DRIFT is designed to accommodate substantial cell-to-cell variation, current benchmarks widely adopted in the literature are limited to single cell lines. Extending the model to multiple cell lines would test whether the invariant–responsive decomposition generalizes across diverse cellular backgrounds. Temporal measurements could also provide a biological interpretation for the flow's time coordinate, capture response dynamics, and connect endpoint mappings into trajectory models.

AI USE STATEMENT

In this work, we used generative AI tools to assist with LaTeX table formatting, TikZ figure refinement, and grammar and style editing; to generate figures and tables from stored results; to support code design, refinement, and unit testing; to facilitate low-level interaction with the GPU cluster scheduler; and to manage artifacts, checkpoints, and datasets. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

All datasets required to reproduce the results in this paper are publicly available through the official repositories of the referenced works Ruan et al. (2026); Yu et al. (2026). Code and scripts for reproducing reported results, including the training and evaluation of DRIFT, are included in the supplementary materials. The repository will be made publicly available upon acceptance.

REFERENCES

Abhinav Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, N. Teyssier, R. Ilango, S. Nagaraj, M. Dong, et al. Predicting cellular responses to perturbation across diverse contexts with state. *BioRxiv*, pp. 2025–06, 2025. doi: 10.1101/2025.06.26.661135.

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. ISSN 1548-7105. doi: 10.1038/s41592-025-02772-6. URL http://dx.doi.org/10.1038/s41592-025-02772-6.

Hananeh Aliee, Ferdinand Kapl, Duy Pham, Batuhan Cakir, Takahiro Jimba, James Cranley, Sarah A. Teichmann, Kerstin B. Meyer, Roser Vento-Tormo, and Fabian J. Theis. invae: Conditionally invariant representation learning for generating multivariate single-cell reference maps. *bioRxiv*, 2024. doi: 10.1101/2024.12.06.627196. URL http://dx.doi.org/10.1101/2024.12.06.627196.

Shaokun An, Jae-Won Cho, Kai Cao, Jiankang Xiong, Martin Hemberg, and Lin Wan. sccausalvi disentangles single-cell perturbation responses with causality-aware generative model. *bioRxiv*, 16(11):101443, 2025. doi: 10.1101/2025.02.02.636136. URL http://dx.doi.org/10.1101/2025.02.02.636136.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. In *Advances in Neural Information Processing Systems 36*, pp. 1–12. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-0001.

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.

Shan Chen, Brook B. Lake, and Kun Zhang. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature Biotechnology*, 37(12):1452–1457, 2019.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. CLUB: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning*, 2020.

Gerold Csendes, Gema Sanz, Kristóf Z. Szalay, and Bence Szalai. Benchmarking foundation cell models for post-perturbation rna-seq prediction. *BMC Genomics*, 26(1), 2025. ISSN 1471-2164. doi: 10.1186/s12864-025-11600-2. URL http://dx.doi.org/10.1186/s12864-025-11600-2.

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.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Neural Information Processing Systems*, volume 26, pp. 2292–2300, 2013.

Pinar Demetci, Ryan Santorella, Björn Sandstede, William Stafford Noble, and Ritambhara Singh. Scot: Single-cell multi-omics alignment with optimal transport. *Journal of Computational Biology*, 29(1):3–18, 2022. doi: 10.1089/cmb.2021.0446.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7): 1853–1866, 2016.

Yicheng Gao, Kejing Dong, Caihua Shan, Dongsheng Li, and Qi Liu. Causal disentanglement for single-cell representations and controllable counterfactual generation. *Nature communications*, 16(1):6775, 2025.

Quentin Garrido, Laurent Najman, and Yann Lecun. Self-supervised learning of split invariant equivariant representations. *arXiv preprint arXiv:2302.10283*, 2023.

Luke A Gilbert, Max A Horlbeck, Britt Adamson, Jacqueline E Villalta, Yuwen Chen, Evan H Whitehead, Carla Guimaraes, Barbara Panning, Hidde L Ploegh, Michael C Bassik, et al. Genome-scale crispr-mediated control of gene repression and activation. *Cell*, 159(3):647–661, 2014.

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.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in neural information processing systems*, 29, 2016.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999. doi: 10.1016/s0893-6080(98)00140-3.

Wenkang Jiang, Yuhang Liu, Yichao Cai, Erdun Gao, Jiayi Dong, Ehsan Abbasnejad, Lina Yao, and Javen Qinfeng Shi. What makes a representation good for single-cell perturbation prediction? *Neural Information Processing Systems*, 2026. doi: 10.48550/arXiv.2605.19343.

Martin Jinek, Krzysztof Chylinski, Ines Fonfara, Michael Hauer, Jennifer A Doudna, and Emmanuelle Charpentier. A programmable dual-rna–guided dna endonuclease in adaptive bacterial immunity. *science*, 337(6096):816–821, 2012.

Kenji Kamimoto, Blerta Stringa, Christy M Hoffmann, Kunal Jindal, Lilianna Solnica-Krezel, and Samantha A Morris. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614(7949):742–751, 2023. doi: 10.1038/s41586-022-05688-9.

Kasia Z. Kedzierska, Lorin Crawford, Ava P. Amini, and Alex X. Lu. Assessing the limits of zero-shot foundation models in single-cell biology. *bioRxiv*, 2023. doi: 10.1101/2023.10.16.561085. URL http://dx.doi.org/10.1101/2023.10.16.561085.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Dynamic conditional optimal transport through simulation-free flows. In *Advances in Neural Information Processing Systems 37*, NeurIPS 2024, pp. 93602–93642. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. doi: 10.52202/079017-2968. URL http://dx.doi.org/10.52202/079017-2968.

Lennart Kester and Alexander Van Oudenaarden. Single-cell transcriptomics meets lineage tracing. *Cell stem cell*, 23(2):166–179, 2018.

Sumeer Ahmad Khan, Alberto Maillo, Vincenzo Lagani, Robert Lehmann, Narsis A. Kiani, David Gomez-Cabrero, and Jesper Tegner. Reusability report: Learning the transcriptional grammar in single-cell rna-sequencing data using transformers. *Nature Machine Intelligence*, 5(12):1437–1446, 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00757-8. URL http://dx.doi.org/10.1038/s42256-023-00757-8.

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

Ilyes Khemakhem, Ricardo Monti, Diederik Kingma, and Aapo Hyvarinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *Advances in Neural Information Processing Systems*, 33:12768–12778, 2020b.

Dominik Klein, Théo Uscidda, Fabian Theis, and Marco Cuturi. Genot: Entropic (gromov) wasserstein flow matching with applications to single-cell genomics. In *Neural Information Processing Systems*, NeurIPS 2024, pp. 103897–103944. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/079017-3301. URL http://dx.doi.org/10.52202/079017-3301.

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

Sebastien Lachapelle, Pau Rodriguez, Yash Sharma, Katie E Everett, Rémi LE PRIOL, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In *First Conference on Causal Learning and Reasoning*, 2022. URL https://openreview.net/forum?id=dHsFFekd_-o.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

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

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

Romain Lopez, Nataša Tagasovska, Stephen Ra, Kyunghyn Cho, Jonathan K. Pritchard, and Aviv Regev. Learning causal representations of single cells via sparse mechanism shift modeling, 2022. URL https://arxiv.org/abs/2211.03553.

M. Lotfollahi, Anna Klimovskaia Susmelj, C. De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, R. Daza, B. Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):e11517, 2023. doi: 10.15252/msb.202211517.

Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019. doi: 10.1038/s41592-019-0494-8.

Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=-e4EXDWXnSn.

Marija Milacic, Deidre Beavers, Patrick Conley, Chuqiao Gong, Marc Gillespie, Johannes Griss, Robin Haw, Bijay Jassal, Lisa Matthews, Bruce May, et al. The reactome pathway knowledgebase 2024. *Nucleic acids research*, 52(D1):D672–D678, 2024.

Daniel Moyer, Shuyang Gao, Rob Brekelmans, Aram Galstyan, and Greg Ver Steeg. Invariant representations without adversarial training. *Advances in neural information processing systems*, 31, 2018.

Thomas M. Norman, Max A. Horlbeck, Joseph M. Replogle, Alex Y. Ge, Albert Xu, Marco Jost, Luke A. Gilbert, and Jonathan S. Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019. doi: 10.1126/science.aax4438.

Joseph M Replogle, Thomas M Norman, Albert Xu, Jeffrey A Hussmann, Jin Chen, J Zachery Cogan, Elliott J Meer, Jessica M Terry, Daniel P Riordan, Niranjan Srinivas, et al. Combinatorial single-cell crispr screens by direct guide rna capture and targeted sequencing. *Nature biotechnology*, 38(8):954–961, 2020.

Joseph M Replogle, Reuben A Saunders, Angela N Pogson, Jeffrey A Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, 42(6):927–935, 2023. doi: 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, et al. Virtual cell challenge: Toward a turing test for the virtual cell. *Cell*, 188(13):3370–3374, 2025.

Yanay Rosen, Yusuf Roohani, Ayush Agrawal, Leon Samotorcan, Stephen R. Quake, and Jure Leskovec. Universal cell embeddings: A foundation model for cell biology. *bioRxiv*, 2026. doi: 10.1101/2023.11.28.568918. URL http://dx.doi.org/10.1101/2023.11.28.568918.

Jiafa Ruan, Ruijie Quan, Xu Liyang, Zongxin Yang, and Yi Yang. Beyond independent genes: Learning module-inductive representations for single-cell gene perturbation prediction. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=onuLk72nRw.

Geoffrey Schiebinger, Jun Shu, Maciej Tabaka, Brian Cleary, V. Subramanian, A. Solomon, S. Liu, S. Lin, P. Berube, L. Lee, J. Chen, J. Brumbaugh, P. Rigollet, K. Hochedlinger, R. Jaenisch, A. Regev, and E. S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22, 2019.

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

Peter Sorrenson, C. Rother, and U. Köthe. Disentanglement by nonlinear ICA with general incompressible-flow networks (GIN). In *International Conference on Learning Representations*. Cornell University, 2020. doi: 10.48550/arxiv.2001.04872.

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

Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, et al. The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic acids research*, 51(D1):D638–D646, 2023.

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

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A Teichmann, Mor Nitzan, and Maria Brbić. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, pp. 1–10, 2025.

Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

Ethan Weinberger, Chris Lin, and Su-In Lee. Isolating salient variations of interest in single-cell data with contrastivevi. *Nature methods*, 20(9):1336–1345, 2023.

Robin Winter, Marco Bertolini, Tuan Le, Frank Noe, and Djork-Arné Clevert. Unsupervised learning of group invariant and equivariant representations. *Advances in Neural Information Processing Systems*, 35:31942–31956, 2022.

Yulun Wu, Robert A. Barton, Zichen Wang, V. Ioannidis, Carlo De Donno, Layne Price, L. Voloch, and G. Karypis. Predicting cellular responses with variational causal inference and refined relational information. In *International Conference on Learning Representations*, 2023. doi: 10.48550/arXiv.2210.00116.

Hanwen Xing and Christopher Yau. Gperturb: Gaussian process modelling of single-cell perturbation data. *bioRxiv*, 16(1), 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-61165-7. URL http://dx.doi.org/10.1038/s41467-025-61165-7.

Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature machine intelligence*, 4(10):852–866, 2022.

Xiaodong Yang, Guole Liu, Guihai Feng, Dechao Bu, Pengfei Wang, Jie Jiang, Shubai Chen, Qinmeng Yang, Hefan Miao, Yiyang Zhang, et al. Genecompass: deciphering universal gene regulatory mechanisms with a knowledge-informed cross-species foundation model. *Cell Research*, 34 (12):830–845, 2024.

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.

Hengshi Yu, Weizhou Qian, Yuxuan Song, and Joshua D Welch. Perturbnet predicts single-cell responses to unseen chemical and genetic perturbations. *Molecular Systems Biology*, 21(8), 2025. ISSN 1744-4292. doi: 10.1038/s44320-025-00131-3. URL http://dx.doi.org/10.1038/s44320-025-00131-3.

Yuansong Zeng, Jiancong Xie, Ningyuan Shangguan, Zhuoyi Wei, Wenbing Li, Yun Su, Shuangyu Yang, Chengyang Zhang, Jinbo Zhang, Nan Fang, et al. Cellfm: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells. *Nature Communications*, 16(1):4679, 2025.

Jiaqi Zhang, Kristjan Greenewald, Chandler Squires, Akash Srivastava, Karthikeyan Shanmugam, and Caroline Uhler. Identifiability guarantees for causal disentanglement from soft interventions. *Advances in Neural Information Processing Systems*, 36:50254–50292, 2023.

Ziqi Zhang, Xinye Zhao, Mehak Bindra, Peng Qiu, and Xiuwei Zhang. scdisinfact: disentangled learning for integration and prediction of multi-batch multi-condition single-cell rna-sequencing data. *Nature Communications*, 15(1):912, 2024.

A METRIC DEFINITIONS

We follow the definitions and released implementation of Ruan et al. (2026). For a perturbation $p \in \mathcal{P}$, $\mathbf{x}_p, \hat{\mathbf{x}}_p \in \mathbb{R}^G$ are the observed and predicted mean expression profiles, $\bar{\mathbf{x}}_{\text{ctrl}}$ is the mean of the training control cells, and $\mathcal{D}_p$ is the set of the 20 genes with the largest $|x_{pg} - \bar{x}_{\text{ctrl},g}|$.

Point-prediction accuracy.

$$\text{MSE} = \frac{1}{|\mathcal{P}|G} \sum_p \|\mathbf{x}_p - \hat{\mathbf{x}}_p\|_2^2, \qquad \text{MAE} = \frac{1}{|\mathcal{P}|G} \sum_p \|\mathbf{x}_p - \hat{\mathbf{x}}_p\|_1, \qquad L_2 = \frac{1}{|\mathcal{P}|} \sum_p \|\mathbf{x}_p - \hat{\mathbf{x}}_p\|_2. \tag{10}$$

Biological response fidelity.

$$\rho\Delta = \frac{1}{|\mathcal{P}|} \sum_p \text{Corr}\left(\hat{\mathbf{x}}_p - \bar{\mathbf{x}}_{\text{ctrl}},\ \mathbf{x}_p - \bar{\mathbf{x}}_{\text{ctrl}}\right), \tag{11}$$

$$\text{ACC}\Delta = \frac{1}{|\mathcal{P}|G} \sum_p \sum_{g=1}^{G} \mathbb{I}\left[\text{sign}(\hat{x}_{pg} - \bar{x}_{\text{ctrl},g}) = \text{sign}(x_{pg} - \bar{x}_{\text{ctrl},g})\right], \tag{12}$$

where Corr is the Pearson correlation; $\rho\Delta^{\text{D}}$ and $\text{ACC}\Delta^{\text{D}}$ restrict both to $\mathcal{D}_p$. DES is the Spearman correlation of log fold changes over the genes identified as significant by a Wilcoxon rank-sum test between perturbed and control cells ($p < 0.05$):

$$\text{DES} = \frac{1}{|\mathcal{P}|} \sum_p \text{Spearman}\left(\log_2 \frac{\hat{\mathbf{x}}_{p,\text{sig}}}{\bar{\mathbf{x}}_{\text{ctrl,sig}}},\ \log_2 \frac{\mathbf{x}_{p,\text{sig}}}{\bar{\mathbf{x}}_{\text{ctrl,sig}}}\right). \tag{13}$$

Discriminative power. PDS ranks the observed profiles of all perturbations in $\mathcal{P}$ by their L_1 distance to $\hat{\mathbf{x}}_p$, excluding the genes targeted by p; with rank_p the position of $\mathbf{x}_p$,

$$\text{PDS} = \frac{1}{|\mathcal{P}|} \sum_p \left(1 - \frac{\text{rank}_p - 1}{|\mathcal{P}| - 1}\right). \tag{14}$$

Distributional metrics. Following Ruan et al. (2026), we assess distributional fidelity using three metrics that compare the predicted and observed cell *populations* for each perturbation, rather than their means, on cells projected onto 50 principal components. **E-Dist** is the energy distance between the two populations: zero exactly when they coincide, and sensitive to their spread as well as their location. **Wasserstein** is the entropic optimal transport cost ($\varepsilon = 1$, uniform weights) between them, the average distance cells must travel to turn one population into the other. **Discr. Cos** is the cosine between the true and predicted mean shift from control, and so measures only the direction of the response, not its magnitude. All three are averaged over $\mathcal{P}$.

ComboSciPlex. We evaluate ComboSciPlex using `cell-eval`, following Yu et al. (2026).

B IMPLEMENTATION

Datasets. Norman (Norman et al., 2019) uses CRISPRa to activate 105 genes and 131 gene pairs in K562 cells; Replogle RPE1 (Replogle et al., 2022) uses CRISPRi to silence essential genes. We use the preprocessing and splits of Ruan et al. (2026): $\ln(\text{CPM}+1)$ on highly variable genes (2051 for Norman; 3754 for Replogle), and follow the same train/validation/test splits as Ruan et al. (2026), ensuring directly comparable evaluation. Baseline results on Norman and Replogle are reported from Ruan et al. (2026), on ComboSciPlex they are reported from Yu et al. (2026). Norman's *additive* split trains on 167 conditions and holds out 32 pairs whose constituent genes are observed singly, testing combinatorial generalization. Its *holdout* split trains on 190 conditions and holds out 26 that include genes absent from training, alone and in pairs, testing zero-shot generalization. Replogle holds out 384 unseen-gene single-perturbation conditions. ComboSciPlex (Srivatsan et al., 2020) profiles A549 cells under 17 compounds across 31 conditions (24 train, 7 test); following Yu et al. (2026), we use the same train/validation/test splits, train on 5000 highly variable genes, evaluate on 1000 test-selected genes, and generate 128 cells per condition.

Algorithm 1: Inference: predicting the response to a perturbation $u^\star$

1: **Input:** target perturbation $u^\star$; observed control cells; number of cells N.
2: **Output:** predicted perturbed cells $\{\hat{x}_i\}_{i=1}^{N}$.
3: $e_{u^\star} \leftarrow$ perturbation encoder applied to $u^\star$
4: **for** $i = 1$ **to** N
5: draw a control cell x and encode it, $(z_{\mathrm{nr}}, z_r) \leftarrow q_\phi(x, u_{\mathrm{ctrl}}, c)$
6: integrate $\dot{z} = v_\theta(z, t \mid z_{\mathrm{nr}}, e_{u^\star})$ from $z_0 = z_r$ to $t = 1$ (50 Euler steps), giving $z'_r = z_1$
7: $\hat{x}_i \leftarrow p_\theta([z_{\mathrm{nr}}, z'_r])$, with z_{nr} copied from the control cell
8: **end for**

Stage 1: the disentangling VAE. The encoder and decoder are MLPs on $\ln(\mathrm{CPM}{+}1)$ expression. The encoder receives the perturbation u and the covariates c and emits the invariant block z_{nr} and the responsive block z_r; u also conditions the responsive prior and c the invariant prior. The decoder has a single mean-squared-error head, used for training and for every evaluation. c comprises the sequencing run together with per-cell sequencing depth, genes detected, and mitochondrial and ribosomal content; we verify that these are close to perturbation-independent. On ComboSciPlex c is empty: the screen is arrayed, so every well receives a single drug condition and the well label is not independent of u. Table 5 gives all settings.

The perturbation encoder is additive with an interaction term: $s_0 = \varphi(g_1) + \varphi(g_2)$, $s = s_0 + \psi([\, s_0,\ \varphi(g_1) \odot \varphi(g_2)\,])$ and $e_u = \rho(s)$, with φ an MLP. Both the sum and the elementwise product are symmetric in (g_1, g_2), so the code does not depend on gene order. The interaction branch ψ is applied only when both genes are present, and is zero-initialized, so composition begins exactly additive and becomes interactive only if the data pay for it; a control cell returns a learned NULL code and a gene with no feature row is routed to a learned UNKNOWN.

Perturbation features. For a genetic perturbation, each gene is represented by the concatenation of three blocks: ESM2 protein embeddings (Lin et al., 2023) reduced by PCA fitted over the full gene dictionary; a neighborhood in the STRING network of protein–protein associations (Szklarczyk et al., 2023), taken over the whole interactome; and co-membership in Reactome biological pathways (Milacic et al., 2024). We use these features following the official implementation of Ruan et al. (2026). The two network blocks describe a gene by what it interacts with and what processes it belongs to, which the protein sequence alone does not carry. Genes absent from a source receive an exactly zero block in that source. Table 9 measures what the two network blocks contribute over the protein sequence alone. For the chemical perturbation, each compound is represented by a Morgan fingerprint (Rogers & Hahn, 2010) of radius 4 and 1024 bits, the representation and settings used for molecule perturbations by Klein et al. (2025), reduced by PCA to 16 dimensions.

Objective. Both KL terms are warmed up linearly, and the CLUB penalty on $I(z_{\mathrm{nr}}; u)$ follows the same schedule, so it does not constrain an untrained latent. The CLUB critic predicts a projection of the condition code e_u from z_{nr} and is trained with its own optimizer at a learning rate of 10^{-3}.

Stage 2: the conditional flow. The velocity field is a DiT in which the two perturbed genes enter as separate tokens, and the stage-1 perturbation encoder is inherited. Training pairs come from an entropic optimal transport plan (Sinkhorn) over blocks of cells drawn from several conditions, with the cost of (9). Generation integrates the probability-flow ODE with Euler steps, using the EMA weights.

C CONDITIONING REGULARIZATION

Response supervision. A linear head $h : \mathbb{R}^{d_{\mathrm{pert}}} \to \mathbb{R}^G$ predicts the condition's mean training response $\bar{\Delta}_c = \bar{x}_c - \bar{x}_{\mathrm{ctrl}}$,

$$\mathcal{L}_{\mathrm{aux}} = \frac{1}{|\mathcal{C}_{\mathrm{tr}}|} \sum_{c \in \mathcal{C}_{\mathrm{tr}}} \left\| h(e_u^{(c)}) - \bar{\Delta}_c \right\|_2^2, \tag{15}$$

Table 5: Configuration of DRIFT, read from the resolved configuration of the checkpoints that produced the tables. A single centered entry means the setting is shared by the three datasets.

Category	Hyperparameter	Norman	Replogle	ComboSciPlex
Model	Invariant block d_{nr}		64	
	Responsive block d_r		192	
	Encoder and decoder width		1024 (2 layers)	
	Condition code d_{pert}		128	
	Perturbation encoder width		256	
	Perturbation features	3×128	3×128	16
	Velocity field (DiT) blocks / tokens		6 / 16	
	Conditioning width		256	
	Time embedding frequencies		128	
Stage 1	Epochs		120	
	Batch size		256	
	Optimizer / learning rate		Adam / 10^{-4}	
	β_r, β_{nr}		0.5, 4	
	KL and MI warm-up		20 epochs	
	λ_{CLUB} / critic dimension / inner steps		5 / 32 / 5	
Stage 2	Steps		60,000	
	Batch size		512	
	Optimizer / learning rate		Adam / 10^{-4}	
	EMA decay		0.999	
	OT block: cells $\times$ conditions		512×16	
	Sinkhorn ε; α_{nr}, α_r		0.5; 1, 1	
	Euler steps at inference		50	

which ties e_u to the cell responses it must eventually explain. Without it φ is shaped only by what reaches it through the KL term of the responsive prior, a weak and indirect signal. The head is kept linear so that it cannot itself absorb the structure we are inducing in e_u, and is not used after training.

Relative isometry. Let $D^e_{ij} = \|e_u^{(i)} - e_u^{(j)}\|$ and $D^{\Delta}_{ij} = \|\bar{\Delta}_i - \bar{\Delta}_j\|$ be the pairwise distances between codes and between mean training responses. We penalize the departure of their correlation from one,

$$\mathcal{L}_{\mathrm{iso}} = 1 - \mathrm{corr}\big(\mathrm{vec}\, D^e,\ \mathrm{vec}\, D^{\Delta}\big). \tag{16}$$

This gives perturbations with similar responses similar codes. Matching relative distances instead of absolute ones constrains only the ordering and spacing of conditions, leaving the scale of e_u free to adapt to the prior.

Noise augmentation. During training, Gaussian noise is added to the condition embedding. This prevents the representation from collapsing into a lookup table over training conditions and ensures that neighborhoods around learned codes remain valid, so unseen genes near seen ones can still be decoded sensibly.

D What each latent block encodes

Figure 3 shows that the responsive block groups cells by perturbation, whereas the invariant block interleaves them and captures covariates in c (Figure 4). Table 6 confirms this separation across vocabularies, from the 20 strongest perturbations to all conditions: z_r predicts perturbation identity accurately, while z_{nr} remains far below z_r. The same pattern holds for mutual information: z_r captures most of u, whereas $I(z_{\mathrm{nr}}, u)$ is very low. This ordering is consistent across linear and MLP probes, as well as variational and MINE estimators.

E Generated populations

Figure 5 compares generated and observed cells for the five held-out conditions of the Norman additive split with the largest shift from control. The generated densities match the observed ones in

Table 6: Decoding *which* perturbation a cell received from each latent block. Classes are balanced, so chance is $1/K$. MINE Belghazi et al. (2018) is a mutual-information lower bound in nats.

Dataset	Conditions	chance	probe accuracy				MINE (nats)	
			linear		MLP			
			z_{nr}	z_r	z_{nr}	z_r	z_{nr}	z_r
Norman additive	top 20	0.0500	0.157	**0.981**	0.171	**0.991**	0.14	**3.35**
	top 50	0.0200	0.100	**0.994**	0.093	**0.998**	0.15	**4.00**
	top 100	0.0100	0.053	**0.980**	0.056	**0.981**	0.17	**3.96**
	all	0.0060	0.042	**0.938**	0.037	**0.962**	0.11	**3.68**
Norman holdout	top 20	0.0500	0.106	**0.991**	0.102	**0.991**	0.12	**3.38**
	top 50	0.0200	0.061	**0.981**	0.048	**0.991**	0.14	**3.72**
	top 100	0.0100	0.035	**0.963**	0.028	**0.981**	0.17	**3.95**
	all	0.0053	0.025	**0.947**	0.022	**0.964**	0.10	**3.79**
Replogle RPE1	top 20	0.0500	0.155	**0.667**	0.131	**0.667**	0.08	**1.55**
	top 50	0.0200	0.160	**0.625**	0.150	**0.600**	0.29	**1.60**
	top 100	0.0100	0.070	**0.605**	0.070	**0.620**	0.15	**1.82**
	all	0.0010	0.017	**0.340**	0.015	**0.296**	0.02	**0.64**

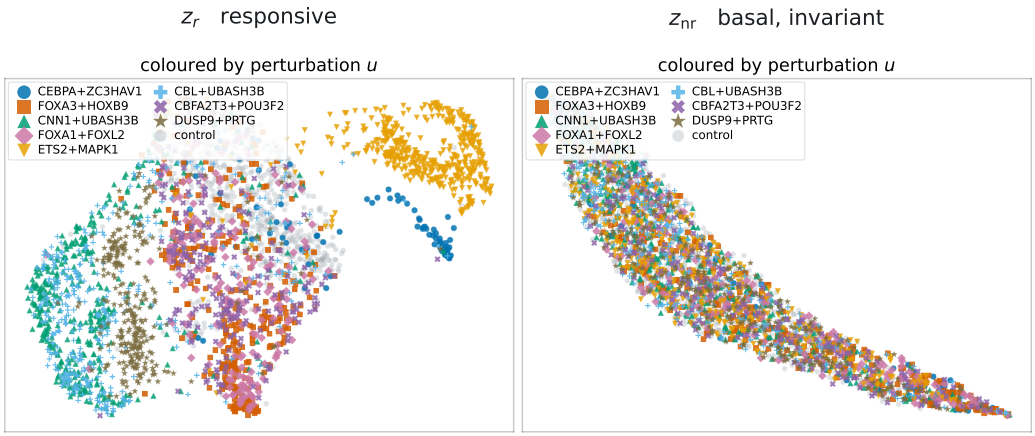


Figure 3: Norman additive: UMAPs of the responsive block z_r and invariant block z_{nr}, colored by perturbation.

location and shape, including the second mode of CNN1+UBASH3B and ETS2+MAPK1. Table 7 extends the comparison to all held-out conditions: DRIFT improves on every baseline on the three distributional metrics, raising Discr. Cos from 0.726 to 0.912 and reducing E-Dist by 22% relative to scBIG.

F Ablations

Table 8 removes one component at a time, holding all others and the training budget fixed. *Without disentanglement* replaces the split latent representation with a single block trained as a standard variational autoencoder, without a conditional prior or invariance penalty. The conditional flow consequently transports the full latent representation, whereas DRIFT transports only its responsive components, isolating the value of latent disentanglement. *Without invariance penalty* retains the split latent representation but sets $\lambda_{\mathrm{CLUB}} = 0$. *Without conditioning regularization* removes the mechanism described in Appendix C that structures the condition code e_u. Because the metrics have different scales, Table 10 reports each component's contribution as a fraction of DRIFT's margin over the Linear baseline.

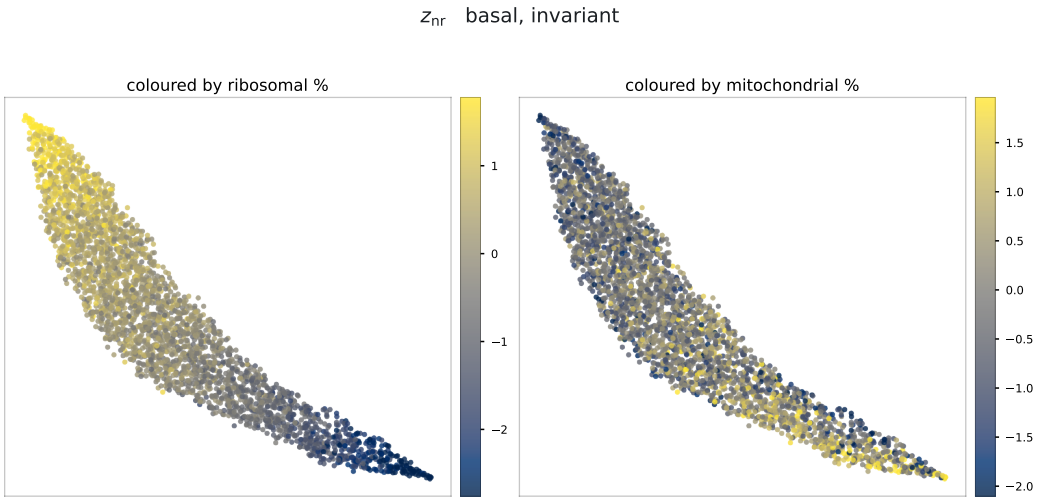


Figure 4: The invariant block of Figure 3, colored by the numeric covariates in c.

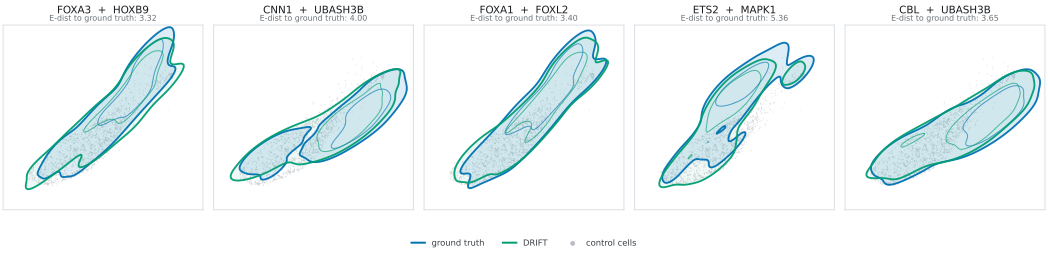


Figure 5: UMAP density visualizations on representative Norman additive perturbations.

Disentanglement. The disentanglement is the dominant component: it improves all 20 entries and yields the largest mean contribution on every dataset, ranging from 11% on the additive split to 35% on holdout double perturbations (21% overall).

Invariance. The invariance penalty provides an additional benefit beyond the split latent. It contributes positively on the majority of metrics. Its mean contribution is positive on every dataset (6% overall), with its largest effect on direction of change (ACCΔ, 17% overall).

Conditioning regularization. Conditioning regularization improves generalization to unseen perturbations by structuring the condition representation. Its effect is largest on Replogle (6.3%), where every test gene is unseen during training: it raises PDS from 0.573 to 0.604, accounting for 43% of DRIFT's margin over Linear. This gain comes with a small 5% decrease in $\rho\Delta$, but DRIFT still substantially outperforms scBIG. The effect is smallest on the additive split, where test pairs combine genes observed individually during training.

Perturbation features. Table 9 removes the STRING and Reactome features from the perturbation context, retaining ESM2 alone. These additional features are most valuable when the model must generalize to unseen perturbed genes. On Replogle, they improve all nine metrics. By contrast, their effect is limited on the Norman splits, where test conditions recombine genes observed individually during training and most metrics are indistinguishable. Notably, DRIFT with ESM2 alone outperforms Ruan et al. (2026) on every split. Thus, STRING and Reactome features strengthen generalization to unseen genes but do not account for DRIFT's advantage over scBIG, which also uses these features. Together, these ablations identify disentanglement as the primary contributor, with invariance and conditioning regularization providing additional gains.

Table 7: Distributional metrics on Norman additive (Appendix A). Baselines are reported by Ruan et al. (2026); DRIFT is the mean and standard deviation over five seeds. **Bold**: best; underline: second best.

Type	Method	Discr. Cos↑	E-Dist↓	Wasserstein↓
Sta.	Control	-0.3009	13.5125	16.5615
	Linear	0.3967	11.1376	15.3741
Found.	scFoundation	0.6218	1.3710	11.2452
	GeneCompass	0.5623	1.3059	11.1429
Gra.	GEARS	0.5871	1.2377	11.0829
Gen.	CellFlow	0.6397	1.1979	10.7579
	scBIG	0.7264	0.8383	9.7842
	DRIFT (ours)	**0.9123**$\pm$0.0039	**0.6516**$\pm$0.0298	**9.4115**$\pm$0.0548

Table 8: Component ablations. Each variant removes only the named component. Rank is the mean rank across five metrics (1 = best), averaged over 5 seeds.

Configuration	$\rho\Delta\uparrow$	$\rho\Delta^{D}\uparrow$	ACC$\Delta\uparrow$	DES↑	PDS↑	rank
Norman additive						
DRIFT	0.8871	0.8914	0.9222	0.9017	0.9167	**1.40**
w/o disentanglement	0.8588	0.8236	0.9220	0.8753	0.9042	3.40
w/o conditioning regularization	0.8974	0.8926	0.9183	0.8965	0.9051	1.80
w/o invariance penalty	0.8706	0.8778	0.9146	0.8964	0.8992	3.40
Norman holdout (single)						
DRIFT	0.6690	0.7046	0.7890	0.6904	0.7596	**1.40**
w/o disentanglement	0.6378	0.6600	0.7853	0.6598	0.7538	3.40
w/o conditioning regularization	0.6662	0.7093	0.7852	0.6811	0.7522	2.40
w/o invariance penalty	0.6585	0.6970	0.7801	0.6751	0.7606	2.80
Norman holdout (double)						
DRIFT	0.8562	0.9319	0.8724	0.8734	0.8075	**1.60**
w/o disentanglement	0.7943	0.8540	0.8571	0.8141	0.7915	4.00
w/o conditioning regularization	0.8600	0.9280	0.8709	0.8779	0.7973	2.00
w/o invariance penalty	0.8534	0.9426	0.8654	0.8657	0.8062	2.40
Replogle RPE1						
DRIFT	0.5085	0.6160	0.6537	0.7021	0.6044	**1.60**
w/o disentanglement	0.4804	0.5942	0.6452	0.6723	0.5739	3.80
w/o conditioning regularization	0.5194	0.6254	0.6569	0.6976	0.5729	1.80
w/o invariance penalty	0.5055	0.6159	0.6528	0.6931	0.5978	2.80

G FULL RESULTS WITH STANDARD DEVIATIONS

Table 9: Contribution of genetic perturbation features. DRIFT represents each perturbed gene using ESM2 ⊕ STRING ⊕ Reactome; *ESM2 only* removes STRING and Reactome while leaving all other components unchanged. Ruan et al. (2026) already integrates all three sources. **Bold**: best; underline: second best.

Split	Method	$\rho\Delta\uparrow$	$\rho\Delta^{D}\uparrow$	ACC$\Delta\uparrow$	ACC$\Delta^{D}\uparrow$	DES$\uparrow$	PDS$\uparrow$	L2$\downarrow$	MSE$\downarrow$	MAE$\downarrow$
Norman additive	scBIG	0.8496	0.8230	0.9197	0.9906	0.8593	0.8548	3.92	0.0091	0.0689
	DRIFT, ESM2 only	**0.8878**	**0.8938**	0.9203	**0.9989**	**0.9020**	0.9045	3.22	0.0062	0.0569
	DRIFT	0.8871	0.8914	**0.9222**	0.9977	0.9017	**0.9167**	**3.12**	**0.0057**	**0.0549**
Replogle RPE1	scBIG	0.4875	0.5925	0.6471	0.8089	0.5145	0.5520	6.12	0.0118	0.0676
	DRIFT, ESM2 only	0.4919	0.6032	0.6465	0.8124	0.6865	0.5696	6.04	0.0115	0.0670
	DRIFT	**0.5085**	**0.6160**	**0.6537**	**0.8149**	**0.7021**	**0.6044**	**5.68**	**0.0101**	**0.0633**

Table 10: Component contributions, expressed as a percentage of DRIFT's margin over the Linear baseline: (DRIFT − without component)/(DRIFT − Linear) for each dataset and metric. Green values indicate improvement; grey values lie within standard error across seeds; red values are negative.

Component	$\rho\Delta$	$\rho\Delta^{D}$	ACCΔ	DES	PDS	mean
Norman additive						
Disentanglement	+10.6	+36.9	+0.2	+4.5	+3.3	+11.1
Conditioning regularization	−3.9	−0.7	+3.2	+0.9	+3.0	+0.5
Invariance penalty	+6.2	+7.4	+6.3	+0.9	+4.6	+5.1
Norman holdout (single)						
Disentanglement	+40.7	+43.5	+19.0	+13.0	+2.3	+23.7
Conditioning regularization	+3.7	−4.6	+19.5	+3.9	+3.0	+5.1
Invariance penalty	+13.7	+7.4	+45.6	+6.5	−0.4	+14.6
Norman holdout (double)						
Disentanglement	+44.8	+74.8	+31.6	+19.1	+5.2	+35.1
Conditioning regularization	−2.7	+3.7	+3.1	−1.5	+3.3	+1.2
Invariance penalty	+2.0	−10.3	+14.5	+2.5	+0.4	+1.8
Replogle RPE1						
Disentanglement	+11.9	+8.6	+11.8	+6.1	+42.0	+16.1
Conditioning regularization	−4.6	−3.7	−4.4	+0.9	+43.4	+6.3
Invariance penalty	+1.3	0.0	+1.2	+1.8	+9.1	+2.7
All datasets						
Disentanglement	+27.0	+40.9	+15.6	+10.7	+13.2	+21.5
Conditioning regularization	−1.9	−1.3	+5.3	+1.1	+13.2	+3.3
Invariance penalty	+5.8	+1.1	+16.9	+2.9	+3.4	+6.0

Table 11: Standard deviation over five seeds, Norman additive split.

Method	$\rho\Delta$	$\rho\Delta^{D}$	ACCΔ	ACCΔ^{D}	DES	PDS	L2	MSE	MAE
GEARS	0.0055	0.0151	0.0031	0.0080	0.0081	0.0048	0.0649	0.0006	0.0014
scGPT	0.0006	0.0045	0.0010	0.0010	0.0084	0.0026	0.0197	0.0002	0.0004
scFoundation	0.0009	0.0031	0.0004	0.0036	0.0067	0.0029	0.0044	0.0001	0.0002
CellFlow	0.0035	0.0146	0.0013	0.0213	0.0033	0.0074	0.0636	0.0004	0.0014
scBIG	0.0026	0.0038	0.0023	0.0029	0.0015	0.0096	0.0411	0.0002	0.0009
DRIFT (ours)	0.0034	0.0045	0.0031	0.0019	0.0028	0.0101	0.0558	0.0002	0.0010

Table 12: Standard deviation over five seeds, Norman holdout split.

Method	$\rho\Delta$	$\rho\Delta^{D}$	ACCΔ	ACCΔ^{D}	DES	PDS	L2	MSE	MAE
scGPT	0.0008	0.0027	0.0003	0.0031	0.0155	0.0006	0.0039	0.0001	0.0001
scFoundation	0.0072	0.0122	0.0042	0.0091	0.0158	0.0023	0.0227	0.0000	0.0005
CellFlow	0.0026	0.0098	0.0013	0.0038	0.0042	0.0092	0.0053	0.0000	0.0001
scBIG	0.0016	0.0036	0.0021	0.0014	0.0042	0.0026	0.0105	0.0000	0.0002
DRIFT (ours)	0.0048	0.0041	0.0023	0.0046	0.0061	0.0040	0.0238	0.0001	0.0004

Table 13: Replogle RPE1, mean and standard deviation over 5 seeds of DRIFT.

Type	Method	$\rho\Delta\uparrow$	$\rho\Delta^{D}\uparrow$	ACC$\Delta\uparrow$	ACC$\Delta^{D}\uparrow$	DES$\uparrow$	PDS$\uparrow$	L2$\downarrow$	MSE$\downarrow$	MAE$\downarrow$
Gen.	**DRIFT (ours)**	0.5085$\pm$0.0074	0.6160$\pm$0.0058	0.6537$\pm$0.0013	0.8149$\pm$0.0015	0.7021$\pm$0.0072	0.6044$\pm$0.0067	5.68$\pm$0.03	0.0101$\pm$0.0001	0.0633$\pm$0.0002

Table 14: ComboSciPlex, mean and standard deviation over five end-to-end seeds of DRIFT.

Method	$\rho\Delta\uparrow$	DE-Spearman $\rho\uparrow$	DS$\uparrow$	L2$\downarrow$	MSE$\downarrow$	MAE$\downarrow$
DRIFT (ours)	0.9467$\pm$0.0057	0.8901$\pm$0.0057	0.8816$\pm$0.0171	1.2804$\pm$0.0536	0.0019$\pm$0.0002	0.0150$\pm$0.0006