SDFM: CROSS-AWARE KNOWLEDGE-CONDITIONED FLOW MATCHING FOR SINGLE-CELL GENETIC PERTURBATION PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Genetic perturbation screens reveal how interventions reshape cell populations, but their limited coverage leaves much of the intervention space unexplored. Predicting unmeasured responses requires generalization across perturbations while preserving cellular diversity. Flow matching provides a natural framework for modeling such population-level transformations, yet its local velocity objective only indirectly supervises the responses induced by complete generative trajectories. We propose SDFM, a cross-aware knowledge-conditioned flow matching framework for perturbation response prediction. Knowledge graph representations from STRING and Gene Ontology condition the velocity field through a cross-aware perturbation representation, allowing related perturbations and combinatorial contexts to influence learned transport dynamics. Mini-batch optimal transport establishes population-level correspondences between unpaired control and perturbed cells, while full-rollout supervision connects local velocity learning with predicted endpoints and population expression shifts. Experiments on Adamson, HepG2, and Norman show that SDFM achieves the highest mean all-gene expression change Pearson correlation among evaluated methods across the three single-gene benchmarks and improves combinatorial response prediction. It also reduces mean sliced Wasserstein and energy distances relative to a uniform mean-shift baseline, supporting its ability to model both average transcriptional effects and heterogeneous population-level responses.

1 INTRODUCTION

Understanding how cells respond to genetic perturbations is fundamental to connecting gene function, regulatory programs, and disease mechanisms. Perturb-seq and related technologies now measure transcriptomic responses to thousands of interventions at single-cell resolution (Dixit et al., 2016; Replogle et al., 2022). Yet the space of possible interventions grows combinatorially with target genes, combinations, and cellular contexts, leaving most of it experimentally inaccessible. Learning transferable principles from existing screens to predict unseen responses is therefore central to functional genomics.

Computational approaches have progressively moved beyond direct expression regression toward structured models of cellular state transitions. scGen and CPA learn perturbation-induced transformations in latent spaces (Lotfollahi et al., 2019; 2023); GEARS incorporates gene-relation graphs to generalize toward unseen interventions (Roohani et al., 2024); and large-scale models such as scGPT and scFoundation learn broadly transferable representations of genes and cellular states (Cui et al., 2024; Hao et al., 2024). More recent approaches, including PerturbNet, State, and TxPert, highlight the importance of cellular heterogeneity, perturbation structure, and external biological knowledge (Yu et al., 2025; Adduri et al., 2026; Wenkel et al., 2026). Benchmarking nevertheless suggests that increasing model capacity alone does not guarantee better prediction, underscoring the need for inductive structure tailored to the biological problem (Wu et al., 2025).

A fundamental feature is that single-cell perturbation data are not naturally paired. Because measurement destroys the cell, experiments typically provide control and perturbed populations sampled independently, rather than before-and-after measurements of the same cells. Observations are

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

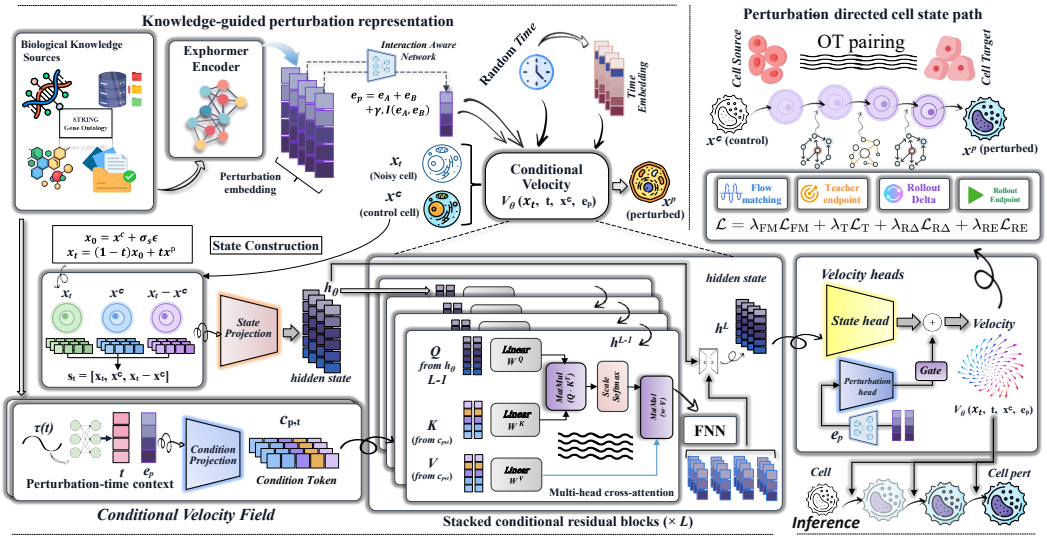

Figure 1: SDFM architecture. Biological graphs are encoded into perturbation representations; OT pairs unpaired control and perturbed cells. Perturbation–time cross-attention conditions the velocity field, whose state and perturbation heads are trained with flow, teacher, rollout, and endpoint losses.

therefore naturally viewed as samples from two distributions. Optimal transport offers a principled language for unpaired populations, and CellOT and CINEMA-OT demonstrate the value of population-level transport for perturbation analysis (Bunne et al., 2023; Dong et al., 2023). Thus, a genetic perturbation is not merely a fixed expression offset but a transformation transporting a control distribution toward a perturbation-specific distribution.

This distributional view aligns with continuous generative modeling. Flow Matching provides an efficient framework for learning time-dependent vector fields that transport one distribution into another (Lipman et al., 2023). CellFlow, scDFM, and CFM-GP have begun to bring conditional Flow Matching into cellular perturbation modeling, enabling generation of complete perturbed populations rather than only point estimates (Klein et al., 2025; Yu et al., 2026). Once response is viewed as a dynamical transport process, however, a deeper question emerges: the model must specify not only *where* cells should move, but also *which biological intervention defines the dynamics*, and whether locally learned velocities ultimately realize the correct population-level response.

Motivated by this view, we formulate genetic perturbation prediction as a *knowledge-conditioned population transport problem*. We first derive structured perturbation representations from biological knowledge graphs (Wenkel et al., 2026) and use them to condition the entire continuous velocity field, rather than treating perturbation information solely as an auxiliary feature for static expression prediction (Lotfollahi et al., 2023). Concretely, multi-head cross-attention over perturbation–time tokens (Vaswani et al., 2017) injects this representation into every velocity residual block, allowing different heads to capture complementary perturbation and temporal factors and coupling perturbation semantics with state-dependent dynamics throughout the transport trajectory. We then use optimal transport to establish a data-driven geometry between unpaired control and perturbed populations (Bunne et al., 2023; Dong et al., 2023; Peyré & Cuturi, 2019), which defines the trajectories on which Flow Matching is learned (Lipman et al., 2023; Tong et al., 2024). The resulting generative process is illustrated in Figure 1 and can be summarized as

$$p \longrightarrow e_p \xrightarrow{\text{Cross-Attn}(e_p,\, x_t)} v_\theta(x, t \mid e_p) \longrightarrow \Phi_p(\mu_c),$$

where biological knowledge determines the perturbation representation e_p, multi-head perturbation–time cross-attention injects e_p into the velocity field through interaction with the current cellular state x_t, the conditional vector field v_θ describes perturbation-specific cellular dynamics, and Φ_p transports the control population μ_c toward the predicted perturbed population. We further introduce trajectory-consistent population learning, which aligns local velocity learning with the final generated population response through multi-scale supervision, ensuring that the training objective remains consistent with the response produced at inference (Klein et al., 2025; Yu et al., 2026).

Our contributions are fourfold. **Knowledge-conditioned perturbation dynamics.** We use perturbation representations derived from multiple biological knowledge sources to condition the continuous velocity field itself, turning biological priors from static representations into determinants of perturbation-specific dynamics (Wenkel et al., 2026; Roohani et al., 2024). **Cross-aware perturbation representation.** We derive perturbation representations from biological knowledge graphs. These representations are tokenized and integrated into multi-head perturbation–time cross-attention to condition the velocity field, allowing biological priors and non-additive interactions to jointly shape perturbation-specific dynamics (Vaswani et al., 2017). For multi-gene perturbations, an interaction-aware composer adds learned pairwise residual information to the additive gene semantics of combinatorial perturbations (Zaheer et al., 2017). **A unified framework for biological population transport.** We integrate biological priors, optimal-transport geometry, and continuous generative dynamics within a single perturbation model (Peyré & Cuturi, 2019; Lipman et al., 2023). Across multiple single-cell perturbation settings, experiments demonstrate improvements in response prediction and perturbation retrieval, alongside recovery of heterogeneous cellular responses. **Trajectory-consistent population learning.** We introduce a multi-scale training objective that connects local Flow Matching velocities with implied endpoints, population-level transcriptional shifts, and complete ODE rollouts, aligning local dynamics with the response realized at inference (Chen et al., 2018; Yu et al., 2026).

2 Related Work

Perturbation representations. scGen and CPA model perturbation effects through latent transformations and compositional representations (Lotfollahi et al., 2019; 2023). GEARS and Attention-Pert incorporate gene relationships and multiscale effects to predict unseen genetic interventions (Roohani et al., 2024; Bai et al., 2024), while scGPT and scFoundation learn transferable cell and gene representations through large-scale pretraining (Cui et al., 2024; Hao et al., 2024). Perturb-Net, State, and TxPert further develop conditional generation, cross-context response modeling, and knowledge-guided perturbation representations (Yu et al., 2025; Adduri et al., 2026; Wenkel et al., 2026). Together with recent benchmarks, these studies motivate inductive structure tailored to perturbation prediction (Wu et al., 2025; Wei et al., 2026).

Population transport. Optimal transport relates independently sampled cellular populations through distribution-level correspondences (Schiebinger et al., 2019; Peyré & Cuturi, 2019). CellOT learns transport maps for heterogeneous perturbation responses, while CINEMA-OT uses transport to construct counterfactual correspondences and separate perturbation effects from confounding variation (Bunne et al., 2023; Dong et al., 2023). These approaches establish population geometry as a useful foundation for modeling unpaired perturbation data.

Continuous generative models. Flow Matching learns time-dependent vector fields through regression on prescribed probability paths (Lipman et al., 2023; Tong et al., 2024). CellFlow, scDFM, and CFM-GP extend conditional flows to cellular perturbation prediction (Klein et al., 2025; Yu et al., 2026; Abir et al., 2026). SDFM connects this formulation with knowledge-derived perturbation conditioning and trajectory-consistent population supervision: biological relations specify the intervention, OT defines training pairs, and teacher- and rollout-level objectives align local velocities with generated responses.

3 Method

3.1 Overview

Given control cells $X^c = \{x_i^c\}_{i=1}^{N_c} \subset \mathbb{R}^G$ and a target perturbation p, SDFM generates a perturbed population $\hat{X}^p$. Training additionally uses observed cells X^p, sampled independently of the controls. As shown in Figure 1, the model combines a biological graph encoder, OT pairing, and a conditional velocity field. At inference, stochasticized control cells are integrated through this field to produce perturbation-specific responses.

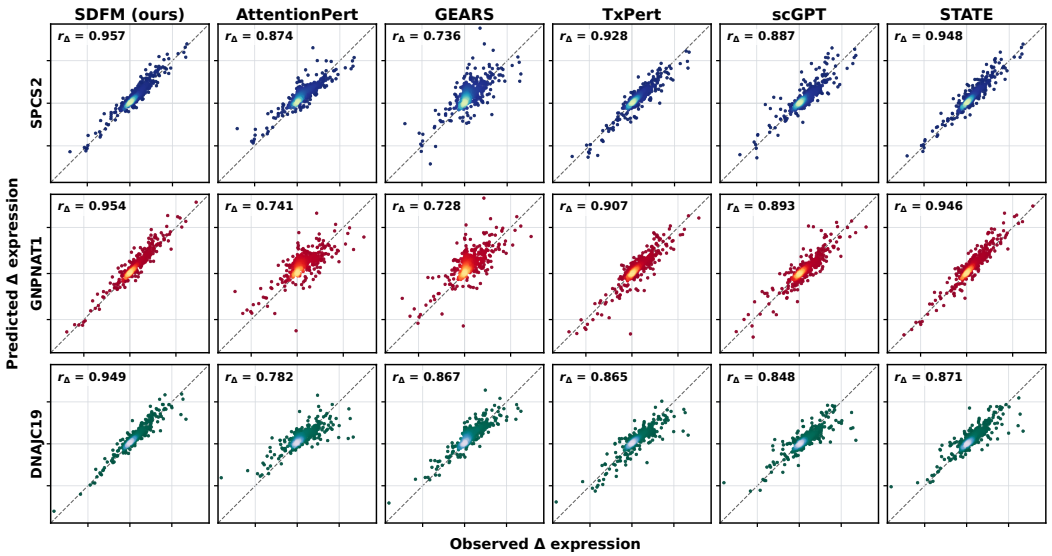

Figure 2: Gene-wise mean response prediction on Adamson. Rows show SPCS2, GNPNAT1, and DNAJC19; columns compare SDFM with five baselines. Points are genes, dashed lines indicate identity, and annotations report Pearson Δ.

3.2 Perturbation Conditioning

We encode molecular associations from STRING and functional relationships from Gene Ontology using Exphormer (Szklarczyk et al., 2023; The Gene Ontology Consortium, 2021; Shirzad et al., 2023):

$$e_p = f_\phi^{\mathrm{KG}}(p, \mathcal{G}_{\mathrm{STRING}}, \mathcal{G}_{\mathrm{GO}}), \qquad e_P = \sum_{p \in P} e_p + \gamma r_\psi(P). \tag{1}$$

Our Interaction-Aware Perturbation Composer averages learned symmetric pairwise features in $r_\psi(P)$, which vanishes for a single gene. The velocity network receives $[x_t, x^c, x_t - x^c]$ and uses multi-head cross-attention to perturbation–time tokens. A gated perturbation residual complements the state-dependent velocity (Appendix C.2).

The perturbation–time context $[e_p, \tau(t)]$ is projected into K condition tokens, which are used as keys and values in multi-head cross-attention with the state representation as the query:

$$C = \mathrm{reshape}_{K \times d_h}(W_c\,[e_p, \tau(t)] + b_c)\,, \qquad h_{l+1} = h_l + \mathrm{CrossAttn}\big(h_l,\, C\big), \tag{2}$$

where h_l is the state representation at block l. This lets each residual block dynamically aggregate perturbation information conditioned on the current cellular state.

3.3 OT-Coupled Flow Matching

For equal-size control and perturbed mini-batches $C_p = \{c_i\}_{i=1}^B$ and $Y_p = \{y_j\}_{j=1}^B$, we solve the squared-distance assignment

$$P^* = \arg\min_{P \in \mathfrak{S}_B} \sum_{i=1}^{B} \|c_i - y_{P(i)}\|_2^2, \qquad (x_i^c, x_i^p) = (c_i, y_{P^*(i)}). \tag{3}$$

This is an optimal transport coupling between uniformly weighted mini-batches (Peyré & Cuturi, 2019; Tong et al., 2024). For each coupled pair, Gaussian source noise and a random time define a linear path:

$$\begin{aligned} x_0 &= x^c + \sigma_s \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_G), \quad t \sim \mathcal{U}(0,1), \\ x_t &= (1-t)x_0 + t x^p, \qquad u_t = x^p - x_0. \end{aligned} \tag{4}$$

The knowledge-conditioned velocity is learned through Flow Matching (Lipman et al., 2023):

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,(x^c,x^p),\epsilon}\big[\|v_\theta(x_t, t, x^c, e_p) - u_t\|_2^2\big]. \tag{5}$$

Source randomness supplies variation around each control cell, while perturbation conditioning shapes the entire trajectory.

Table 1: Single-gene perturbation prediction. Values are mean ± SD; higher is better. DE20/DE100 report Pearson Δ on the top 20/100 DE genes. Bold and underlined means mark the best and second-best results, including ties.

	All genes		DE genes		Retrieval	
Method	Pearson Δ	Spearman Δ	DE20	DE100	Normalized	Fast
Adamson single ($n = 22$)						
SDFM (ours)	**0.8071** ± *0.0084*	**0.5517** ± *0.0066*	**0.8889** ± *0.0102*	**0.8540** ± *0.0096*	**0.6649** ± *0.0081*	**0.6642** ± *0.0059*
AttentionPert	0.6640 ± *0.0262*	0.5282 ± *0.0188*	0.7575 ± *0.0240*	0.7455 ± *0.0291*	0.5423 ± *0.0255*	0.5427 ± *0.0208*
GEARS	0.6704 ± *0.0236*	0.5101 ± *0.0215*	0.7736 ± *0.0357*	0.7489 ± *0.0203*	0.5862 ± *0.0170*	0.5856 ± *0.0147*
TxPert	0.7385 ± *0.0213*	0.5250 ± *0.0094*	0.7878 ± *0.0551*	0.7768 ± *0.0480*	0.5143 ± *0.0130*	0.5136 ± *0.0074*
scGPT	0.7016 ± *0.0224*	0.4974 ± *0.0076*	0.7753 ± *0.0174*	0.7497 ± *0.0115*	0.6369 ± *0.0142*	0.6363 ± *0.0131*
STATE	0.7189 ± *0.0425*	0.5325 ± *0.0055*	0.7575 ± *0.0252*	0.7514 ± *0.0171*	0.4857 ± *0.0161*	0.4862 ± *0.0088*
scDFM	0.6148 ± *0.0143*	0.3934 ± *0.0041*	0.7623 ± *0.0220*	0.7229 ± *0.0330*	0.4456 ± *0.0064*	0.4451 ± *0.0192*
CPA	0.5063 ± *0.0134*	0.3019 ± *0.0109*	0.6963 ± *0.0288*	0.6265 ± *0.0148*	0.4672 ± *0.0153*	0.4667 ± *0.0100*
HepG2 single ($n = 209$)						
SDFM (ours)	**0.2634** ± *0.0032*	**0.2363** ± *0.0037*	**0.3170** ± *0.0055*	**0.3153** ± *0.0054*	**0.5751** ± *0.0067*	**0.5855** ± *0.0075*
AttentionPert	0.2421 ± *0.0080*	0.2231 ± *0.0068*	0.3004 ± *0.0073*	0.2883 ± *0.0172*	0.5245 ± *0.0123*	0.5536 ± *0.0182*
GEARS	0.2427 ± *0.0094*	0.2203 ± *0.0059*	0.2982 ± *0.0222*	0.2946 ± *0.0084*	0.5043 ± *0.0184*	0.5235 ± *0.0230*
TxPert	0.2486 ± *0.0027*	0.2231 ± *0.0042*	0.2982 ± *0.0136*	0.2934 ± *0.0067*	0.5363 ± *0.0146*	0.5619 ± *0.0118*
scGPT	0.2467 ± *0.0069*	0.2103 ± *0.0085*	0.2837 ± *0.0198*	0.2709 ± *0.0091*	0.5329 ± *0.0216*	0.5503 ± *0.0126*
STATE	0.2311 ± *0.0044*	0.1893 ± *0.0029*	0.2814 ± *0.0082*	0.2931 ± *0.0060*	0.4752 ± *0.0154*	0.5063 ± *0.0166*
scDFM	0.2171 ± *0.0092*	0.1847 ± *0.0047*	0.2894 ± *0.0087*	0.2674 ± *0.0072*	0.4723 ± *0.0108*	0.5037 ± *0.0210*
CPA	0.0521 ± *0.0038*	0.0212 ± *0.0019*	0.1521 ± *0.0014*	0.1326 ± *0.0031*	0.4932 ± *0.0173*	0.4741 ± *0.0200*
Norman single ($n = 36$)						
SDFM (ours)	**0.3266** ± *0.0043*	**0.1587** ± *0.0048*	0.3681 ± *0.0127*	0.3536 ± *0.0078*	**0.5649** ± *0.0061*	**0.5531** ± *0.0056*
AttentionPert	0.2985 ± *0.0060*	0.1061 ± *0.0040*	0.3416 ± *0.0093*	0.3283 ± *0.0101*	0.5455 ± *0.0128*	0.5313 ± *0.0135*
GEARS	0.3043 ± *0.0071*	0.1193 ± *0.0053*	0.3371 ± *0.0113*	0.3304 ± *0.0107*	0.5341 ± *0.0209*	0.5204 ± *0.0135*
TxPert	0.3192 ± *0.0041*	0.1426 ± *0.0058*	0.3482 ± *0.0079*	0.3465 ± *0.0148*	0.5470 ± *0.0099*	0.5341 ± *0.0086*
scGPT	0.3101 ± *0.0034*	0.1451 ± *0.0020*	**0.3694** ± *0.0204*	**0.3571** ± *0.0120*	0.5413 ± *0.0056*	0.5425 ± *0.0062*
STATE	0.1248 ± *0.0049*	0.1331 ± *0.0035*	0.2501 ± *0.0077*	0.2036 ± *0.0095*	0.5155 ± *0.0119*	0.5294 ± *0.0125*
scDFM	0.2990 ± *0.0149*	0.1327 ± *0.0030*	0.3193 ± *0.0168*	0.3183 ± *0.0133*	0.4946 ± *0.0097*	0.4791 ± *0.0162*
CPA	0.1554 ± *0.0054*	0.0108 ± *0.0003*	0.2848 ± *0.0070*	0.2501 ± *0.0083*	0.5244 ± *0.0183*	0.5058 ± *0.0223*

3.4 Trajectory-Consistent Learning

The generated endpoint is obtained by integrating the same velocity field used in training (Chen et al., 2018):

$$\dot{x}(t) = v_\theta(x(t), t, x^c, e_p), \qquad x(0) = x_0, \qquad \hat{x}^p = \rho(x(1)), \tag{6}$$

where ρ maps the endpoint to nonnegative expression values. To connect local learning with this complete rollout, we also form the implied endpoint $\tilde{x}^p = \rho(x_t + (1-t)v_\theta(x_t, t, x^c, e_p))$. Its population shift and the rollout shift are compared with the observed effect $\Delta_p = \bar{x}^p - \bar{x}^c$, using scale-normalized Smooth-L_1 penalties. A paired endpoint loss further constrains the rolled-out states. The joint objective is

$$\mathcal{L} = \lambda_{\mathrm{FM}}\mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{T}}\mathcal{L}_{\mathrm{T}} + \lambda_{\mathrm{R}\Delta}\mathcal{L}_{\mathrm{R}\Delta} + \lambda_{\mathrm{RE}}\mathcal{L}_{\mathrm{RE}}. \tag{7}$$

The four terms align local velocities, implied population effects, and realized trajectories. We denote the Flow Matching, teacher-Δ, rollout-Δ, and endpoint reconstruction losses as $\mathcal{L}_{\mathrm{FM}}$, $\mathcal{L}_{\mathrm{T}}$, $\mathcal{L}_{\mathrm{R}\Delta}$, and $\mathcal{L}_{\mathrm{RE}}$, respectively, matching the notation used in Figure 1. Here, $\lambda_{\mathrm{FM}} = 1$. Model details and derivations are provided in Appendices C and D.

4 Results

4.1 Experimental Setup

We evaluated single-gene response prediction on Adamson, HepG2, and Norman, and dual-gene prediction on Norman (Adamson et al., 2016; Nadig et al., 2025; Norman et al., 2019).. Baselines included AttentionPert, GEARS, TxPert, scGPT, State, scDFM, and CPA (Bai et al., 2024; Roohani et al., 2024; Wenkel et al., 2026; Cui et al., 2024; Adduri et al., 2026; Yu et al., 2026; Lotfollahi et al., 2023). Norman dual-gene conditions were stratified into seen0, seen1, and seen2 according to the number of constituent genes observed during training.

Table 2: Dual-gene prediction on Norman. Overall values are condition-weighted means; subgroup values are mean ± SD. Bold and underlined values mark the best and second-best methods. Complete subgroup results appear in Table 8.

	All genes		DE genes		Retrieval	
Method	Pearson Δ	Spearman Δ	DE20	DE100	Normalized	Fast
(a) Overall ($n = 71$)						
SDFM (ours)	**0.6820**	**0.2594**	**0.7214**	**0.7224**	**0.8104**	**0.7979**
AttentionPert	0.6409	0.2111	<u>0.6890</u>	<u>0.6779</u>	<u>0.7794</u>	0.7640
GEARS	0.5610	0.1950	0.6858	0.6605	0.7568	0.7454
TxPert	<u>0.6496</u>	<u>0.2203</u>	0.6459	0.6453	0.7658	<u>0.7658</u>
scGPT	0.6363	0.2173	0.6645	0.6725	0.7527	0.7562
STATE	0.1161	0.1354	0.2743	0.2291	0.5360	0.5488
scDFM	0.4988	0.1890	0.6022	0.5691	0.6679	0.6525
CPA	0.3211	0.0936	0.4484	0.3772	0.6359	0.6270
(b) Comparisons by the number of seen genes						
Seen0 ($n = 9$)						
SDFM (ours)	**0.5734** ± *0.0093*	<u>0.1934</u> ± *0.0057*	**0.6392** ± *0.0068*	**0.6726** ± *0.0042*	**0.6888** ± *0.0094*	**0.6697** ± *0.0086*
AttentionPert	0.5517 ± *0.0097*	0.1181 ± *0.0043*	<u>0.6231</u> ± *0.0088*	<u>0.6126</u> ± *0.0107*	<u>0.6483</u> ± *0.0132*	0.6310 ± *0.0121*
TxPert	<u>0.5521</u> ± *0.0085*	0.1419 ± *0.0062*	0.4290 ± *0.0093*	0.5051 ± *0.0081*	0.6478 ± *0.0120*	0.6323 ± *0.0154*
scDFM	0.5294 ± *0.0141*	**0.2044** ± *0.0048*	0.5959 ± *0.0153*	0.5720 ± *0.0136*	0.6325 ± *0.0167*	0.6235 ± *0.0154*
GEARS	0.4481 ± *0.0087*	0.1650 ± *0.0051*	0.6121 ± *0.0096*	0.6030 ± *0.0089*	0.6281 ± *0.0103*	<u>0.6366</u> ± *0.0098*
Seen1 ($n = 43$)						
SDFM (ours)	**0.6429** ± *0.0068*	**0.2477** ± *0.0084*	**0.6912** ± *0.0097*	**0.6815** ± *0.0082*	**0.7829** ± *0.0075*	**0.7705** ± *0.0069*
AttentionPert	<u>0.6234</u> ± *0.0072*	<u>0.2132</u> ± *0.0053*	0.6508 ± *0.0081*	0.6369 ± *0.0076*	0.7535 ± *0.0093*	0.7375 ± *0.0087*
GEARS	0.5551 ± *0.0114*	0.1863 ± *0.0047*	<u>0.6628</u> ± *0.0108*	<u>0.6468</u> ± *0.0095*	<u>0.7577</u> ± *0.0121*	<u>0.7441</u> ± *0.0112*
Seen2 ($n = 19$)						
SDFM (ours)	**0.8221** ± *0.0143*	**0.3171** ± *0.0127*	**0.8287** ± *0.0105*	**0.8387** ± *0.0076*	**0.9302** ± *0.0098*	**0.9205** ± *0.0087*
TxPert	<u>0.8018</u> ± *0.0078*	<u>0.2936</u> ± *0.0057*	<u>0.8087</u> ± *0.0083*	0.7997 ± *0.0072*	<u>0.9124</u> ± *0.0091*	0.9023 ± *0.0084*
AttentionPert	0.7226 ± *0.0086*	0.2504 ± *0.0054*	0.8065 ± *0.0079*	<u>0.8017</u> ± *0.0068*	0.9001 ± *0.0104*	0.8870 ± *0.0096*
scGPT	0.7772 ± *0.0092*	0.2827 ± *0.0051*	0.7810 ± *0.0087*	0.7745 ± *0.0076*	0.9108 ± *0.0073*	<u>0.9027</u> ± *0.0068*

We assessed mean expression changes relative to controls (Δ), differentially expressed gene subsets, perturbation retrieval, and population distributions. These complementary views follow the emphasis on task-specific evaluation in recent perturbation benchmarks (Wu et al., 2025; Wei et al., 2026). Data and evaluation details appear in Appendix B, with extended results in Appendix E.

4.2 Single-Gene Prediction

SDFM achieved the highest mean all-gene Pearson Δ on all three single-gene benchmarks (Table 1). On Adamson, it reached 0.8071, compared with 0.7385 for TxPert, and led all six metrics. Figure 2 illustrates the agreement between predicted and observed expression changes for SPCS2, GNPNAT1, and DNAJC19, with Pearson Δ values of 0.957, 0.954, and 0.949, respectively. On HepG2 and Norman single, SDFM reached Pearson Δ values of 0.2634 and 0.3266, compared with 0.2486 and 0.3192 for TxPert. SDFM led all six metrics on HepG2 and both retrieval metrics on every dataset. On Norman, scGPT had slightly higher DE20 and DE100 means, while SDFM retained the strongest all-gene correlations and perturbation retrieval. Further comparisons with CellOT, CellFlow and related methods appear in Appendix E.

4.3 Combinatorial Prediction

SDFM achieved the highest condition-weighted overall mean across all six metrics on Norman dual-gene prediction (Table 2). Pearson Δ increased from 0.6496 for TxPert to 0.6820 for SDFM, while Spearman Δ increased from 0.2203 to 0.2594. SDFM also improved DE100 from 0.6779 for AttentionPert to 0.7224 and normalized retrieval from 0.7794 to 0.8104. These gains extend from the full response profile to differentially expressed genes and perturbation identification.

The stratified results show gains even when neither constituent gene was seen during training. In seen0, SDFM led five of the six metrics: Pearson Δ reached 0.5734, compared with 0.5521 for TxPert, and normalized retrieval reached 0.6888, compared with 0.6483 for AttentionPert. scDFM led Spearman Δ at 0.2044, followed by SDFM at 0.1934.

Table 3: Single-gene component ablations. Values are mean ± SD; bold and underlined means mark the best and second-best variants. Retrieval is normalized.

	Adamson		HepG2		Norman	
Variant	Pearson Δ	Retrieval	Pearson Δ	Retrieval	Pearson Δ	Retrieval
Full SDFM	**0.8071** ± *0.0084*	**0.6649** ± *0.0081*	**0.2634** ± *0.0032*	**0.5751** ± *0.0067*	**0.3266** ± *0.0043*	**0.5649** ± *0.0061*
Knowledge graph encoding						
w/o STRING	<u>0.7894</u> ± *0.0059*	0.6359 ± *0.0127*	0.2550 ± *0.0070*	0.5388 ± *0.0135*	0.2932 ± *0.0080*	0.5168 ± *0.0183*
w/o GO	0.7779 ± *0.0096*	0.6275 ± *0.0141*	<u>0.2581</u> ± *0.0026*	0.5575 ± *0.0195*	0.2862 ± *0.0085*	0.5360 ± *0.0105*
Emb-MLP	0.7079 ± *0.0114*	0.6081 ± *0.0209*	0.2269 ± *0.0040*	0.5221 ± *0.0111*	0.2790 ± *0.0054*	0.5063 ± *0.0148*
GATv2	0.7878 ± *0.0069*	0.6428 ± *0.0053*	0.2239 ± *0.0064*	0.5402 ± *0.0071*	<u>0.3065</u> ± *0.0097*	0.5266 ± *0.0221*
Dual-branch KG	0.7616 ± *0.0077*	0.6421 ± *0.0107*	0.2373 ± *0.0034*	0.5436 ± *0.0051*	0.2989 ± *0.0023*	0.5469 ± *0.0041*
OT pairing						
w/o OT	0.7391 ± *0.0106*	0.6400 ± *0.0099*	0.2257 ± *0.0043*	0.5467 ± *0.0183*	0.2953 ± *0.0038*	0.5333 ± *0.0090*
Sinkhorn	0.7423 ± *0.0122*	0.6379 ± *0.0120*	0.2210 ± *0.0061*	0.5319 ± *0.0048*	0.3035 ± *0.0127*	<u>0.5494</u> ± *0.0116*
Random	0.6950 ± *0.0150*	0.6050 ± *0.0200*	0.2100 ± *0.0070*	0.5200 ± *0.0150*	0.2700 ± *0.0120*	0.5100 ± *0.0180*
Trajectory supervision						
w/o teacher-Δ	0.7387 ± *0.0091*	0.6239 ± *0.0149*	0.2305 ± *0.0055*	0.5438 ± *0.0145*	0.2998 ± *0.0141*	0.5297 ± *0.0197*
w/o rollout-Δ	0.7817 ± *0.0063*	<u>0.6477</u> ± *0.0050*	0.2374 ± *0.0057*	0.5625 ± *0.0174*	0.3007 ± *0.0116*	0.5338 ± *0.0046*
w/o endpoint	0.7837 ± *0.0045*	0.6409 ± *0.0088*	0.2419 ± *0.0031*	<u>0.5657</u> ± *0.0044*	0.3051 ± *0.0041*	0.5328 ± *0.0174*
Conditioning mechanism						
Concat	0.7670 ± *0.0131*	0.6383 ± *0.0113*	0.2246 ± *0.0074*	0.5312 ± *0.0153*	0.2751 ± *0.0046*	0.5225 ± *0.0233*
w/o cell state	0.7250 ± *0.0113*	0.6050 ± *0.0179*	0.2320 ± *0.0057*	0.5250 ± *0.0126*	0.2850 ± *0.0070*	0.5100 ± *0.0154*
w/o time emb	0.7600 ± *0.0086*	0.6300 ± *0.0140*	0.2480 ± *0.0042*	0.5500 ± *0.0140*	0.3050 ± *0.0063*	0.5350 ± *0.0105*
Auxiliary loss						
MSE aux.	0.7784 ± *0.0139*	0.6429 ± *0.0157*	0.2357 ± *0.0048*	0.5131 ± *0.0121*	0.2959 ± *0.0064*	0.5208 ± *0.0208*

SDFM led all six metrics in both seen1 and seen2. Its Pearson Δ increased from 0.6429 in seen1 to 0.8221 in seen2, with normalized retrieval increasing from 0.7829 to 0.9302. In seen2, DE100 reached 0.8387, compared with 0.8017 for AttentionPert. These results connect stronger response prediction with more accurate identification of the perturbation as the constituent genes become better represented in training.

Table 4: Effect of removing key components on Norman dual-gene prediction.

Method	Pearson Δ	DE100	Retrieval
Full SDFM	**0.6820**	**0.7224**	**0.8104**
w/o OT	0.5866	0.6360	0.7192
w/o rollout-Δ	0.6254	0.6679	0.7575
w/o Inter-Aware	0.6667	0.7071	0.7943

To assess each design choice, we conducted component ablations and replacements (Table 3). Removing biological knowledge, OT pairing, or population-level supervision consistently reduced performance. On Adamson, removing OT pairing reduced Pearson Δ from 0.8071 to 0.7391, teacher-Δ supervision to 0.7387, and replacing the graph encoder with Emb-MLP to 0.7079. These results highlight the contribution of biological representations, source–target alignment, and transport conditioning.Ablations on dual-gene perturbations showed similar trends. Removing OT pairing reduced overall Pearson Δ from 0.6820 to 0.5866, while removing teacher-Δ, rollout-Δ, or the Interaction-Aware module reduced it to 0.6140, 0.6254, and 0.6667, respectively. This is consistent with the model's motivation: population-level effect supervision beyond local flow matching helps improve the perturbation responses generated by a full rollout, supporting the contribution of each component (Table 4). Supplementary deterministic and graph-randomization controls further examine the contributions of cell-state conditioning and biological connectivity.

4.4 Predicted Interaction Structure

We used SDFM to map pairwise perturbation responses beyond the experimentally assayed conditions (Figure 3). The predicted landscapes contain distinct, partially overlapping response regions organized by biological marker groups. The added held-out comparison quantifies agreement in response geometry: Spearman correlations between predicted and observed distance matrices were 0.42 on Norman, 0.61 on Adamson, and 0.46 on HepG2. The coverage curves further locate unas-

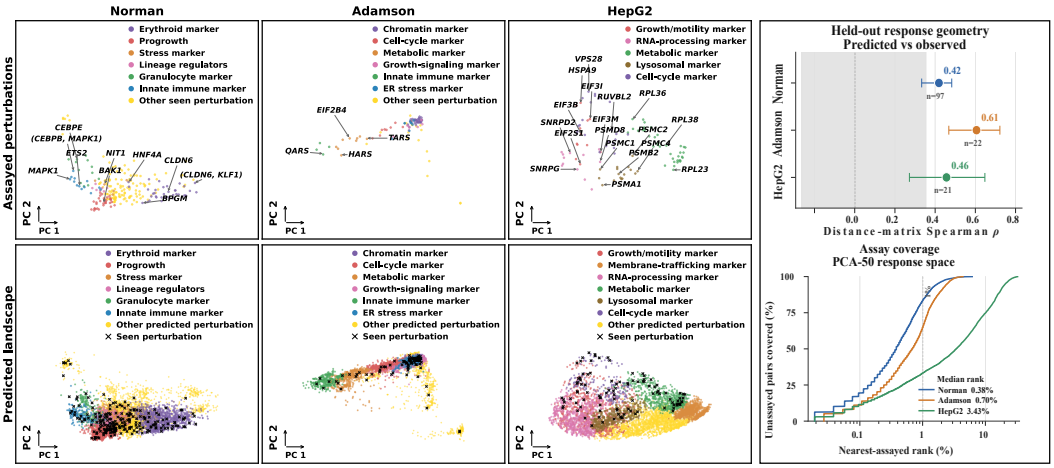

Figure 3: Perturbation response landscapes and pairwise-space coverage. Left: assayed responses (top) and expanded pairwise responses (bottom) for Norman, Adamson, and HepG2. Norman uses learned combinatorial predictions; Adamson and HepG2 use additive virtual pairs, $\hat{\Delta}^{\mathrm{add}}_{A+B} = \hat{\Delta}_A + \hat{\Delta}_B$. Right: held-out response-geometry agreement and nearest-assayed coverage in PCA-50 space.

sayed combinations relative to experimentally sampled responses, connecting the expanded prediction space to the available measurements.

We next examined the predicted interaction structure on Norman, which contains experimentally measured paired perturbations and therefore provides the appropriate setting for analyzing combinatorial effects. The predicted GI matrix summarizes five interaction-like categories—synergy, suppression, neomorphism, relative epistasis, and redundancy—and reveals structured block and stripe patterns across gene pairs (Figure 4) (Norman et al., 2019). This analysis complements the quantitative dual-gene benchmark by visualizing the interaction structure induced by the predicted combinatorial responses. Exploratory pairwise maps for Adamson and HepG2, which contain only single-gene perturbation measurements, are reported separately in Appendix E.7.

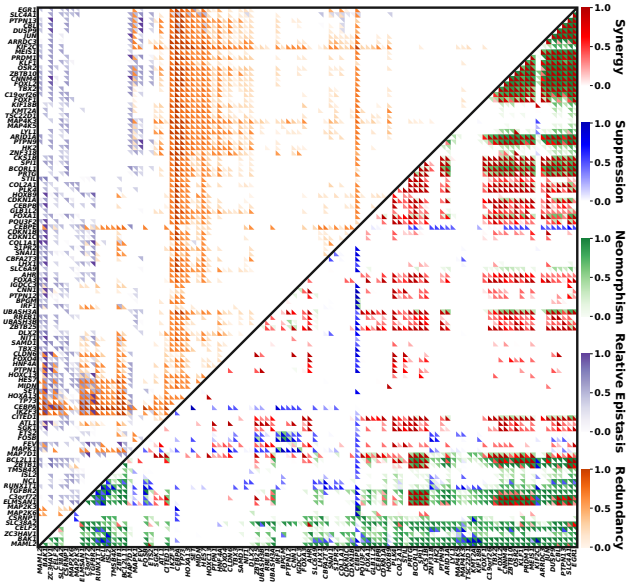

Figure 4: Predicted interaction-like score maps from virtual perturbation combinations on Norman.

4.5 Population Recovery

Accurate prediction of mean expression changes does not guarantee that generated populations exhibit cell-to-cell variability similar to that of true populations. To examine this, we further compared the response distributions of predicted and observed cell populations (Figure 5). For TBX3+TBX2, DUSP9+IGDCC3, and CBL+PTPN9, SDFM achieved SWD ratios of 0.27, 0.31, and 0.32, respectively, the lowest in each displayed comparison. Its contours followed the observed population locations and orientations, while retaining variation along the response axis. The marginal densities show how this agreement differs from predictions that concentrate too tightly or shift away from the observed response. These examples connect accurate average effects with recovery of heterogeneous populations. Across perturbation conditions, SDFM reduced mean sliced Wasserstein and

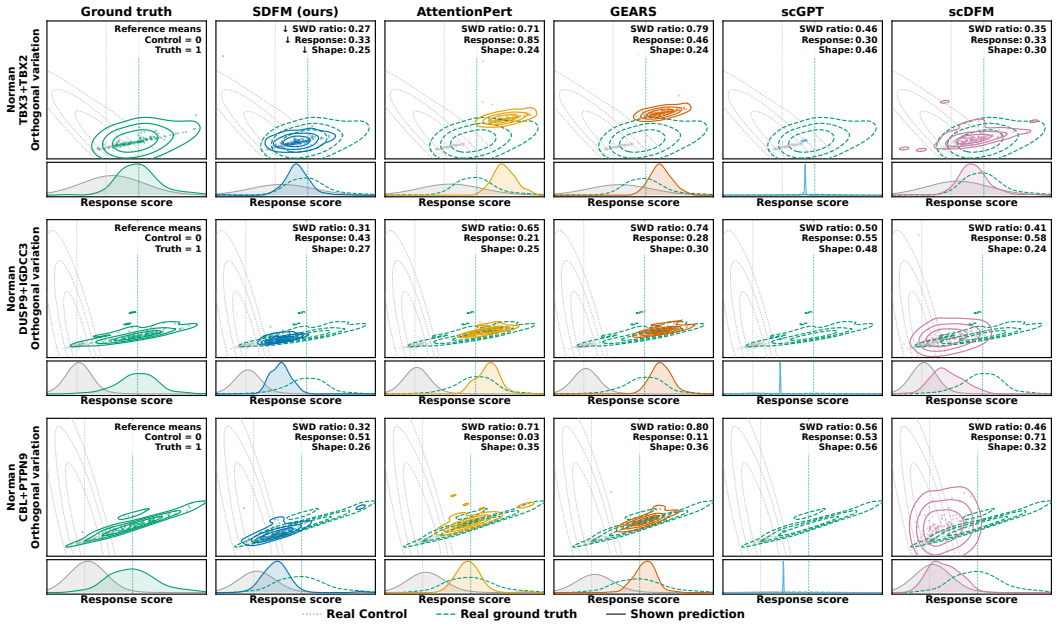

Figure 5: Population response recovery for TBX3+TBX2, DUSP9+IGDCC3, and CBL+PTPN9 on Norman. Contours and marginal densities compare control, observed perturbed, and predicted populations along response-score and orthogonal-variation axes. Annotations summarize SWD ratio, response, and shape.

energy distances relative to the uniform mean-shift predictor (Bunne et al., 2023). The supplementary population panels show how these gains vary across individual conditions.

5 Conclusion

We presented SDFM, a knowledge-conditioned flow matching framework for predicting genetic perturbation responses from unpaired cell populations. SDFM integrates biological knowledge, optimal transport pairing, and trajectory-consistent supervision to connect local velocity learning with population-level responses. Experiments on Adamson, HepG2, and Norman demonstrate improvements in genome-wide mean response prediction and perturbation retrieval, with complementary evidence of improved population distribution recovery. Ablation studies support the contributions of these components, while showing that their benefits vary across datasets and evaluation metrics, consistent with recent observations that perturbation prediction performance is highly task- and dataset-dependent (Wu et al., 2025; Wei et al., 2026).

6 Discussion

Our results highlight the value of jointly modeling perturbation semantics, population geometry, and trajectory-level outcomes, while revealing several limitations. Improvements in genome-wide response prediction do not always translate to differential gene prediction, consistent with prior benchmarking observations that evaluation criteria may favor different models (Wu et al., 2025; Wei et al., 2026). Combinatorial generalization remains limited by perturbations observed during training, as also noted in multigene prediction methods (Roohani et al., 2024; Bai et al., 2024). Moreover, trajectories inferred from unpaired endpoint populations represent distribution-level correspondences rather than experimentally measured cellular dynamics (Bunne et al., 2023; Dong et al., 2023). Future work will focus on improving unseen-combination prediction, cross-context transfer, and experimental validation of predicted interactions (Adduri et al., 2026).

AI USAGE STATEMENT

This work used AI-assisted tools for writing assistance, language polishing, literature exploration, and code debugging. Specifically, ChatGPT was used for problem discussion and writing refinement, while Claude and Codex were used for code debugging and assistance with figure and formatting organization. All AI-assisted contents and suggestions were reviewed, verified, and revised by the authors when necessary. The authors take full responsibility for the accuracy, originality, citation integrity, and ethical compliance of the final manuscript.

REFERENCES

Abrar Rahman Abir, Sajib Acharjee Dip, and Liqing Zhang. Cfm-gp: Unified conditional flow matching to learn gene perturbation across cell types. *NAR Genomics and Bioinformatics*, 8(3): lqag087, 2026. doi: 10.1093/nargab/lqag087.

Britt Adamson, Thomas M. Norman, Marco Jost, Min Y. Cho, James K. Nuñez, Yuwen Chen, Jacqueline E. Villalta, Luke A. Gilbert, Max A. Horlbeck, Marco Y. Hein, Ryan A. Pak, Andrew N. Gray, Carol A. Gross, Atray Dixit, Oren Parnas, Aviv Regev, and Jonathan S. Weissman. A multiplexed single-cell CRISPR screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882.e21, 2016. doi: 10.1016/j.cell.2016.11.048.

Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Mohsen Naghipourfar, Alishba Imran, Rohan Shah, Noam Teyssier, et al. Predicting cellular responses to perturbation across diverse contexts with state. *Cell*, 2026. doi: 10.1016/j.cell.2026.07.052.

Ding Bai, Caleb N. Ellington, Shentong Mo, Le Song, and Eric P. Xing. AttentionPert: accurately modeling multiplexed genetic perturbations with multi-scale effects. *Bioinformatics*, 40 (Supplement 1):i453–i461, 2024. doi: 10.1093/bioinformatics/btae244.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2105.14491.

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*, 20:1759–1768, 2023. doi: 10.1038/s41592-023-01969-x.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, Bo Wang, et al. scgpt: Toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21:1470–1480, 2024. doi: 10.1038/s41592-024-02201-0.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Ishan Deshpande, Ziyu Zhang, and Alexander G. Schwing. Generative modeling using the sliced Wasserstein distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3483–3491, 2018. URL https://arxiv.org/abs/1803.11188.

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.e17, 2016. doi: 10.1016/j.cell.2016.11.038.

Mingze Dong, Bao Wang, Jessica Wei, Antonio H. de O. Fonseca, Curtis J. Perry, Alexander Frey, et al. Causal identification of single-cell experimental perturbation effects with cinema-ot. *Nature Methods*, 20:1769–1779, 2023. doi: 10.1038/s41592-023-02040-5.

J. R. Dormand and P. J. Prince. A family of embedded Runge–Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980. doi: 10.1016/0771-050X(80)90013-3.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012. URL https://jmlr.org/papers/v13/gretton12a.html.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, second edition, 1993. doi: 10.1007/978-3-540-78862-1.

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*, 21:1481–1491, 2024. doi: 10.1038/s41592-024-02305-7.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732.

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. doi: 10.1101/2025.04.11.648220.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L. Ibarra, Sanjay R. Srivatsan, Mohsen Naghipourfar, Riza M. Daza, Beth Martin, Jay Shendure, Jose L. McFaline-Figueroa, Pierre Boyeau, F. Alexander Wolf, Nafissa Yakubova, Stephan Günnemann, Cole Trapnell, David Lopez-Paz, and Fabian J. Theis. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):e11517, 2023. doi: 10.15252/msb.202211517.

Sergei Maslov and Kim Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913, 2002. doi: 10.1126/science.1065103.

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*, 57:1228–1237, 2025. doi: 10.1038/s41588-025-02169-3.

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.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

Joseph M. Replogle, Reuben A. Saunders, Andrew 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.e28, 2022. doi: 10.1016/j.cell.2022.05.013.

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

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev, and Eric S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22, 2019. doi: 10.1016/j.cell.2019.01.006.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J. Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31613–31632, 2023.

Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908. doi: 10.1093/biomet/6.1.1.

Gábor J. Székely and Maria L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013. doi: 10.1016/j.jspi.2013.03.018.

Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L. Gable, Tao Fang, Nadezhda T. Doncheva, Sampo Pyysalo, Peer Bork, Lars J. Jensen, and Christian von Mering. 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. doi: 10.1093/nar/gkac1000.

The Gene Ontology Consortium. The gene ontology resource: enriching a gold mine. *Nucleic Acids Research*, 49(D1):D325–D334, 2021. doi: 10.1093/nar/gkaa1113.

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. *Transactions on Machine Learning Research*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zhiting Wei, Yiheng Wang, Yicheng Gao, Shuguang Wang, Ping Li, Duanmiao Si, Yuli Gao, Siqi Wu, Danlu Li, Kejing Dong, Xingbo Yang, Chen Tang, Shaliu Fu, Xiaohan Chen, Wannian Li, Yuzhou You, Chen Zhang, Aibin Liang, Guohui Chuai, and Qi Liu. Benchmarking algorithms for generalizable single-cell perturbation response prediction. *Nature Methods*, 23(2):451–464, 2026. doi: 10.1038/s41592-025-02980-0.

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.

Yan Wu, Esther Wershof, Sebastian M. Schmon, Marcel Nassar, Błażej Osiński, Ridvan Eksi, Zichao Yan, Rory Stark, Kun Zhang, and Thore Graepel. Perturbench: Benchmarking machine learning models for cellular perturbation analysis. In *Advances in Neural Information Processing Systems*, 2025.

Chenglei Yu, Chuanrui Wang, Bangyan Liao, and Tailin Wu. scdfm: Distributional flow matching model for robust single-cell perturbation prediction. In *International Conference on Learning Representations*, 2026.

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):960–982, 2025. doi: 10.1038/s44320-025-00131-3.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Appendix

A Notation

Table 5 collects the symbols used in the model and derivations. We use d_e for embedding dimension, d_p for the endpoint-loss scale, M for the number of perturbed genes, and K for the number of condition tokens.

Table 5: Notation for SDFM and its derivations.

Symbol	Meaning
$G; N_c, N_p$	Number of expression genes; numbers of control and perturbed cells.
$X^c, X^p, \hat{X}^p$	Control, observed perturbed, and generated cell populations.
$x^c, x^p, \hat{x}^p$	Corresponding individual cell expression vectors.
$\mu_c, \mu_p, \hat{\mu}_p$	Control, observed perturbed, and generated distributions.
$p; P, M$	A target perturbation; a set of perturbed genes and its size.
$\mathcal{G}_{\mathrm{STRING}}, \mathcal{G}_{\mathrm{GO}}$	Biological graphs, also denoted $\mathcal{G}_S, \mathcal{G}_G$.
$f_\phi^{\mathrm{KG}}; e_p, e_P, d_e$	Graph encoder; single and combined embeddings and their dimension.
B, B_r	OT pairing and rollout-supervision batch sizes.
$D_{ij}; P^*, \pi_p^*$	Squared cell-distance cost; optimal assignment and induced coupling.
$\mathfrak{S}_B$	Permutations of the B target cells.
$x_0, x_t, t; \sigma_s, \epsilon$	Noisy source, path state, and time; source-noise scale and Gaussian draw.
$u_t; v_\theta$	Pair-conditional target velocity; learned conditional velocity field.
$s_t, h_l, c_{p,t}$	State input, hidden state at layer l, and perturbation–time context.
$C_l; K, d_h, L$	Condition tokens; token count, hidden dimension, and block count.
$\phi_\psi, r_\psi(P); \gamma$	Pairwise MLP, averaged interaction residual, and interaction gate.
$H, d_a; Q_l^{(a)}, K_l^{(a)}, V_l^{(a)}$	Attention heads, head width, and head-specific query, key, and value.
g_l, α	Attention-residual and perturbation-velocity gates.
$\rho; \Phi_p$	Non-negativity map; endpoint map of control cells and source noise.
$\tilde{x}^p$	One-step teacher endpoint derived from a local velocity estimate.
$\Delta_p, \tilde{\Delta}_p, \hat{\Delta}_p$	Observed, teacher-predicted, and rollout-predicted mean shifts.
$s_p, s_{\min}, d_p$	Effect scale, its lower bound, and OT endpoint-displacement scale.
ℓ_β	Smooth-L_1 penalty with transition parameter β.
$\lambda_{\mathrm{FM}}, \lambda_{\mathrm{T}}, \lambda_{\mathrm{R}\Delta}, \lambda_{\mathrm{RE}}$	Weights of the four training objectives.
$q_t; (\cdot)_\#$	Conditional path density; distribution push-forward.
h	Maximum numerical integration step size.

B Data and Evaluation

B.1 Datasets and Splits

Adamson measures single-gene CRISPRi responses associated with the unfolded protein response in K562 cells (Adamson et al., 2016). Norman uses single and paired CRISPRa perturbations in K562 cells to characterize genetic interactions (Norman et al., 2019). HepG2 provides a larger essential-gene CRISPRi screen in a distinct cellular context (Nadig et al., 2025). Table 6 summarizes the splits used here. Controls are included in training; all listed condition counts exclude controls.

Adamson and HepG2 evaluate held-out single-gene perturbations. Norman includes 36 unseen single-gene conditions and 71 dual-gene conditions: 9 seen0, 43 seen1, and 19 seen2. The latter grouping counts how many constituent genes occur in training perturbations, following the distinction between unseen genes and unseen combinations used in combinatorial prediction (Roohani et al., 2024; Bai et al., 2024). The 33 genes shared between Norman training and test conditions support this stratification.

The evaluation scope is specified for each analysis. The single-gene comparison in Table 1 contains 22 Adamson, 209 HepG2, and 36 Norman conditions; the split inventory in Table 6 lists 594 held-out HepG2 conditions. The population-distribution analysis in Appendix E.2 covers 22 Adamson,

Table 6: Dataset sizes and condition splits, excluding `ctrl`. Seen-k counts genes observed during training; overlap counts genes shared by training and test conditions.

Dataset	Cells	Train/Val/Test(Pert)	Unused	Overlap	Test conditions
Adamson	68,603	57/7/22	0	0	Unseen single: 22
Norman	91,205	138/31/107	7	33	Seen0: 9; Seen1: 43; Seen2: 19; Unseen single: 36
HepG2	145,473	1601/178/594	20	0	Unseen single: 594

583 HepG2, and all 107 Norman test conditions. The additional deterministic and graph controls are presented within their own experimental panels in Appendices E.4 and E.5.

B.2 Response Metrics

For each perturbation, we compare the observed and predicted population-mean shifts, $\Delta_p = \bar{x}^p - \bar{x}^c$ and $\hat{\Delta}_p = \bar{\hat{x}}^p - \bar{x}^c$. Pearson Δ measures agreement in their gene-wise profiles, and Spearman Δ measures rank agreement. DE20 and DE100 restrict Pearson Δ to the top 20 and 100 differentially expressed genes for the condition. These metrics describe complementary aspects of response prediction (Roohani et al., 2024; Bai et al., 2024; Wei et al., 2026).

Normalized retrieval and fast retrieval quantify how well a predicted response identifies its matching perturbation among candidate conditions; higher values indicate better discrimination. Table 1 reports both scores. All-gene, DE-subset, and retrieval results are considered together because matching a broad response profile and distinguishing related perturbations impose different demands on a model (Wu et al., 2025; Wenkel et al., 2026).

For the Norman dual-gene benchmark and ablations, Overall means combine the seen0, seen1, and seen2 groups according to their numbers of perturbation conditions:

$$\bar{z}_{\text{Overall}} = \frac{9\bar{z}_{\text{seen0}} + 43\bar{z}_{\text{seen1}} + 19\bar{z}_{\text{seen2}}}{71}.$$

This weighting gives each of the 71 evaluation conditions equal weight. Group means and standard deviations summarize five experimental replicates.

B.3 Distributional Evaluation

We fit a standardization transform and a 50-dimensional PCA representation on control and training-condition cells, using at most 20,000 cells with seed 42. For each condition and method, observed and generated populations are subsampled to $N = \min(n_{\text{real}}, n_{\text{pred}}, 512)$ cells each. We average ten repetitions using shared subsampling seeds and then aggregate over perturbation conditions.

Sliced Wasserstein distance (SWD) is computed from the root mean square of one-dimensional Wasserstein distances over 100 fixed projection directions (Deshpande et al., 2018). Energy distance summarizes between-population distances relative to within-population distances (Székely & Rizzo, 2013). MMD uses a biased squared-MMD estimator with an RBF kernel whose bandwidth is chosen by the median heuristic on the pooled observed and generated samples (Gretton et al., 2012). These three measures and normalized covariance error are computed in PCA space. The latter is $\|\Sigma_{\text{pred}} - \Sigma_{\text{real}}\|_F / \|\Sigma_{\text{real}}\|_F$. Per-gene variance correlation is computed in expression space. This combination evaluates distributional location, spread, and dependence, extending the population-level view of perturbation response modeling (Bunne et al., 2023; Yu et al., 2026).

Table 10 reports condition-averaged distances for SDFM and MeanShift-Delta. Figures 5 and 6–8 provide the corresponding qualitative view through two-dimensional density contours and one-dimensional response profiles; their annotations report the case-level SWD ratios.

B.4 Statistical Comparisons

The benchmark and ablation comparisons use two-sided independent-sample Student t-tests with equal variances (Student, 1908). From the tabulated means $\bar{z}_i$ and standard deviations s_i, we com-

pute

$$s_p^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}, \qquad t = \frac{\bar{z}_1 - \bar{z}_2}{s_p\sqrt{1/n_1 + 1/n_2}}, \qquad df = n_1 + n_2 - 2.$$

Each method contributes five replicates, giving $df = 8$ and $p = 2\Pr(T_8 \geq |t|)$. Perturbation counts in the table headings describe evaluation groups, not replicate counts. We compute tests before rounding, report t to three decimal places and p to three significant figures, and use an unadjusted threshold of $p < 0.05$.

Benchmark tests compare the best and second-best means for the six main-text metrics; tied runners-up are tested separately. Ablation tests compare Full SDFM with the best remaining variant for each reported metric. These benchmark and component-ablation tests cover the single-gene datasets and the three seen-gene groups of Norman dual-gene prediction. Tables 7 and 9 give the benchmark comparisons; Tables **??** and **??** give the ablation comparisons. All calculations use the means and standard deviations in the corresponding tables. The deterministic and graph controls are summarized descriptively in Tables 14 and 15. The response-metric comparisons in the additional controls use the same Student t-test convention.

C Model Details

C.1 Population Transport

Single-cell perturbation prediction aims to characterize the transcriptional distribution induced by a target genetic perturbation given a population of unperturbed cells. This task has been broadly studied as a conditional cell-state transformation problem (Lotfollahi et al., 2019; 2023; Roohani et al., 2024; Wenkel et al., 2026).

For a perturbation condition p, let the control and observed perturbed populations be

$$X^c = \{x_i^c\}_{i=1}^{N_c}, \qquad X^p = \{x_j^p\}_{j=1}^{N_p}.$$

Their empirical distributions are

$$\mu_c = \frac{1}{N_c}\sum_{i=1}^{N_c}\delta_{x_i^c}, \qquad \mu_p = \frac{1}{N_p}\sum_{j=1}^{N_p}\delta_{x_j^p}, \tag{8}$$

where $x \in \mathbb{R}^G$ and G denotes the number of genes in the expression space.

We learn a perturbation-conditioned continuous transport operator Φ_p such that the control distribution is mapped to the predicted perturbed distribution:

$$\hat{\mu}_p = (\Phi_p)_{\#}(\mu_c \otimes \mathcal{N}(0, I_G)), \qquad \Phi_p(x^c, \epsilon) = \hat{x}^p, \tag{9}$$

where $(\cdot)_{\#}$ denotes the push-forward operator and Φ_p includes initialization at $x^c + \sigma_s\epsilon$, ODE integration, and the expression-space mapping ρ. The construction transports the joint distribution of control cells and source noise (Peyré & Cuturi, 2019; Lipman et al., 2023).

The model therefore learns a perturbation-conditioned transport of cell populations.

C.2 Conditioning and Velocity

The transcriptional consequence of a genetic perturbation is shaped by both the perturbed gene and its broader molecular and functional context. We therefore construct gene-relation graphs from STRING (Szklarczyk et al., 2023) and Gene Ontology (The Gene Ontology Consortium, 2021), and encode their structure using Exphormer (Shirzad et al., 2023). This follows the general principle of learning perturbation representations from multiple biological knowledge graphs (Wenkel et al., 2026), while further using these representations to condition continuous cellular dynamics.

Let the two knowledge graphs be

$$\mathcal{G}_S = (V, E_S), \qquad \mathcal{G}_G = (V, E_G).$$

For a target perturbation p, its representation is defined as

$$e_p = f_\phi^{\mathrm{KG}}(p, \mathcal{G}_S, \mathcal{G}_G) \in \mathbb{R}^{d_e}. \tag{10}$$

Interaction-Aware Perturbation Composer. For $P = \{p_1, \ldots, p_M\}$, let $e_i = e_{p_i}$. The additive component preserves the individual gene representations, while a symmetric pairwise residual models their joint context:

$$\begin{aligned} z_{ij} &= [e_i \odot e_j,\ |e_i - e_j|], \qquad & \phi_\psi(z) &= W_2\,\mathrm{SiLU}(W_1 z + b_1) + b_2, \\ r_\psi(P) &= \begin{cases} \binom{M}{2}^{-1} \sum_{i<j} \phi_\psi(z_{ij}), & M \geq 2, \\ 0, & M = 1, \end{cases} & e_P &= \sum_{i=1}^{M} e_i + \gamma r_\psi(P). \end{aligned} \tag{11}$$

Here $\odot$ denotes elementwise multiplication, absolute values are elementwise, and γ is a learned scalar initialized to 0.1. The MLP maps $2d_e$ inputs through a hidden layer back to d_e outputs; its default hidden width is d_e. Averaging over unordered pairs makes the interaction residual invariant to gene ordering and controls its scaling as the number of pairs increases. Both the additive term and the residual are symmetric set functions (Zaheer et al., 2017). For two genes the residual contains one pair; for one gene the output is exactly its original embedding.

Training, validation, and generation use the same condition-composition function. The composer is optimized jointly with the graph encoder in the same optimizer parameter group. Setting `interaction_aware=false` recovers additive composition. When loading a checkpoint that predates the composer, the implementation disables the interaction branch to preserve that checkpoint's predictions. In the remaining transport equations, e_p denotes the embedding of a condition, using the composed e_P for multi-gene conditions.

Multi-head perturbation–time attention. During continuous transport, the velocity field observes both the current state and its original control context. We define

$$s_t = [x_t,\ x^c,\ x_t - x^c], \qquad h_0 = f_s(s_t), \qquad c_{p,t} = f_c([e_p, \tau(t)]), \tag{12}$$

where $\tau(t)$ denotes a continuous time embedding and h_0 is the initial hidden representation of the cellular state. Each conditional residual block injects the perturbation–time context through multi-head cross-attention (Vaswani et al., 2017). For the l-th block, condition tokens are generated by a learned projection,

$$C_l = \mathrm{reshape}_{K \times d_h}\left(W_l^C\,\mathrm{LN}_c(c_{p,t}) + b_l^C\right), \tag{13}$$

Here K is the number of learned condition tokens and d_h is the hidden width. Tokens are projected from the composed perturbation–time context; they are not one token per perturbed gene. The normalized state supplies one query, $q_l = \mathrm{LN}_q(h_l) \in \mathbb{R}^{1 \times d_h}$. With H heads and head width $d_a = d_h/H$, head a uses

$$\begin{aligned} Q_l^{(a)} &= q_l W_{Q,l}^{(a)} + b_{Q,l}^{(a)}, \\ K_l^{(a)} &= C_l W_{K,l}^{(a)} + \mathbf{1}_K b_{K,l}^{(a)}, \\ V_l^{(a)} &= C_l W_{V,l}^{(a)} + \mathbf{1}_K b_{V,l}^{(a)}, \\ U_l^{(a)} &= \mathrm{Dropout}\left[\mathrm{softmax}\left(\frac{Q_l^{(a)} K_l^{(a)\top}}{\sqrt{d_a}}\right)\right] V_l^{(a)}, \\ A_l &= \mathrm{Concat}_{a=1}^{H}(U_l^{(a)}) W_{O,l} + b_{O,l}. \end{aligned} \tag{14}$$

The softmax is taken over the K condition tokens. Each head therefore produces a state-dependent mixture of values, and the output projection combines the heads (Vaswani et al., 2017). Query normalization has no learned affine parameters; the condition, attention-output, and residual-output normalizations do. Attention dropout is active during training and disabled for evaluation. The implementation defaults to $H = 4$ heads and $K = 4$ tokens, with d_h divisible by H.

The attention output is integrated through a gated residual update followed by the block-wise feed-forward transformation,

$$r_l = h_l + g_l \,\mathrm{LN}_a(A_l), \qquad h_{l+1} = \frac{\mathrm{LN}_o(r_l) + \mathrm{MLP}_l(\mathrm{LN}_o(r_l))}{\sqrt{2}}, \tag{15}$$

The final velocity combines state-dependent dynamics with a direct perturbation-dependent component:

$$v_\theta(x_t, t, x^c, e_p) = v_{\mathrm{state}}(h_L) + \alpha \, v_{\mathrm{pert}}(e_p), \tag{16}$$

The block MLP expands d_h to $4d_h$, applies SiLU and dropout, and projects back to d_h. The attention gates g_l and direct-velocity gate α are learned scalars initialized to 0.1. The state head applies layer normalization, SiLU, and a linear map to expression space; the perturbation head applies layer normalization and a two-layer SiLU MLP to the condition embedding.

Accordingly, the knowledge-derived perturbation representation does not act only at the output layer; it is injected throughout the velocity network through stacked cross-attention conditioning blocks and contributes to the learned perturbation-specific velocity field.

C.3 Mini-Batch OT Pairing

Control and perturbed cells in single-cell experiments are independently observed populations rather than paired measurements of the same cells. Optimal transport provides a natural geometric formulation for establishing relationships between such empirical distributions and has been extensively applied to cellular trajectory and perturbation modeling (Schiebinger et al., 2019; Bunne et al., 2023).

For a given perturbation condition, we sample two mini-batches of equal size,

$$C_p = \{c_i\}_{i=1}^{B}, \qquad Y_p = \{y_j\}_{j=1}^{B},$$

and define the squared Euclidean transport cost

$$D_{ij} = \|c_i - y_j\|_2^2. \tag{17}$$

The mini-batch coupling is obtained by solving the equal-mass discrete assignment problem

$$P^* = \arg\min_{P \in \mathfrak{S}_B} \sum_{i=1}^{B} D_{i,P(i)}, \tag{18}$$

where $\mathfrak{S}_B$ denotes the set of permutations of B elements.

The corresponding empirical transport plan is

$$\pi^*_{ij} = \frac{1}{B}\mathbb{I}[j = P^*(i)], \qquad (x^c, x^p) \sim \pi^*_p. \tag{19}$$

The endpoints used to define Flow Matching trajectories are therefore determined by perturbation-specific optimal-transport geometry. This construction is closely related to the use of minibatch optimal transport for defining direct conditional transport paths in OT-conditioned Flow Matching (Tong et al., 2024).

C.4 Conditional Paths

Flow Matching trains continuous generative models by directly regressing the vector field associated with a prescribed probability path, providing a simulation-free objective for local velocity learning (Lipman et al., 2023).

For an OT-coupled pair (x^c, x^p), we first construct a stochastic source state around the control cell:

$$x_0 = x^c + \sigma_s \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_G). \tag{20}$$

Equivalently,

$$q_0(x_0 \mid x^c) = \mathcal{N}\left(x^c, \sigma_s^2 I_G\right).$$

For a randomly sampled time $t \sim \mathcal{U}(0,1)$, we define a linear conditional path between the source and OT-coupled target,

$$x_t = (1-t)x_0 + t x^p. \tag{21}$$

Its corresponding conditional target velocity is

$$u_t = \frac{dx_t}{dt} = x^p - x_0. \tag{22}$$

The knowledge-conditioned Flow Matching objective is then

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\left\| v_\theta(x_t, t, x^c, e_p) - (x^p - x_0) \right\|_2^2\right], \tag{23}$$

where the expectation is taken over perturbation conditions, OT-coupled endpoints, source noise, and random time.

The resulting velocity model integrates three complementary sources of information:

$$\text{cellular state} + \text{control context} + \text{perturbation knowledge} \longrightarrow \text{conditional dynamics}.$$

C.5 Teacher Supervision

Beyond local velocity matching, we derive an implied endpoint directly from any intermediate state. For a sampled time t, the one-step teacher endpoint is defined as

$$\tilde{x}_1 = x_t + (1-t) v_\theta(x_t, t, x^c, e_p), \qquad \tilde{x}^p = \rho(\tilde{x}_1), \tag{24}$$

where $\rho(\cdot)$ denotes the non-negativity mapping in expression space. The implied-endpoint identity is derived in Appendix D.4.

For perturbation p, the observed population-level transcriptional shift and its teacher-predicted counterpart are

$$\Delta_p = \bar{x}^p - \bar{x}^c, \qquad \tilde{\Delta}_p = \frac{1}{B}\sum_{i=1}^{B} \tilde{x}_i^p - \frac{1}{B}\sum_{i=1}^{B} x_i^c. \tag{25}$$

To normalize perturbations with different effect magnitudes, we define

$$s_p = \max\left(s_{\min}, \sqrt{\frac{1}{G}\sum_{g=1}^{G} \Delta_{p,g}^2}\right).$$

Using the Smooth-L_1 penalty ℓ_β, a scaled Huber loss (Huber, 1964), the teacher-level population objective becomes

$$\mathcal{L}_{\text{teacher-}\Delta} = \frac{1}{G}\sum_{g=1}^{G} \ell_\beta\left(\frac{\tilde{\Delta}_{p,g} - \Delta_{p,g}}{s_p}\right). \tag{26}$$

This objective directly links a local velocity estimate at an arbitrary time point to the global transcriptional effect induced by the perturbation.

C.6 Rollout Supervision

Both inference and trajectory-level supervision are defined by the same conditional velocity field. The corresponding continuous dynamical system is

$$\frac{dx(t)}{dt} = v_\theta\left(x(t), t, x^c, e_p\right), \qquad x(0) = x_0, \qquad \hat{x}^p = \rho(x(1)). \tag{27}$$

This formulation follows the continuous-depth perspective of Neural ODEs (Chen et al., 2018). In practice, standard explicit solvers such as Euler or Heun integration can be used.

After a complete rollout, the predicted population-level perturbation effect is

$$\hat{\Delta}_p = \frac{1}{B_r}\sum_{i=1}^{B_r} \hat{x}_i^p - \frac{1}{B_r}\sum_{i=1}^{B_r} x_i^c.$$

The rollout-level losses are

$$\begin{aligned} \mathcal{L}_{\text{rollout-}\Delta} &= \frac{1}{G}\sum_{g=1}^{G} \ell_\beta\left(\frac{\hat{\Delta}_{p,g} - \Delta_{p,g}}{s_p}\right), \\ \mathcal{L}_{\text{RE}} &= \frac{1}{B_r G}\sum_{i,g} \ell_\beta\left(\frac{\hat{x}_{i,g}^p - x_{i,g}^p}{d_p}\right), \end{aligned} \tag{28}$$

where d_p denotes the perturbation-specific transport scale computed from OT-coupled endpoint displacements.

The complete training objective is

$$\boxed{\mathcal{L} = \lambda_{\text{FM}}\mathcal{L}_{\text{FM}} + \lambda_{\text{T}}\mathcal{L}_{\text{teacher-}\Delta} + \lambda_{\text{R}\Delta}\mathcal{L}_{\text{rollout-}\Delta} + \lambda_{\text{RE}}\mathcal{L}_{\text{RE}}} \tag{29}$$

which establishes a hierarchy of supervision,

$$\text{local velocity} \rightarrow \text{implied perturbation effect} \rightarrow \text{complete population transport}.$$

At inference time, only the control population X^c and the target perturbation p are required. The corresponding representation e_p is obtained from the biological knowledge graph, and stochasticized control states are integrated from $t = 0$ to $t = 1$ to generate the predicted perturbed population.

D Theoretical Properties

D.1 Assignment and Discrete OT

We first establish the correspondence between the assignment procedure used during training and discrete optimal transport between equally weighted empirical distributions.

Let

$$\hat{\mu}_c = \frac{1}{B}\sum_{i=1}^{B}\delta_{c_i}, \qquad \hat{\mu}_p = \frac{1}{B}\sum_{j=1}^{B}\delta_{y_j}.$$

The set of feasible transport plans is

$$\Pi(\hat{\mu}_c, \hat{\mu}_p) = \left\{ \pi \geq 0 : \sum_j \pi_{ij} = \frac{1}{B}, \sum_i \pi_{ij} = \frac{1}{B} \right\}. \tag{30}$$

By the Birkhoff–von Neumann theorem (Peyré & Cuturi, 2019), the extreme points of the doubly stochastic polytope are permutation matrices. Since the optimal-transport objective is linear in π, an optimum can be attained at such an extreme point. Consequently, there exists a permutation P^* satisfying

$$\pi^*_{ij} = \frac{1}{B}\mathbb{I}[j = P^*(i)], \qquad P^* = \arg\min_{P\in\mathfrak{S}_B} \sum_i D_{i,P(i)}. \tag{31}$$

Hence, the assignment in Eq. (18) can be interpreted directly as an optimal Kantorovich coupling between two uniformly weighted mini-batch empirical measures.

D.2 Path Geometry

Substituting Eq. (20) into Eq. (21) gives

$$x_t = (1-t)x^c + tx^p + (1-t)\sigma_s\epsilon.$$

Conditioned on (x^c, x^p), the corresponding mean and covariance are

$$\mathbb{E}[x_t \mid x^c, x^p] = (1-t)x^c + tx^p, \qquad \mathrm{Cov}(x_t \mid x^c, x^p) = (1-t)^2\sigma_s^2 I. \tag{32}$$

Thus, stochastic variation is maximal around the source state and contracts continuously toward the target endpoint.

Differentiating the conditional path with respect to time yields

$$\frac{dx_t}{dt} = x^p - x_0 = u_t. \tag{33}$$

For each endpoint pair, the chosen displacement interpolation therefore induces a constant pair-conditional velocity. This is a central geometric property underlying the combination of optimal-transport paths and Flow Matching (Lipman et al., 2023; Tong et al., 2024).

D.3 Optimal Conditional Field

Let

$$Z = (x_t, t, x^c, e_p), \qquad U = x^p - x_0.$$

The squared Flow Matching risk is

$$\mathcal{R}(v) = \mathbb{E}\left[\|v(Z) - U\|_2^2\right].$$

The squared-error projection onto conditional expectations gives the population minimizer used in conditional Flow Matching (Lipman et al., 2023; Tong et al., 2024):

$$v^*(Z) = \mathbb{E}\left[U \mid Z\right] = \mathbb{E}\left[x^p - x_0 \mid x_t, t, x^c, e_p\right]. \tag{34}$$

Hence, whenever multiple admissible source–target trajectories pass through a common state, the network learns their conditional mean transport velocity given the current cellular state, control context, and perturbation representation.

Let $q_t(x \mid x^c, p)$ denote the corresponding conditional marginal density. The induced probability evolution satisfies the continuity equation (Lipman et al., 2023; Peyré & Cuturi, 2019)

$$\frac{\partial q_t(x \mid x^c, p)}{\partial t} + \nabla_x \cdot \left[q_t(x \mid x^c, p) v^*(x, t, x^c, e_p)\right] = 0. \tag{35}$$

Thus, the learned conditional vector field provides a continuous dynamical representation of probability transport from the source distribution toward the perturbation-conditioned target distribution.

Table 7: Single-gene Student t-tests for Table 1: best versus second-best means, including tied runners-up. Tests are two-sided with $n = 5$ per method and $df = 8$; p-values are unadjusted. Retrieval is normalized.

Metric	Best (mean)	Second (mean)	t (df)	p-value
Adamson single ($n = 22$)				
Pearson Δ	SDFM (0.8071)	TxPert (0.7385)	6.699 (8)	1.53×10^{-4}
Spearman Δ	SDFM (0.5517)	STATE (0.5325)	4.997 (8)	1.06×10^{-3}
DE20	SDFM (0.8889)	TxPert (0.7878)	4.034 (8)	3.77×10^{-3}
DE100	SDFM (0.8540)	TxPert (0.7768)	3.527 (8)	7.77×10^{-3}
Retrieval	SDFM (0.6649)	scGPT (0.6369)	3.830 (8)	5.02×10^{-3}
Fast retrieval	SDFM (0.6642)	scGPT (0.6363)	4.342 (8)	2.47×10^{-3}
HepG2 single ($n = 209$)				
Pearson Δ	SDFM (0.2634)	TxPert (0.2486)	7.904 (8)	4.76×10^{-5}
Spearman Δ	SDFM (0.2363)	AttentionPert (0.2231)	3.813 (8)	5.14×10^{-3}
Spearman Δ	SDFM (0.2363)	TxPert (0.2231)	5.273 (8)	7.52×10^{-4}
DE20	SDFM (0.3170)	AttentionPert (0.3004)	4.061 (8)	3.63×10^{-3}
DE100	SDFM (0.3153)	GEARS (0.2946)	4.635 (8)	1.68×10^{-3}
Retrieval	SDFM (0.5751)	TxPert (0.5363)	5.401 (8)	6.45×10^{-4}
Fast retrieval	SDFM (0.5855)	TxPert (0.5619)	3.774 (8)	5.43×10^{-3}
Norman single ($n = 36$)				
Pearson Δ	SDFM (0.3266)	TxPert (0.3192)	2.785 (8)	2.37×10^{-2}
Spearman Δ	SDFM (0.1587)	scGPT (0.1451)	5.848 (8)	3.84×10^{-4}
DE20	scGPT (0.3694)	SDFM (0.3681)	0.121 (8)	9.07×10^{-1}
DE100	scGPT (0.3571)	SDFM (0.3536)	0.547 (8)	5.99×10^{-1}
Retrieval	SDFM (0.5649)	TxPert (0.5470)	3.442 (8)	8.80×10^{-3}
Fast retrieval	SDFM (0.5531)	scGPT (0.5425)	2.837 (8)	2.19×10^{-2}

D.4 Teacher Consistency

The one-step teacher follows directly from the linear conditional path.

Assume that the predicted velocity at x_t exactly recovers the target velocity,

$$v_\theta(x_t, t, x^c, e_p) = u_t = x^p - x_0.$$

Then

$$\begin{aligned} \tilde{x}_1 &= x_t + (1-t)u_t \\ &= (1-t)x_0 + tx^p + (1-t)(x^p - x_0) \\ &= x^p. \end{aligned} \tag{36}$$

Table 8: Complete Norman dual-gene results by seen-gene group. Values and ranking conventions follow Table 1.

	All genes		DE genes		Retrieval	
Method	Pearson Δ	Spearman Δ	DE20	DE100	Normalized	Fast
Seen0 ($n = 9$)						
SDFM (ours)	**0.5734** ± *0.0093*	0.1934 ± *0.0057*	**0.6392** ± *0.0068*	**0.6726** ± *0.0042*	**0.6888** ± *0.0094*	**0.6697** ± *0.0086*
AttentionPert	0.5517 ± *0.0097*	0.1181 ± *0.0043*	0.6231 ± *0.0088*	0.6126 ± *0.0107*	0.6483 ± *0.0132*	0.6310 ± *0.0121*
GEARS	0.4481 ± *0.0087*	0.1650 ± *0.0051*	0.6121 ± *0.0096*	0.6030 ± *0.0089*	0.6281 ± *0.0103*	0.6366 ± *0.0098*
TxPert	0.5521 ± *0.0085*	0.1419 ± *0.0062*	0.4290 ± *0.0093*	0.5051 ± *0.0081*	0.6478 ± *0.0120*	0.6323 ± *0.0154*
scGPT	0.5278 ± *0.0164*	0.1721 ± *0.0060*	0.4995 ± *0.0172*	0.5944 ± *0.0158*	0.6079 ± *0.0186*	0.6169 ± *0.0173*
STATE	0.1224 ± *0.0053*	0.1402 ± *0.0058*	0.4020 ± *0.0161*	0.2827 ± *0.0147*	0.5447 ± *0.0205*	0.5578 ± *0.0192*
scDFM	0.5294 ± *0.0141*	**0.2044** ± *0.0048*	0.5959 ± *0.0153*	0.5720 ± *0.0136*	0.6325 ± *0.0167*	0.6235 ± *0.0154*
CPA	0.3180 ± *0.0102*	0.0678 ± *0.0037*	0.5291 ± *0.0128*	0.4466 ± *0.0089*	0.4653 ± *0.0178*	0.4709 ± *0.0165*
Seen1 ($n = 43$)						
SDFM (ours)	**0.6429** ± *0.0068*	**0.2477** ± *0.0084*	**0.6912** ± *0.0097*	**0.6815** ± *0.0082*	**0.7829** ± *0.0075*	**0.7705** ± *0.0069*
AttentionPert	0.6234 ± *0.0072*	0.2132 ± *0.0053*	0.6508 ± *0.0081*	0.6369 ± *0.0076*	0.7535 ± *0.0093*	0.7375 ± *0.0087*
GEARS	0.5551 ± *0.0114*	0.1863 ± *0.0047*	0.6628 ± *0.0108*	0.6468 ± *0.0095*	0.7577 ± *0.0121*	0.7441 ± *0.0112*
TxPert	0.6028 ± *0.0059*	0.2044 ± *0.0061*	0.6193 ± *0.0073*	0.6064 ± *0.0066*	0.7258 ± *0.0084*	0.7334 ± *0.0079*
scGPT	0.5967 ± *0.0063*	0.1978 ± *0.0052*	0.6476 ± *0.0088*	0.6438 ± *0.0074*	0.7131 ± *0.0067*	0.7206 ± *0.0062*
STATE	0.1159 ± *0.0042*	0.1206 ± *0.0056*	0.2313 ± *0.0140*	0.2175 ± *0.0128*	0.5323 ± *0.0153*	0.5444 ± *0.0146*
scDFM	0.4707 ± *0.0098*	0.1793 ± *0.0044*	0.5915 ± *0.0117*	0.5562 ± *0.0103*	0.6439 ± *0.0126*	0.6245 ± *0.0110*
CPA	0.3181 ± *0.0076*	0.0907 ± *0.0038*	0.4369 ± *0.0092*	0.3743 ± *0.0085*	0.6227 ± *0.0130*	0.6099 ± *0.0127*
Seen2 ($n = 19$)						
SDFM (ours)	**0.8221** ± *0.0143*	**0.3171** ± *0.0127*	**0.8287** ± *0.0105*	**0.8387** ± *0.0076*	**0.9302** ± *0.0098*	**0.9205** ± *0.0087*
AttentionPert	0.7226 ± *0.0086*	0.2504 ± *0.0054*	0.8065 ± *0.0079*	0.8017 ± *0.0068*	0.9001 ± *0.0104*	0.8870 ± *0.0096*
GEARS	0.6280 ± *0.0135*	0.2289 ± *0.0049*	0.7726 ± *0.0112*	0.7188 ± *0.0097*	0.8156 ± *0.0128*	0.7999 ± *0.0110*
TxPert	0.8018 ± *0.0078*	0.2936 ± *0.0057*	0.8087 ± *0.0083*	0.7997 ± *0.0072*	0.9124 ± *0.0091*	0.9023 ± *0.0084*
scGPT	0.7772 ± *0.0092*	0.2827 ± *0.0051*	0.7810 ± *0.0087*	0.7745 ± *0.0076*	0.9108 ± *0.0073*	0.9027 ± *0.0068*
STATE	0.1134 ± *0.0041*	0.1665 ± *0.0055*	0.3112 ± *0.0157*	0.2299 ± *0.0134*	0.5404 ± *0.0168*	0.5545 ± *0.0152*
scDFM	0.5480 ± *0.0118*	0.2036 ± *0.0046*	0.6294 ± *0.0124*	0.5971 ± *0.0079*	0.7391 ± *0.0141*	0.7295 ± *0.0132*
CPA	0.3294 ± *0.0089*	0.1122 ± *0.0039*	0.4362 ± *0.0106*	0.3509 ± *0.0094*	0.7467 ± *0.0155*	0.7396 ± *0.0147*

More generally, the endpoint prediction error is exactly related to the local velocity error by

$$\tilde{x}_1 - x^p = (1-t)\left[v_\theta(x_t, t, x^c, e_p) - u_t\right]. \tag{37}$$

The teacher endpoint is therefore not a separate prediction mechanism, but the natural extrapolation implied by the current velocity estimate over the remaining time interval. The teacher-Δ objective consequently projects local velocity information directly onto the population-level perturbation effect.

Table 9: Dual-gene Student t-tests by seen-gene group: best versus second-best means in Table 8. Tests are two-sided with $n = 5$ per method and $df = 8$; p-values are unadjusted. Retrieval is normalized.

Metric	Best (mean)	Second (mean)	t (df)	p-value
Seen0 ($n = 9$)				
Pearson Δ	SDFM (0.5734)	TxPert (0.5521)	3.780 (8)	5.39×10^{-3}
Spearman Δ	scDFM (0.2044)	SDFM (0.1934)	3.301 (8)	1.08×10^{-2}
DE20	SDFM (0.6392)	AttentionPert (0.6231)	3.237 (8)	1.19×10^{-2}
DE100	SDFM (0.6726)	AttentionPert (0.6126)	11.672 (8)	2.65×10^{-6}
Retrieval	SDFM (0.6888)	AttentionPert (0.6483)	5.588 (8)	5.17×10^{-4}
Fast retrieval	SDFM (0.6697)	GEARS (0.6366)	5.677 (8)	4.67×10^{-4}
Seen1 ($n = 43$)				
Pearson Δ	SDFM (0.6429)	AttentionPert (0.6234)	4.403 (8)	2.28×10^{-3}
Spearman Δ	SDFM (0.2477)	AttentionPert (0.2132)	7.767 (8)	5.40×10^{-5}
DE20	SDFM (0.6912)	GEARS (0.6628)	4.375 (8)	2.37×10^{-3}
DE100	SDFM (0.6815)	GEARS (0.6468)	6.183 (8)	2.64×10^{-4}

Continued on next page

Table 9 continued

Metric	Best (mean)	Second (mean)	t (df)	p-value
Retrieval	SDFM (0.7829)	GEARS (0.7577)	3.958 (8)	4.19×10^{-3}
Fast retrieval	SDFM (0.7705)	GEARS (0.7441)	4.487 (8)	2.04×10^{-3}
Seen2 ($n = 19$)				
Pearson Δ	SDFM (0.8221)	TxPert (0.8018)	2.787 (8)	2.37×10^{-2}
Spearman Δ	SDFM (0.3171)	TxPert (0.2936)	3.775 (8)	5.43×10^{-3}
DE20	SDFM (0.8287)	TxPert (0.8087)	3.341 (8)	1.02×10^{-2}
DE100	SDFM (0.8387)	AttentionPert (0.8017)	8.113 (8)	3.95×10^{-5}
Retrieval	SDFM (0.9302)	TxPert (0.9124)	2.976 (8)	1.77×10^{-2}
Fast retrieval	SDFM (0.9205)	scGPT (0.9027)	3.605 (8)	6.94×10^{-3}

D.5 Population and Set Consistency

The true and predicted population-level perturbation effects are

$$\Delta_p = \mathbb{E}_{\mu_p}[x] - \mathbb{E}_{\mu_c}[x], \qquad \hat{\Delta}_p = \mathbb{E}_{\hat{\mu}_p}[x] - \mathbb{E}_{\mu_c}[x].$$

If the learned transport recovers the target distribution, $\hat{\mu}_p = \mu_p$, then

$$\hat{\Delta}_p = \Delta_p. \tag{38}$$

Thus, the rollout-Δ objective directly constrains a first-order statistic of the generated perturbation distribution and is aligned with the population transport objective itself.

The composer in Eq. (11) is permutation invariant (Zaheer et al., 2017). Swapping either member of a pair leaves z_{ij} unchanged, and any permutation of the genes merely reorders the terms in the single-gene sum and unordered-pair average. Thus, for any permutation σ,

$$\mathrm{Compose}_\psi(e_1, \ldots, e_M) = \mathrm{Compose}_\psi(e_{\sigma(1)}, \ldots, e_{\sigma(M)}). \tag{39}$$

The representation therefore captures learned pairwise context without depending on gene order. For $M = 1$ it equals e_1; disabling the interaction residual yields $\sum_i e_i$. These properties hold throughout the shared training and inference composition path.

D.6 Numerical Integration

At inference time, the model numerically solves

$$\dot{x}(t) = v_\theta(x(t), t, x^c, e_p).$$

Assume that v_θ is locally Lipschitz in x, sufficiently smooth along the trajectory, and has a solution throughout $[0, 1]$. The solution is then unique, and standard numerical integration estimates apply (Hairer et al., 1993; Chen et al., 2018). For maximum integration step size h, the global discretization errors of first-order Euler integration and second-order Heun integration satisfy

$$\max_k \|x(t_k) - x_k^{\mathrm{Euler}}\| = \mathcal{O}(h), \qquad \max_k \|x(t_k) - x_k^{\mathrm{Heun}}\| = \mathcal{O}(h^2). \tag{40}$$

Under the ideal pair-conditioned velocity

$$v(x, t) = x^p - x_0,$$

the ODE admits the closed-form solution

$$x(t) = x_0 + t(x^p - x_0), \qquad x(1) = x^p. \tag{41}$$

Table 10: Population distribution errors: SDFM and MeanShift-Delta. Values are condition means; lower is better.

Dataset	Method	SWD ↓	Energy ↓	Covariance error ↓
Adamson	SDFM	**1.1995**	**2.4960**	**0.7588**
	MeanShift-Delta	2.2063	11.6001	0.8518
HepG2	SDFM	**1.6691**	**2.3995**	**0.7436**
	MeanShift-Delta	1.7176	2.6500	0.7596
Norman	SDFM	**1.8767**	**2.7611**	**0.3390**
	MeanShift-Delta	2.2522	5.5672	0.4259

Therefore, the local Flow Matching target, the one-step teacher endpoint, and the complete ODE rollout share the same endpoint under the ideal vector field.

The overall learning process can consequently be summarized as

$$\boxed{\text{OT coupling} \rightarrow \text{conditional path} \rightarrow \text{local velocity} \rightarrow \text{population effect} \rightarrow \text{generated distribution}}$$

where biological knowledge specifies the perturbation condition, optimal transport defines the population geometry, Flow Matching learns the continuous dynamics, and teacher- and rollout-level objectives connect local dynamics to the final perturbation response.

E Supplementary Results

E.1 Benchmark Performance

Table 1 summarizes single-gene prediction across datasets. SDFM led all-gene Pearson Δ on Adamson, HepG2, and Norman, with Student t-test comparisons against TxPert giving $p = 1.53 \times 10^{-4}$, $p = 4.76 \times 10^{-5}$, and $p = 2.37 \times 10^{-2}$, respectively (Table 7). Its largest absolute improvement was on Adamson, from 0.7385 to 0.8071.

Perturbation retrieval evaluates whether a predicted response identifies the matching intervention. SDFM achieved normalized retrieval scores of 0.6649, 0.5751, and 0.5649 on Adamson, HepG2, and Norman, respectively; its corresponding fast retrieval scores were 0.6642, 0.5855, and 0.5531. All six comparisons with the respective second-best methods were significant (Tables 1 and 7), indicating that the improvements in response correlation were accompanied by stronger condition discrimination.

DE-gene prediction varied across datasets. On Adamson, SDFM led DE20 and DE100 at 0.8889 and 0.8540, compared with 0.7878 and 0.7768 for TxPert ($p = 3.77 \times 10^{-3}$ and $p = 7.77 \times 10^{-3}$). On HepG2, SDFM reached 0.3170 for DE20 and 0.3153 for DE100, exceeding AttentionPert at 0.3004 and GEARS at 0.2946 ($p = 3.63 \times 10^{-3}$ and $p = 1.68 \times 10^{-3}$). On Norman, scGPT and SDFM were close on DE20 (0.3694 vs. 0.3681; $p = 9.07 \times 10^{-1}$) and DE100 (0.3571 vs. 0.3536; $p = 5.99 \times 10^{-1}$). The strongest DE-gene gains therefore occurred on Adamson and HepG2.

Table 8 provides complete stratified results for all baselines, and Table 9 reports the six-metric comparisons within each seen-gene group. The condition-weighted overall results in Table 2 place TxPert second for Pearson Δ, Spearman Δ, and fast retrieval, and AttentionPert second for DE20, DE100, and normalized retrieval. In seen0, SDFM led five metrics, while scDFM had higher Spearman Δ ($p = 1.08 \times 10^{-2}$). All six comparisons favored SDFM in seen1 and seen2. In seen2, the DE20 advantage over TxPert corresponded to $p = 1.02 \times 10^{-2}$, and the DE100 advantage over AttentionPert to $p = 3.95 \times 10^{-5}$.

E.2 Distribution Recovery

Following distribution-level evaluations of cellular perturbations (Bunne et al., 2023; Yu et al., 2026), we compared generated and true populations at the perturbation-condition level, covering

Table 11: Population distribution errors across SDFM and generative baselines. Values are condition means; lower is better. Best per dataset is bold; second-best is underlined.

Dataset	Method	SWD ↓	Energy ↓	Covariance error ↓
Adamson	**SDFM (ours)**	**1.1995**	**2.4960**	**0.7588**
	CellFlow	<u>1.4236</u>	<u>3.0427</u>	<u>0.7802</u>
	scDFM	1.4572	3.2289	0.7864
	TxPert	1.5813	3.8715	0.8017
	AttentionPert	1.6198	4.1153	0.8092
	MeanShift-Delta	2.2063	11.6001	0.8518
HepG2	**SDFM (ours)**	**1.6691**	**2.3995**	**0.7436**
	TxPert	<u>1.6987</u>	<u>2.5037</u>	<u>0.7513</u>
	AttentionPert	1.7042	2.5324	0.7536
	scDFM	1.7125	2.5571	0.7548
	CellFlow	1.7218	2.6035	0.7584
	MeanShift-Delta	1.7176	2.6500	0.7596
Norman	**SDFM (ours)**	**1.8767**	**2.7611**	**0.3390**
	scDFM	<u>1.9538</u>	<u>3.1072</u>	<u>0.3649</u>
	CellFlow	1.9817	3.2048	0.3718
	TxPert	2.0235	3.3506	0.3776
	AttentionPert	2.0869	3.6123	0.3908
	MeanShift-Delta	2.2522	5.5672	0.4259

SWD, sliced Wasserstein distance; Energy, energy distance; Covariance error, normalized covariance error. All metrics are computed in the training-fitted PCA space with balanced subsampling following Appendix B.3.

22, 583, and 107 conditions from Adamson, HepG2, and Norman, respectively. The training-fitted PCA space, repeated balanced subsampling, and condition-level aggregation are described in Appendix B.3.

SDFM reduced all three distributional errors relative to MeanShift-Delta (Table 10). SWD decreased by 45.6%, 2.8%, and 16.7% on Adamson, HepG2, and Norman, respectively. Energy distance decreased from 11.6001 to 2.4960 on Adamson and from 5.5672 to 2.7611 on Norman, linking the visual population comparisons to condition-averaged improvements.

The covariance comparisons also favored SDFM. Normalized covariance error decreased from 0.8518 to 0.7588 on Adamson, from 0.7596 to 0.7436 on HepG2, and from 0.4259 to 0.3390 on Norman. Together with SWD and energy distance, these results describe improved agreement in population location, spread, and dependence.

Figures 6–8 show population shapes corresponding to these quantitative results. The differences between predicted and true populations appear both as shifts in response location and as changes in distribution width and contour. Together with the cross-condition statistics, these figures connect mean response prediction with cell population recovery.

Table 11 extends the comparison to additional generative baselines under the same distributional evaluation.

E.3 Component Analysis

Table 3 reports single-gene ablations, and Table 12 reports dual-gene ablations. Tables **??** and **??** compare Full SDFM with the best remaining variant on the single-gene benchmarks and within each dual-gene seen-gene group, respectively. The experiments examine biological representations, OT pairing, trajectory supervision, and conditioning.

Perturbation representation and conditioning. Replacing the graph encoder with Emb-MLP reduced Pearson Δ from 0.8071, 0.2634, and 0.3266 to 0.7079, 0.2269, and 0.2790 on Adamson, HepG2, and Norman, respectively. On Adamson and Norman, GATv2 (Brody et al., 2022) and the dual-branch encoder retained higher correlations than Emb-MLP. The strongest remaining single-gene variants were w/o STRING on Adamson, w/o GO on HepG2, and GATv2 on Norman; their

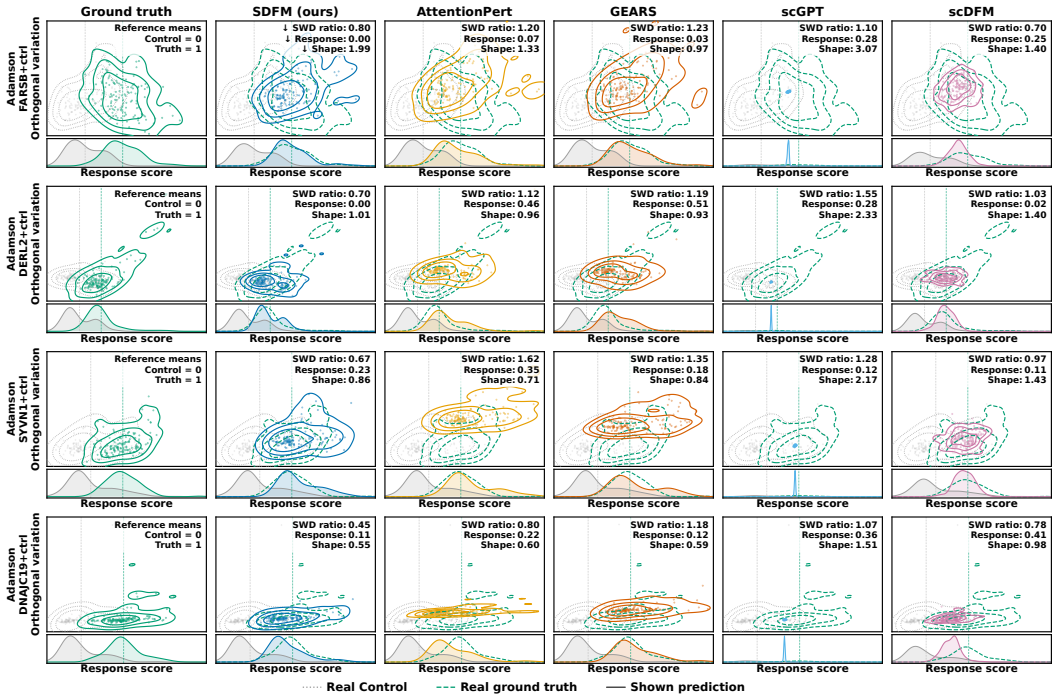

Figure 6: Population response comparisons on Adamson for FARSB, DERL2, SYVN1, and DNAJC19. Columns show observed populations and five prediction methods. Contours, marginal densities, and SWD-ratio annotations follow Figure 5.

comparisons with Full SDFM gave $p = 4.84 \times 10^{-3}$, $p = 2.07 \times 10^{-2}$, and $p = 2.85 \times 10^{-3}$. Replacing cross-attention with concatenation also reduced Pearson Δ on all three datasets (Tables 3 and **??**).

OT pairing. After removing OT, Adamson's Pearson Δ decreased from 0.8071 to 0.7391, and Norman single from 0.3266 to 0.2953; the overall dual-perturbation result decreased from 0.6820 to 0.5866. With entropy-regularized Sinkhorn pairing (Cuturi, 2013), Adamson's Pearson Δ was 0.7423, lower than the full model's 0.8071 (Tables 3 and 12). Under these settings, the pairing between source and target populations clearly affected subsequent velocity field learning.

Population and trajectory supervision. Removing teacher-Δ reduced Pearson Δ to 0.7387 on Adamson and 0.2305 on HepG2. Removing the rollout endpoint loss reduced Norman single-gene Pearson Δ from 0.3266 to 0.3051. On the overall dual-gene task, removing rollout-Δ reduced DE100 from 0.7224 to 0.6679 and normalized retrieval from 0.8104 to 0.7575 (Table 12).

The contribution of population supervision extended across seen-gene groups (Table **??**). In seen0, Full SDFM reached Pearson $\Delta = 0.5734$, compared with 0.5132 after removing teacher-Δ ($p = 8.38 \times 10^{-6}$). Removing rollout-Δ reduced DE100 from 0.6815 to 0.6322 in seen1 and from 0.8387 to 0.7826 in seen2 ($p = 1.24 \times 10^{-5}$ and $p = 5.50 \times 10^{-5}$). Together, the ablations support supervision that connects implied population shifts with complete generated trajectories.

E.4 Deterministic Controls

Motivated by the importance of informative baselines in perturbation benchmarking (Wu et al., 2025; Wei et al., 2026), we compared SDFM, MeanShift-Delta, and a Δ-supervised deterministic predictor. MeanShift-Delta predicts a uniform shift from the perturbation representation; the deterministic predictor additionally incorporates control cell states, generates cell-dependent expression residuals, and jointly optimizes reconstruction loss and population Δ loss.

Adding cell states substantially improved the deterministic predictor over the uniform-shift model (Table 14). On Adamson, Pearson Δ increased from 0.5055 to 0.7777, close to SDFM's 0.7823,

Table 12: Norman dual-gene ablations. Overall values are condition-weighted means; subgroup values are mean ± SD. Bold and underlined values mark the best and second-best variants. Retrieval is normalized.

Variant	Pearson Δ	DE100	Retrieval
Norman double overall ($n = 71$)			
Full SDFM	**0.6820**	**0.7224**	**0.8104**
Emb-MLP	0.5931	0.5996	0.7066
w/o OT	0.5866	0.6360	0.7192
w/o teacher-Δ	0.6140	0.6474	0.7481
w/o rollout-Δ	<u>0.6254</u>	<u>0.6679</u>	<u>0.7575</u>
MSE aux.	0.6103	0.6310	0.7334
Norman seen0 ($n = 9$)			
Full SDFM	**0.5734** ± *0.0093*	**0.6726** ± *0.0042*	**0.6888** ± *0.0094*
Emb-MLP	0.4928 ± *0.0117*	0.5349 ± *0.0062*	0.5815 ± *0.0074*
w/o OT	0.4802 ± *0.0111*	0.5859 ± *0.0132*	0.6017 ± *0.0047*
w/o teacher-Δ	<u>0.5132</u> ± *0.0097*	<u>0.6002</u> ± *0.0139*	<u>0.6272</u> ± *0.0114*
w/o rollout-Δ	0.5098 ± *0.0141*	0.5965 ± *0.0224*	0.6235 ± *0.0220*
MSE aux.	0.5095 ± *0.0049*	0.5857 ± *0.0068*	0.6221 ± *0.0091*
Norman seen1 ($n = 43$)			
Full SDFM	**0.6429** ± *0.0068*	**0.6815** ± *0.0082*	**0.7829** ± *0.0075*
Emb-MLP	0.5558 ± *0.0067*	0.5651 ± *0.0086*	0.6782 ± *0.0152*
w/o OT	0.5487 ± *0.0130*	0.5921 ± *0.0038*	0.6905 ± *0.0103*
w/o teacher-Δ	0.5772 ± *0.0110*	0.6169 ± *0.0122*	0.7230 ± *0.0094*
w/o rollout-Δ	<u>0.5842</u> ± *0.0123*	<u>0.6322</u> ± *0.0082*	<u>0.7320</u> ± *0.0247*
MSE aux.	0.5746 ± *0.0088*	0.5947 ± *0.0061*	0.7081 ± *0.0165*
Norman seen2 ($n = 19$)			
Full SDFM	**0.8221** ± *0.0143*	**0.8387** ± *0.0076*	**0.9302** ± *0.0098*
Emb-MLP	0.7250 ± *0.0093*	0.7084 ± *0.0183*	0.8301 ± *0.0079*
w/o OT	0.7228 ± *0.0220*	0.7589 ± *0.0109*	0.8397 ± *0.0217*
w/o teacher-Δ	0.7449 ± *0.0119*	0.7388 ± *0.0099*	0.8622 ± *0.0101*
w/o rollout-Δ	<u>0.7735</u> ± *0.0185*	<u>0.7826</u> ± *0.0143*	<u>0.8788</u> ± *0.0241*
MSE aux.	0.7388 ± *0.0098*	0.7345 ± *0.0118*	0.8433 ± *0.0202*

Table 13: Difference between Full SDFM and w/o Inter-Aware. Δ = Full SDFM − w/o Inter-Aware. Positive values indicate that removing Inter-Aware decreases performance. Subgroup SDs are not differenced; Δ is computed from means only.

Subset	Metric	Full SDFM	w/o Inter-Aware	Δ
Norman double overall ($n = 71$)				
	Pearson	0.6820	0.6667	0.0153
	DE100	0.7224	0.7071	0.0153
	Retrieval	0.8104	0.7943	0.0161
Norman seen0 ($n = 9$)				
	Pearson	0.5734 ± *0.0093*	0.5587 ± *0.0120*	0.0147
	DE100	0.6726 ± *0.0042*	0.6549 ± *0.0080*	0.0177
	Retrieval	0.6888 ± *0.0094*	0.6765 ± *0.0130*	0.0123
Norman seen1 ($n = 43$)				
	Pearson	0.6429 ± *0.0068*	0.6263 ± *0.0090*	0.0166
	DE100	0.6815 ± *0.0082*	0.6671 ± *0.0100*	0.0144
	Retrieval	0.7829 ± *0.0075*	0.7636 ± *0.0120*	0.0193
Norman seen2 ($n = 19$)				
	Pearson	0.8221 ± *0.0143*	0.8094 ± *0.0130*	0.0127
	DE100	0.8387 ± *0.0076*	0.8225 ± *0.0110*	0.0162
	Retrieval	0.9302 ± *0.0098*	0.9196 ± *0.0120*	0.0106

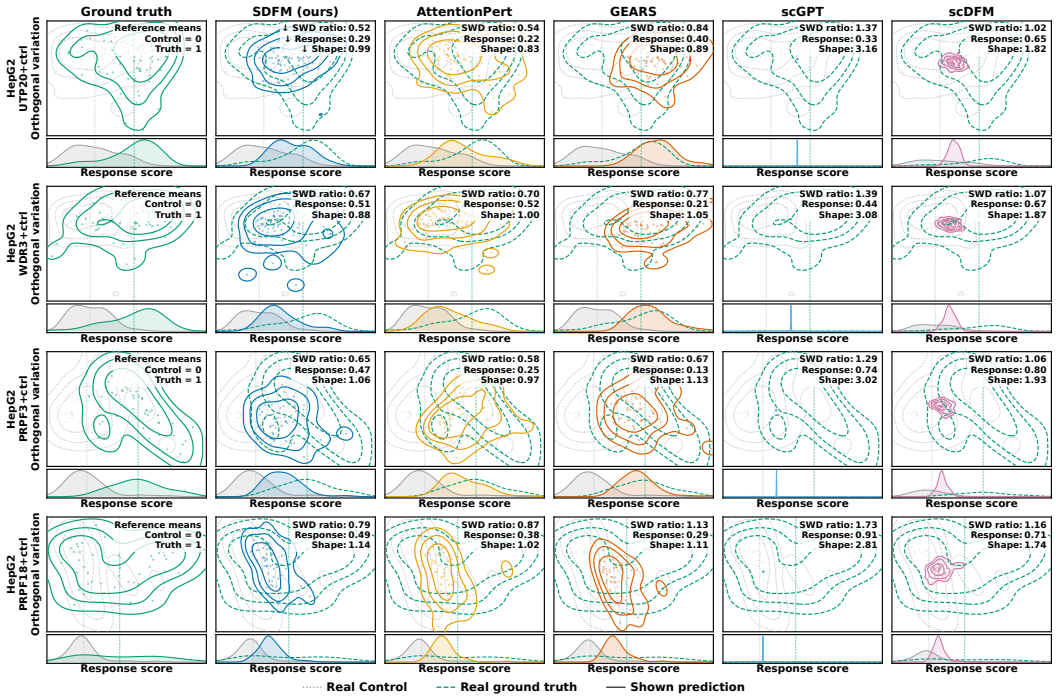

Figure 7: Population response comparisons on HepG2 for UTP20, WDR3, PRPF3, and PRPF18. Columns show observed populations and five prediction methods. Contours, marginal densities, and SWD-ratio annotations follow Figure 5.

Table 14: Deterministic controls under population Δ supervision.

Dataset	Method	Pearson $\Delta \uparrow$	DE100 $\uparrow$
Adamson	SDFM	**0.7823**	**0.8672**
	MeanShift-Delta	0.5055	0.5333
	Deterministic-Δ	0.7777	0.8326
HepG2	SDFM	**0.2726**	**0.3673**
	MeanShift-Delta	0.2277	0.3132
	Deterministic-Δ	0.2659	0.3526
Norman	SDFM	**0.5962**	**0.6787**
	MeanShift-Delta	0.3780	0.4430
	Deterministic-Δ	0.5599	0.6396

Deterministic-Δ denotes the cell-state predictor with reconstruction and population-Δ supervision. Values are means across inference seeds; DE100 is the Pearson Δ on the corresponding gene subset.

while SDFM retained a larger advantage on DE100 (0.8672 vs. 0.8326). On HepG2, SDFM reached 0.2726 for Pearson Δ and 0.3673 for DE100, compared with 0.2659 and 0.3526 for the deterministic predictor. On Norman, the corresponding comparisons were 0.5962 vs. 0.5599 and 0.6787 vs. 0.6396.

Cell-state inputs and population supervision thus provide a strong deterministic baseline, while SDFM improves both the all-gene response profile and DE100 prediction across the three datasets.

E.5 Graph Connectivity

We examined the role of knowledge graph connectivity through a degree-preserving randomization control. STRING and GO were rewired using directed double-edge swaps, following degree-preserving network randomization (Maslov & Sneppen, 2002), preserving the node set, node fea-

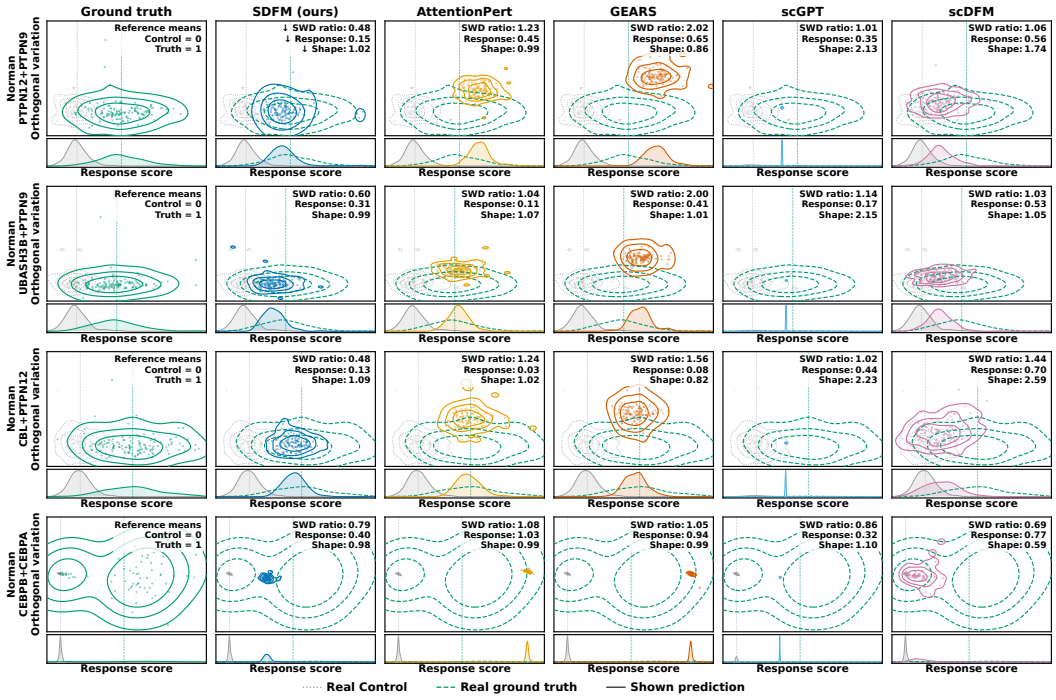

Figure 8: Population response comparisons on Norman for PTPN12+PTPN9, UBASH3B+PTPN9, CBL+PTPN12, and CEBPB+CEBPA. Columns show observed populations and five prediction methods. Contours, marginal densities, and SWD-ratio annotations follow Figure 5.

Table 15: Real versus degree-preserving randomized biological graphs.

	Pearson Δ		Retrieval	
Dataset	Real-KG	Shuffled-KG	Real-KG	Shuffled-KG
Adamson	**0.7823**	0.6757	**0.6438**	0.4999
HepG2	**0.2726**	0.2263	**0.5883**	0.4997
Norman	**0.5962**	0.5637	**0.7326**	0.7296

Retrieval is normalized retrieval; all metrics are higher-is-better, and bold indicates a higher mean.

tures, and the in-degree and out-degree of each node, while using the same model and training configuration.

Real-KG achieved higher means for both metrics on all three datasets (Table 15). Pearson Δ was 0.7823 vs. 0.6757 on Adamson, 0.2726 vs. 0.2263 on HepG2, and 0.5962 vs. 0.5637 on Norman. Normalized retrieval showed larger differences on Adamson (0.6438 vs. 0.4999) and HepG2 (0.5883 vs. 0.4997), and a smaller difference on Norman (0.7326 vs. 0.7296).

These comparisons complement the knowledge-source ablations: the organization of biological connections contributes to response prediction, while its effect on perturbation retrieval varies across datasets.

E.6 Extended Cases

Figures 9–11 compare predicted and observed mean expression changes for additional conditions. On Adamson, SDFM reached Pearson Δ values of 0.930 for FARSB, 0.934 for DERL2, and 0.949 for DNAJC19. On SYVN1, STATE was slightly higher than SDFM (0.933 vs. 0.922; Figure 9), illustrating variation across individual interventions.

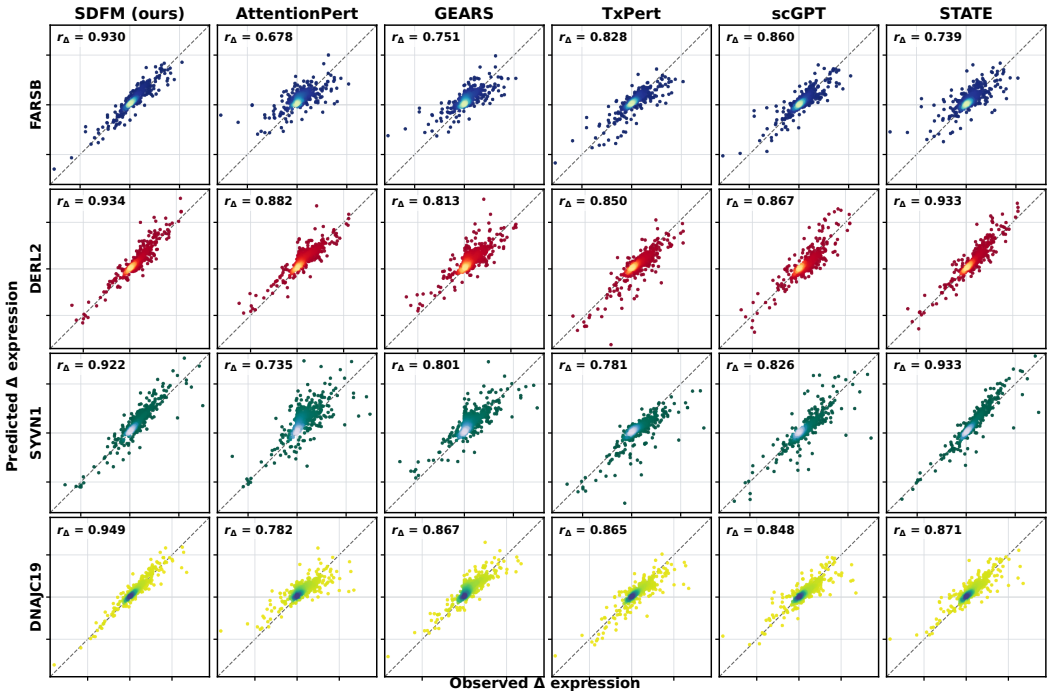

Figure 9: Mean expression changes on Adamson for FARSB, DERL2, SYVN1, and DNAJC19. Points are genes; columns compare SDFM with five baselines, and annotations report Pearson Δ.

On HepG2, scGPT had higher correlations on UTP20, WDR3, and PRPF3, whereas SDFM led PRPF18 at 0.755 (Figure 10). These contrasts show that aggregate performance combines interventions with different response profiles and different method rankings.

Mean-response correlation and population recovery provide complementary views. For CBL+PTPN12 on Norman, SDFM and scGPT had Pearson Δ values of 0.987 and 0.957, while their SWD ratios were 0.48 and 1.02 (Figures 11 and 8). Both methods recovered the mean response direction, but their generated populations differed in distributional fit.

Figures 6–8 also show distributional changes across datasets. Under the DNAJC19 condition on Adamson (Figure 6), SDFM's SWD ratio was 0.45, lower than the other methods in the figure; under the UTP20 condition on HepG2 (Figure 7), SDFM and AttentionPert gave 0.52 and 0.54, respectively. On the other hand, for FARSB on Adamson and CEBPB+CEBPA on Norman, scDFM obtained lower SWD ratios. Combining these cases with the cross-condition statistics in Table 10 allows simultaneous observation of overall model performance and condition-specific response differences.

E.7 Additional Pairwise Response Landscapes

Figure 3 summarizes the response-space geometry of observed perturbations and an expanded set of pairwise responses. The interpretation of the pairwise responses differs across datasets. Norman contains experimentally measured paired perturbations and is therefore used for quantitative evaluation of combinatorial prediction and interaction-aware composition in the main text. Adamson and HepG2 contain single-gene perturbation measurements only; their pairwise landscapes are included here as exploratory extensions of the learned single-gene response space.

For Adamson and HepG2, we deliberately keep this exploratory construction separate from the learned interaction-aware composition. For an unassayed pair of genes A and B, we construct an additive virtual response from the independently predicted single-gene expression shifts,

$$\hat{\Delta}^{\mathrm{add}}_{A+B} = \hat{\Delta}_A + \hat{\Delta}_B. \tag{42}$$

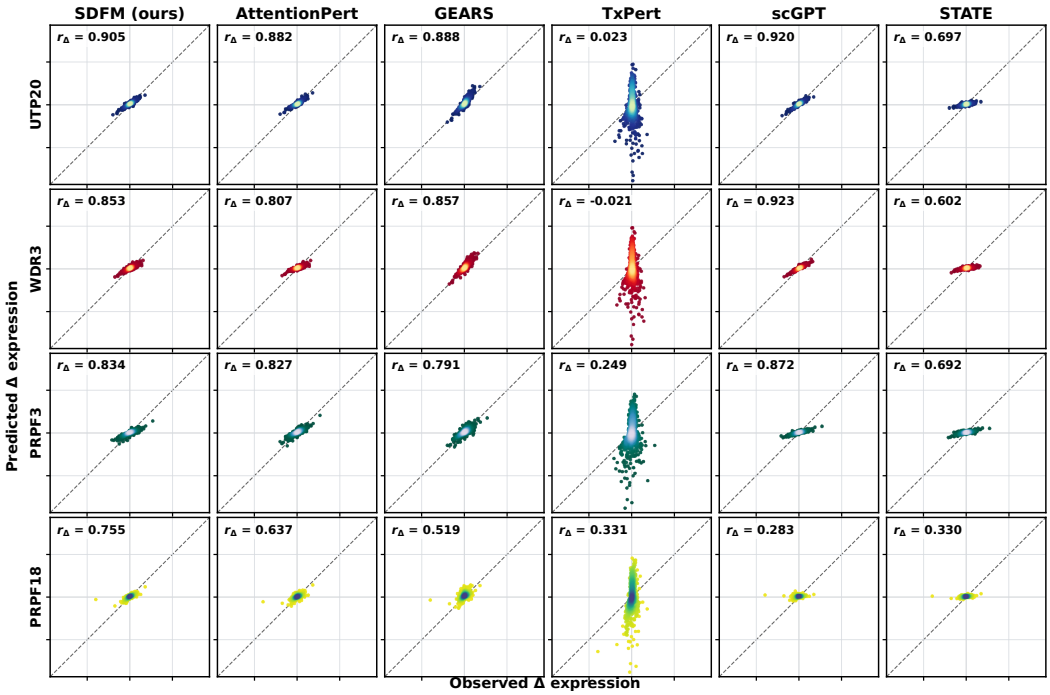

Figure 10: Mean expression changes on HepG2 for UTP20, WDR3, PRPF3, and PRPF18. Points are genes; columns compare SDFM with five baselines, and annotations report Pearson Δ.

No interaction-aware correction is applied when constructing these virtual pairs. Consequently, the resulting Adamson and HepG2 maps should be interpreted as additive in silico extensions for hypothesis generation rather than as experimentally validated predictions of non-additive genetic interactions.

The upper-right panel evaluates held-out response geometry through Spearman correlation between predicted and observed pairwise-distance matrices for experimentally assayed perturbations. The displayed subsets contain 97 Norman, 22 Adamson, and 21 HepG2 perturbations, with correlations of 0.42, 0.61, and 0.46, respectively. This comparison evaluates whether the model preserves the relative organization of measured perturbation responses; for Adamson and HepG2, it does not constitute validation of the unassayed virtual gene pairs introduced above.

The lower-right panel reports the cumulative fraction of unassayed pairs covered at each nearest-assayed rank in a 50-dimensional PCA response space. Median ranks were 0.38% for Norman, 0.70% for Adamson, and 3.43% for HepG2. Norman and Adamson therefore show closer correspondence to assayed response neighborhoods under this rank-based measure, whereas HepG2 contains a larger fraction of virtual combinations farther from sampled response neighborhoods. For Adamson and HepG2, this analysis is descriptive and is intended to identify regions of the response space that may be informative for subsequent combinatorial screening.

The interaction-aware analysis on Norman is presented in the main text (Figure 4), where experimentally measured paired perturbations permit direct evaluation of combinatorial responses. In contrast, Figures 12 and 13 visualize the additive virtual-pair constructions for Adamson and HepG2. For consistency with the Norman visualization, we display the same five score channels—synergy, suppression, neomorphism, relative epistasis, and redundancy—but emphasize that, on these two single-perturbation datasets, they are descriptive projections derived from additive virtual responses rather than experimentally validated interaction classes. These maps are intended only to organize candidate gene pairs for future combinatorial experiments.

These supplementary maps therefore serve a different purpose from the quantitative combinatorial experiments on Norman. The Norman results test whether SDFM can predict experimentally observed paired perturbations, whereas the Adamson and HepG2 maps use an explicit additive con-

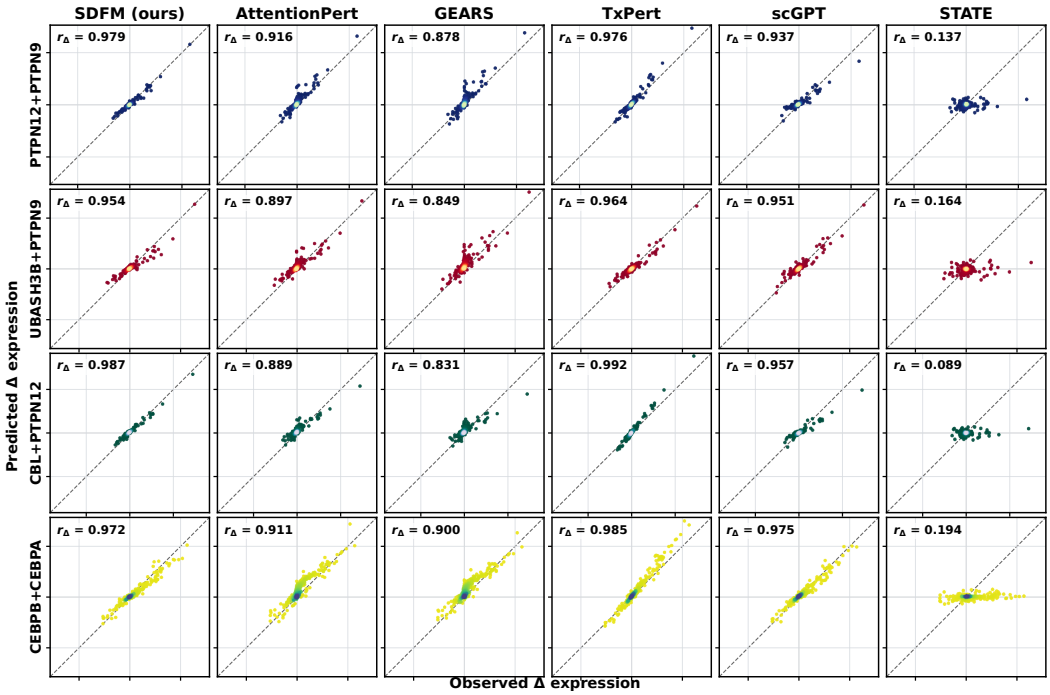

Figure 11: Mean expression changes on Norman for PTPN12+PTPN9, UBASH3B+PTPN9, CBL+PTPN12, and CEBPB+CEBPA. Points are genes; columns compare SDFM with five baselines, and annotations report Pearson Δ.

struction to explore unassayed regions of the pairwise response space. The latter should be viewed as hypothesis-generation tools for prioritizing candidate combinations rather than evidence of learned non-additive interactions.

E.8 Additional Comparisons with Transport and Flow-based Generative Models

To further evaluate SDFM against representative population transport and flow-based generative approaches, we perform additional comparisons with OT-based methods and conditional flow models.

E.8.1 Comparison with OT-based Population Transport Methods

We compare SDFM with CellOT and CINEMA-OT, two representative optimal-transport-based approaches for modeling perturbation-induced population transitions. Following their original formulation, we evaluate these methods on single-gene perturbation prediction using the Adamson, HepG2, and Norman benchmarks. Results are reported using the same evaluation metrics as the main experiments.

E.8.2 Comparison with Flow-based Generative Models

We further compare SDFM with flow-based generative perturbation models on Norman dual-gene perturbation prediction. We follow the overall evaluation protocol used in the main experiments and report performance across all dual-gene perturbation conditions.

F Parameter Sensitivity

To evaluate the robustness of the proposed framework, we performed systematic sensitivity analyses on numerical integration schemes, noise perturbations, optimal transport batch sizes, and key optimization hyperparameters. All experiments were conducted using the same training protocol as the

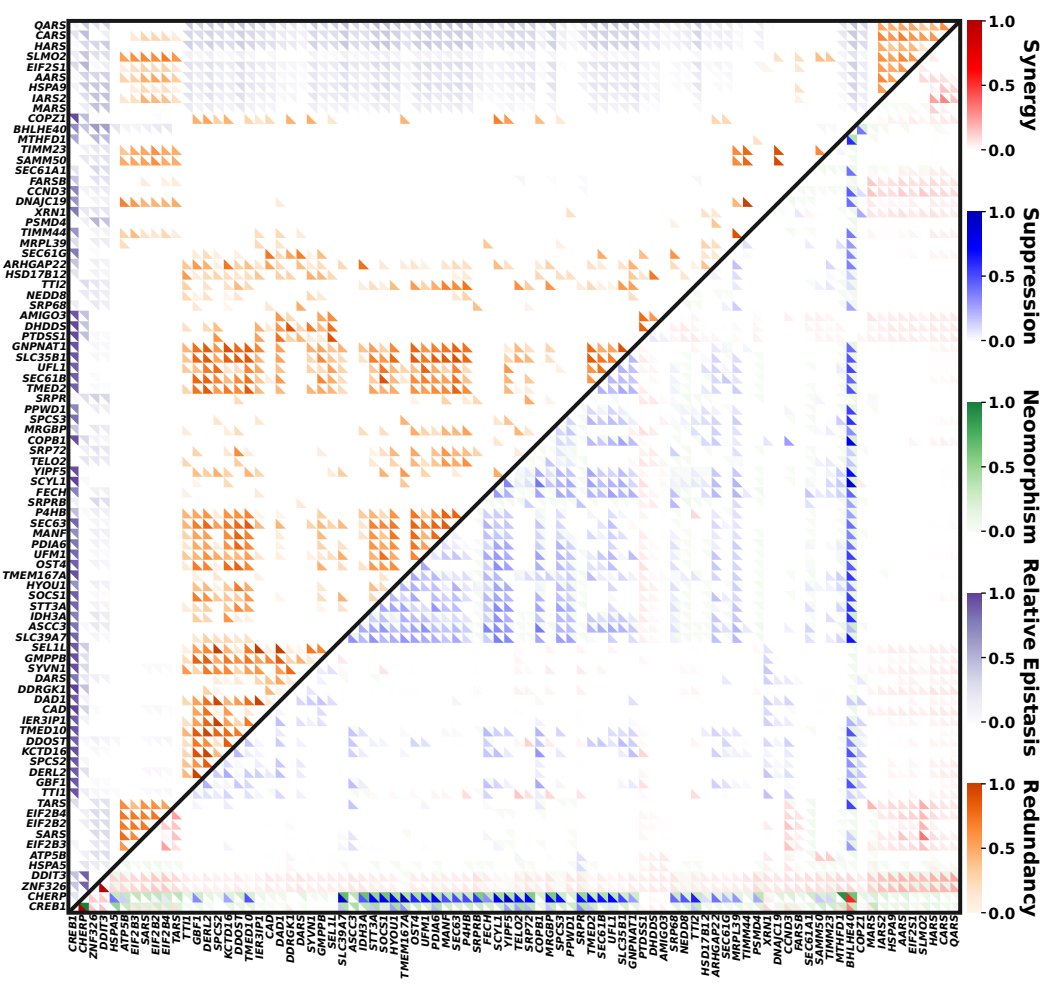

Figure 12: Additive virtual-pair score maps on Adamson (Synergy-like score). Adamson contains single-gene perturbation measurements only; each unassayed gene-pair response is constructed as the sum of the independently predicted single-gene expression shifts. The displayed score channels are exploratory and should not be interpreted as experimentally validated genetic-interaction classes.

main experiments. Unless otherwise specified, the reported metrics include Pearson Δ, DE100, and normalized retrieval performance.

F.1 Solvers and Integration Steps

We first investigated the influence of the numerical ODE solver and integration resolution on model performance. As shown in Fig. 14, different ODE solvers, including Euler, Heun, RK4, and DOPRI5, exhibited highly similar performance across all evaluated datasets (Fig. 14a). The RK4 and DOPRI5 solvers achieved marginally higher Pearson Δ and DE100 values, while the simpler Euler solver remained competitive, indicating that the learned continuous transformation is relatively robust to the choice of numerical integration scheme.

We further examined the effect of Euler integration steps (Fig. 14b). Increasing the number of integration steps generally led to rapid performance improvement from one to several steps, followed by a saturation trend. Beyond approximately 8–16 steps, additional integration steps provided limited gains. These results suggest that the learned flow field can be accurately approximated with a moderate number of solver evaluations, providing a favorable balance between computational cost and prediction accuracy.

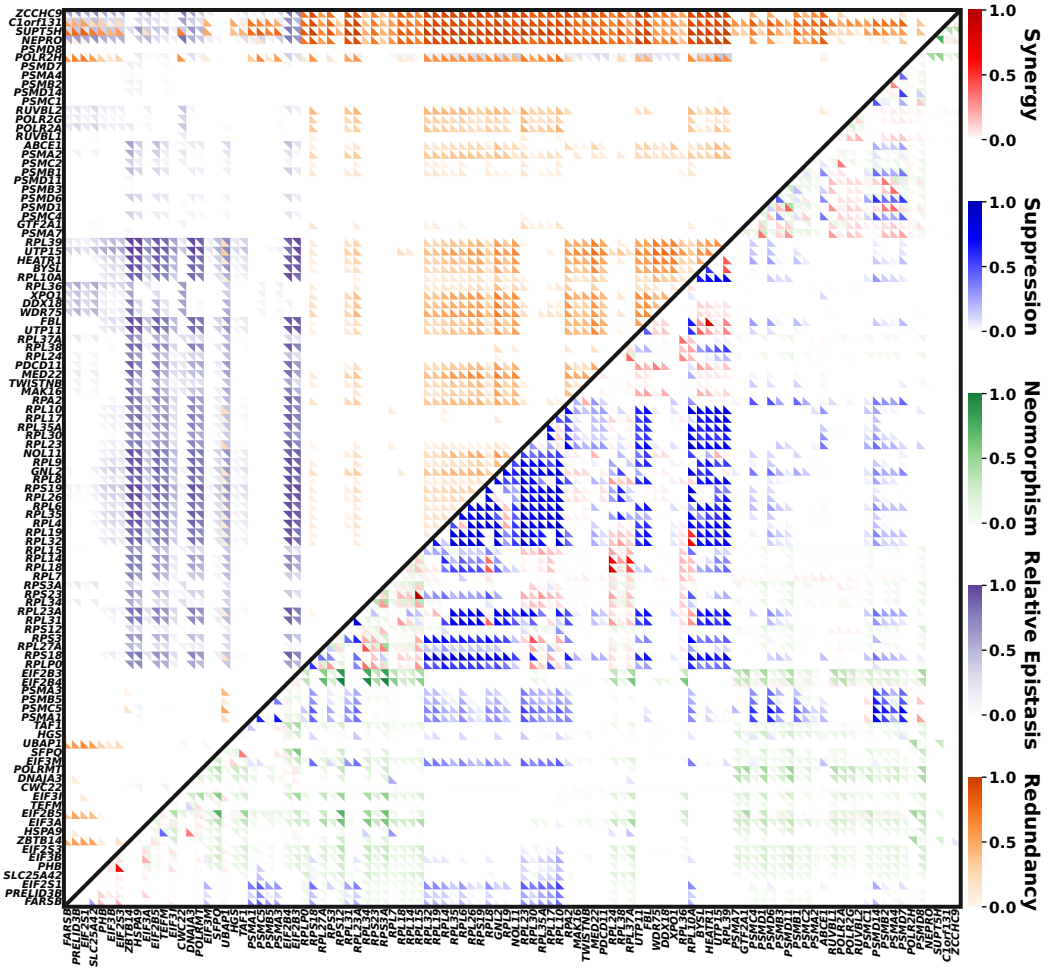

Figure 13: Additive virtual-pair score maps on HepG2 (Synergy-like score). HepG2 contains single-gene perturbation measurements only; each unassayed gene-pair response is constructed as the sum of the independently predicted single-gene expression shifts. The displayed score channels are exploratory and are not experimentally validated genetic-interaction classes.

F.2 Source Noise and OT Batch Size

We next evaluated the robustness of the model with respect to source noise perturbations and optimal transport (OT) batch size variations (Fig. 15). For source noise intensity, the model maintained stable performance under small and moderate perturbations, whereas excessive noise ($\sigma_s = 1.0$) resulted in decreased Pearson Δ and DE100 values, particularly on HepG2 data (Fig. 15a). This indicates that overly strong perturbations may introduce deviations in the source distribution and reduce the quality of the learned transport trajectory.

The OT batch size sensitivity analysis showed that increasing the batch size from 128 to 512 generally improved performance, while further increasing to 2048 resulted in only marginal changes (Fig. 15b). This suggests that a moderate OT batch size is sufficient to estimate reliable transport relationships, while larger batches mainly provide diminishing returns.

F.3 Capacity and Optimization

Finally, we examined the influence of major architectural and optimization hyperparameters (Fig. 16). The hidden dimension and network depth showed a typical capacity-performance trade-off. Increasing the hidden dimension improved performance until 512 dimensions, after which the

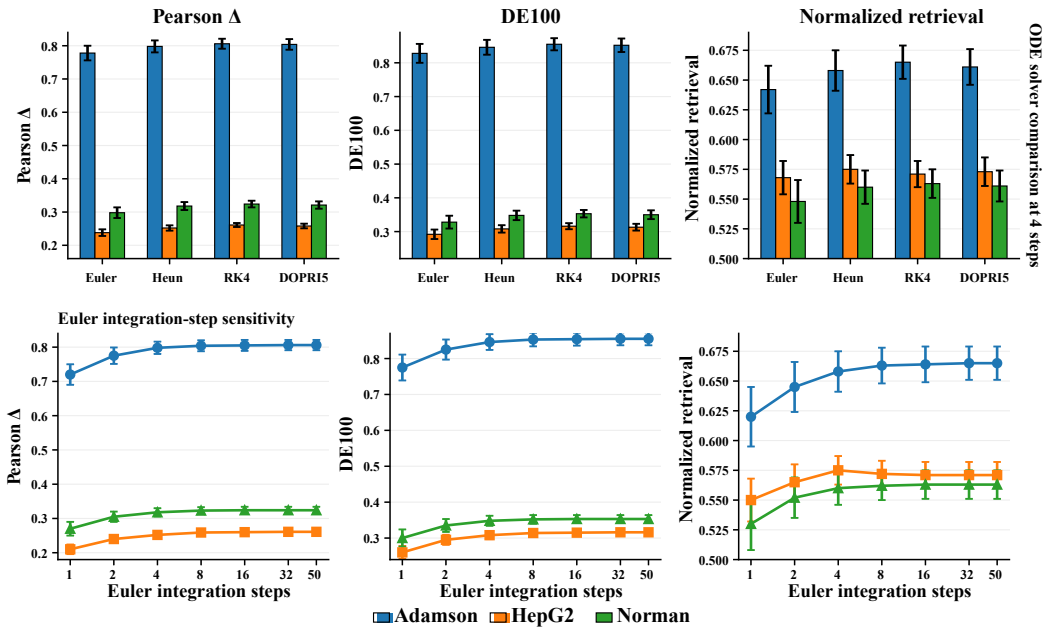

Figure 14: Sensitivity analysis of ODE solver selection and integration steps. (a) Comparison of Euler, Heun, RK4, and DOPRI5 solvers. (b) Performance variation with different Euler integration steps.

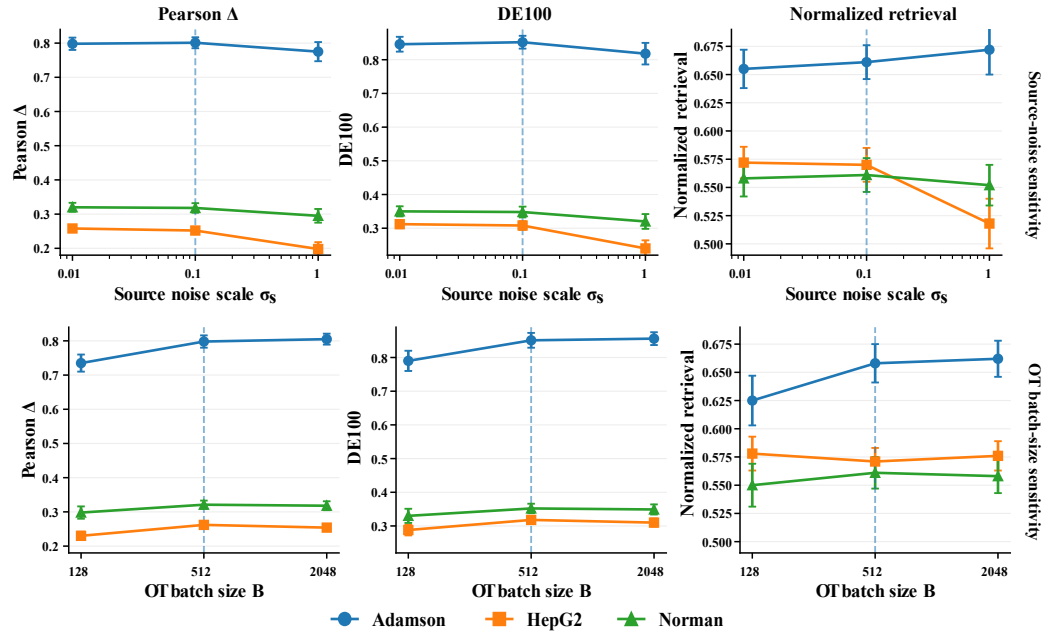

Figure 15: Sensitivity analysis of source noise scale and OT batch size. (a) Effect of source noise magnitude σ_s. (b) Effect of OT batch size B.

Table 16: Comparison with OT-based population transport methods on single-gene perturbation. Values are mean ± SD.

	All genes		DE genes		Retrieval	
Method	Pearson Δ	Spearman Δ	DE20	DE100	Normalized	Fast
Adamson single ($n = 22$)						
CellOT	0.6804 ± *0.0123*	0.4407 ± *0.0098*	0.7602 ± *0.0157*	0.7301 ± *0.0142*	0.5703 ± *0.0111*	0.5678 ± *0.0105*
CINEMA-OT	0.7012 ± *0.0117*	0.4609 ± *0.0093*	0.7705 ± *0.0149*	0.7403 ± *0.0136*	0.5681 ± *0.0108*	0.5646 ± *0.0102*
SDFM (ours)	**0.8071** ± *0.0084*	**0.5517** ± *0.0066*	**0.8889** ± *0.0102*	**0.8540** ± *0.0096*	**0.6649** ± *0.0081*	**0.6642** ± *0.0059*
HepG2 single ($n = 209$)						
CellOT	0.2104 ± *0.0047*	0.1803 ± *0.0039*	0.2501 ± *0.0058*	0.2433 ± *0.0053*	0.5002 ± *0.0069*	0.5004 ± *0.0071*
CINEMA-OT	0.2206 ± *0.0045*	0.1901 ± *0.0037*	0.2703 ± *0.0056*	0.2608 ± *0.0051*	0.5105 ± *0.0067*	0.5103 ± *0.0069*
SDFM (ours)	**0.2634** ± *0.0032*	**0.2363** ± *0.0037*	**0.3170** ± *0.0055*	**0.3153** ± *0.0054*	**0.5751** ± *0.0067*	**0.5855** ± *0.0075*
Norman single ($n = 36$)						
CellOT	0.2802 ± *0.0059*	0.1207 ± *0.0041*	0.3203 ± *0.0073*	0.3005 ± *0.0068*	0.5204 ± *0.0082*	0.5102 ± *0.0077*
CINEMA-OT	0.2906 ± *0.0057*	0.1304 ± *0.0040*	0.3301 ± *0.0071*	0.3107 ± *0.0066*	0.5308 ± *0.0080*	0.5205 ± *0.0075*
SDFM (ours)	**0.3266** ± *0.0043*	**0.1587** ± *0.0048*	**0.3681** ± *0.0127*	**0.3536** ± *0.0078*	**0.5649** ± *0.0061*	**0.5531** ± *0.0056*

Table 17: Comparison with flow-based generative models on Norman dual-gene perturbation (overall).

Method	Pearson Δ	Spearman Δ	DE20	DE100	Normalized Retrieval	Fast Retrieval
CellFlow	0.5603	0.2205	0.6602	0.6304	0.7301	0.7105
scDFM	0.4988	0.1890	0.6022	0.5691	0.6679	0.6525
SDFM (ours)	0.6820	0.2594	0.7214	0.7224	0.8104	0.7979

gain saturated. Similarly, increasing the number of network layers improved performance from 2 to 4 layers, while deeper networks did not provide further benefits. These results indicate that an intermediate model capacity provides a suitable balance between representation ability and optimization stability.

We further evaluated the weighting coefficients of different training objectives, including λ_{T}, $\lambda_{\mathrm{R}\Delta}$, and λ_{RE}. The expanded search range revealed clear optimal regions rather than monotonic trends. In particular, λ_{T} achieved the highest performance near 2.0, $\lambda_{\mathrm{R}\Delta}$ around 1.0, and λ_{RE} around 0.5. Excessively large weighting coefficients reduced performance, suggesting that overly emphasizing individual objectives may disturb the balance between flow matching and auxiliary supervision.

The learning-rate analysis further demonstrated dataset-dependent but consistent trends (Fig. 16). Performance increased when the learning rate moved from very small values to an intermediate range, followed by degradation at overly large values. This confirms the importance of proper optimization scale selection for stable training.

G Additional Controls

We examine OT pairing, knowledge masking, generalization to held-out gene families, intermediate trajectories, and distributional comparisons with additional generative baselines.

G.1 OT Pairing

Table 18 compares OT coupling with latent-space OT, PCA-space OT, entropy-regularized Sinkhorn coupling, nearest-neighbor matching, and random pairing. OT coupling gives the highest mean Pearson Δ, DE100, and normalized retrieval across the three datasets. On Adamson, Pearson Δ reaches 0.8071, compared with 0.7812 for latent-space OT ($p = 1.37 \times 10^{-3}$), and DE100 reaches 0.8540, compared with 0.8321 for PCA-space OT ($p = 8.34 \times 10^{-3}$). The comparison links the choice of training correspondences to both broad response prediction and recovery of differentially expressed genes.

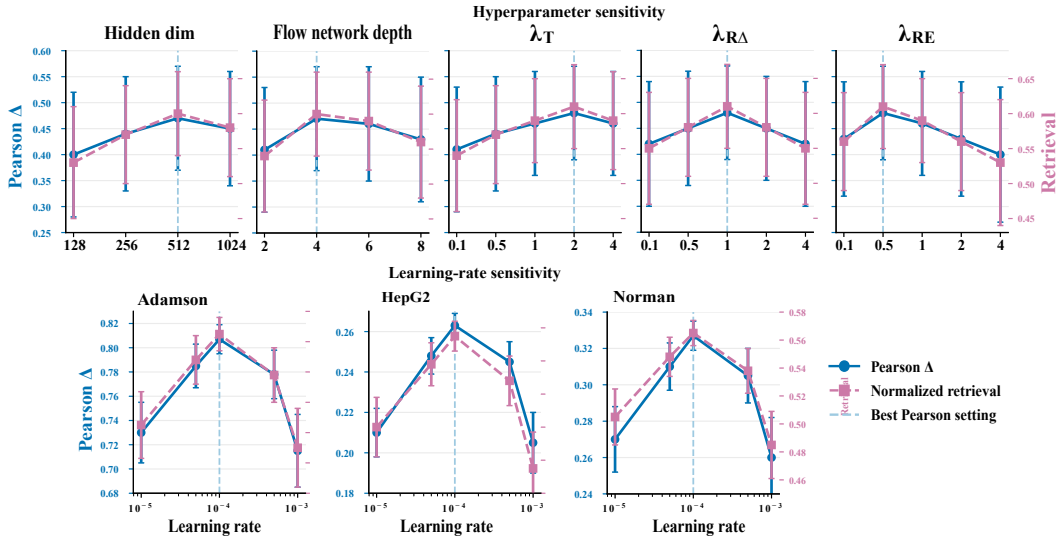

Figure 16: Sensitivity analysis of architectural and optimization hyperparameters. Upper panels show the effect of hidden dimension, network depth, and loss weighting coefficients. Lower panels show learning-rate sensitivity on Adamson, HepG2, and Norman datasets.

Table 18: OT pairing robustness. Values are mean ± SD over five replicates. Best per column is bold; second-best is underlined.

Dataset	Pairing strategy	Pearson Δ	DE100	Retrieval	SWD ↓
Adamson	**SDFM (OT coupling)**	**0.8071** ± *0.0084*	**0.8540** ± *0.0096*	**0.6649** ± *0.0081*	**1.1995** ± *0.0519*
	Latent-space OT	<u>0.7812</u> ± *0.0087*	0.8193 ± *0.0107*	<u>0.6574</u> ± *0.0091*	<u>1.2761</u> ± *0.0558*
	PCA-space OT	0.7698 ± *0.0094*	<u>0.8321</u> ± *0.0103*	0.6412 ± *0.0103*	1.3427 ± *0.0617*
	Sinkhorn OT	0.7423 ± *0.0122*	0.7987 ± *0.0143*	0.6379 ± *0.0120*	1.4521 ± *0.0703*
	Nearest neighbor	0.7534 ± *0.0107*	0.8046 ± *0.0121*	0.6518 ± *0.0097*	1.5210 ± *0.0737*
	Random pairing	0.7012 ± *0.0157*	0.7654 ± *0.0179*	0.6128 ± *0.0142*	1.7823 ± *0.0931*
HepG2	**SDFM (OT coupling)**	**0.2634** ± *0.0032*	**0.3153** ± *0.0054*	**0.5751** ± *0.0067*	**1.6691** ± *0.0411*
	PCA-space OT	<u>0.2568</u> ± *0.0037*	0.3012 ± *0.0068*	0.5712 ± *0.0069*	<u>1.6842</u> ± *0.0433*
	Latent-space OT	0.2531 ± *0.0036*	<u>0.3087</u> ± *0.0062*	<u>0.5723</u> ± *0.0071*	1.6927 ± *0.0457*
	Sinkhorn OT	0.2210 ± *0.0061*	0.2945 ± *0.0071*	0.5319 ± *0.0048*	1.7210 ± *0.0581*
	Nearest neighbor	0.2418 ± *0.0047*	0.2876 ± *0.0075*	0.5634 ± *0.0083*	1.7315 ± *0.0579*
	Random pairing	0.2214 ± *0.0068*	0.2654 ± *0.0097*	0.5312 ± *0.0101*	1.8942 ± *0.0723*
Norman	**SDFM (OT coupling)**	**0.3266** ± *0.0043*	**0.3536** ± *0.0078*	**0.5649** ± *0.0061*	**1.8767** ± *0.0481*
	PCA-space OT	<u>0.3214</u> ± *0.0049*	0.3418 ± *0.0083*	0.5487 ± *0.0067*	<u>1.8893</u> ± *0.0511*
	Latent-space OT	0.3187 ± *0.0053*	<u>0.3482</u> ± *0.0081*	0.5532 ± *0.0063*	1.9014 ± *0.0547*
	Nearest neighbor	0.3102 ± *0.0073*	0.3298 ± *0.0091*	<u>0.5583</u> ± *0.0077*	1.9328 ± *0.0613*
	Sinkhorn OT	0.3035 ± *0.0127*	0.3376 ± *0.0085*	0.5494 ± *0.0116*	1.9742 ± *0.0689*
	Random pairing	0.2781 ± *0.0089*	0.3012 ± *0.0103*	0.5187 ± *0.0103*	2.1056 ± *0.0807*

G.2 Knowledge Masking

Table 19 compares the full knowledge graphs with masking STRING annotations alone and masking both STRING and GO annotations for test perturbations. Full KG gives higher Pearson Δ, DE100, and normalized retrieval than either masking variant on all three datasets; all 18 comparisons have $p < 0.01$. Masking both sources produces larger decreases than masking STRING alone. On HepG2, Pearson Δ decreases from 0.2634 to 0.2187 and DE100 from 0.3153 to 0.2641, corresponding to relative reductions of 17.0% and 16.2%. These results quantify the contribution of the available biological annotations to prediction.

Table 19: Knowledge masking. Full KG uses all annotations; the masked variants remove STRING alone or both STRING and GO annotations for test perturbations. Values are mean ± SD over five replicates; best means are bold.

Dataset	Setting	Pearson Δ	DE100	Retrieval
Adamson	**Full KG (SDFM)**	**0.8071** ± *0.0084*	**0.8540** ± *0.0096*	**0.6649** ± *0.0081*
	Masked STRING only	0.7726 ± *0.0107*	0.8214 ± *0.0123*	0.6384 ± *0.0113*
	Masked KG (both)	0.7215 ± *0.0141*	0.7684 ± *0.0181*	0.5982 ± *0.0167*
HepG2	**Full KG (SDFM)**	**0.2634** ± *0.0032*	**0.3153** ± *0.0054*	**0.5751** ± *0.0067*
	Masked STRING only	0.2438 ± *0.0043*	0.2987 ± *0.0073*	0.5382 ± *0.0093*
	Masked KG (both)	0.2187 ± *0.0057*	0.2641 ± *0.0091*	0.4983 ± *0.0121*
Norman	**Full KG (SDFM)**	**0.3266** ± *0.0043*	**0.3536** ± *0.0078*	**0.5649** ± *0.0061*
	Masked STRING only	0.3012 ± *0.0059*	0.3287 ± *0.0091*	0.5357 ± *0.0103*
	Masked KG (both)	0.2843 ± *0.0071*	0.3012 ± *0.0113*	0.5127 ± *0.0137*

Table 20: Gene-family holdouts. Values are mean ± SD over five replicates. Bold and underlined means mark the best and second-best methods.

Method	Pearson Δ	DE100	Retrieval
SDFM (ours)	**0.5987** ± *0.0091*	**0.6324** ± *0.0117*	**0.7215** ± *0.0087*
TxPert	<u>0.5412</u> ± *0.0127*	0.6187 ± *0.0131*	0.6812 ± *0.0129*
AttentionPert	0.5237 ± *0.0143*	<u>0.6223</u> ± *0.0117*	<u>0.6943</u> ± *0.0117*
scDFM	0.4876 ± *0.0167*	0.5721 ± *0.0169*	0.6538 ± *0.0133*
CellFlow	0.4793 ± *0.0173*	0.5824 ± *0.0153*	0.6382 ± *0.0147*
GEARS	0.4218 ± *0.0197*	0.5024 ± *0.0203*	0.5216 ± *0.0217*

G.3 Gene-Family Holdouts

Holding out entire gene families tests transfer beyond genes represented in training (Table 20). SDFM has the highest mean on all three metrics. Pearson Δ reaches 0.5987, compared with 0.5412 for TxPert ($p = 3.56 \times 10^{-5}$), and normalized retrieval reaches 0.7215, compared with 0.6943 for AttentionPert ($p = 3.12 \times 10^{-3}$). DE100 reaches 0.6324, with AttentionPert next at 0.6223. The response-correlation and retrieval gains support generalization across the held-out families.

G.4 Intermediate Trajectories

Table 21 evaluates generated populations at four integration times on Norman. SDFM's SWD decreases from 2.8741 at $t = 0.25$ to 1.8767 at $t = 1.00$, with the lowest mean at each sampled time. scDFM has the next-lowest SWD at $t = 0.25$, 0.50, and 1.00, while CellFlow is next at $t = 0.75$. SDFM also has the highest reported monotonicity score, 0.943, followed by scDFM at 0.911.

G.5 Extended Distributional Evaluation

Table 22 extends the population comparison with MMD and replicate variability. The SWD, energy, and covariance means agree with Tables 10 and 11. SDFM has the lowest mean on all four metrics across the three datasets. For SWD, it reaches 1.1995 on Adamson, compared with 1.4236 for CellFlow; 1.6691 on HepG2, compared with 1.6987 for TxPert; and 1.8767 on Norman, compared with 1.9538 for scDFM. The complementary metrics describe agreement in population location, spread, dependence, and kernel-based distributional similarity.

H Implementation Details and Computational Analysis

This section summarizes the implementation details, optimization configurations, and computational characteristics of SDFM and compared baselines. Architectural choices, training settings, transport scaling, and computational comparisons are organized separately for clarity.

Table 21: Intermediate trajectories on Norman. Values are mean ± SD over five replicates. Lower SWD and higher monotonicity are better; best and second-best means are bold and underlined.

Method	$t = 0.25$	$t = 0.50$	$t = 0.75$	$t = 1.00$	Monotonicity ↑
SDFM (ours)	**2.8741** ± *0.0983*	**2.4128** ± *0.0817*	**2.0437** ± *0.0641*	**1.8767** ± *0.0481*	**0.943** ± *0.021*
scDFM	<u>2.9834</u> ± *0.1117*	<u>2.5162</u> ± *0.0937*	2.1284 ± *0.0779*	<u>1.9538</u> ± *0.0557*	<u>0.911</u> ± *0.029*
CellFlow	3.1207 ± *0.1269*	2.6741 ± *0.1027*	<u>2.1276</u> ± *0.0889*	1.9817 ± *0.0613*	0.851 ± *0.041*
MeanShift-Delta	3.4182 ± *0.1557*	3.1274 ± *0.1339*	2.7891 ± *0.1123*	2.2522 ± *0.0979*	0.681 ± *0.059*

Table 22: Extended population distribution errors. Values are mean ± SD over five replicates; lower is better. Bold and underlined means mark the best and second-best methods.

Dataset	Method	SWD ↓	Energy ↓	Cov. error ↓	MMD ↓
Adamson	**SDFM (ours)**	**1.1995** ± *0.0519*	**2.4960** ± *0.1207*	**0.7588** ± *0.0179*	**0.3215** ± *0.0147*
	CellFlow	<u>1.4236</u> ± *0.0679*	<u>3.0427</u> ± *0.1583*	<u>0.7802</u> ± *0.0227*	0.3487 ± *0.0139*
	scDFM	1.4572 ± *0.0713*	3.2289 ± *0.1537*	0.7864 ± *0.0219*	0.3682 ± *0.0181*
	TxPert	1.5813 ± *0.0787*	3.8715 ± *0.1813*	0.8017 ± *0.0193*	0.4532 ± *0.0243*
	AttentionPert	1.6198 ± *0.0821*	4.1153 ± *0.1921*	0.8092 ± *0.0271*	<u>0.3384</u> ± *0.0127*
	MeanShift-Delta	2.2063 ± *0.1153*	11.6001 ± *0.5817*	0.8518 ± *0.0339*	0.6824 ± *0.0381*
HepG2	**SDFM (ours)**	**1.6691** ± *0.0411*	**2.3995** ± *0.0979*	**0.7436** ± *0.0157*	**0.3012** ± *0.0129*
	TxPert	<u>1.6987</u> ± *0.0457*	<u>2.5037</u> ± *0.1171*	<u>0.7513</u> ± *0.0197*	0.3562 ± *0.0159*
	AttentionPert	1.7042 ± *0.0479*	2.5324 ± *0.1043*	0.7536 ± *0.0191*	0.3718 ± *0.0171*
	scDFM	1.7125 ± *0.0513*	2.5571 ± *0.1123*	0.7548 ± *0.0173*	0.3384 ± *0.0137*
	CellFlow	1.7218 ± *0.0541*	2.6035 ± *0.1081*	0.7584 ± *0.0209*	<u>0.3287</u> ± *0.0119*
	MeanShift-Delta	1.7176 ± *0.0617*	2.6500 ± *0.1227*	0.7596 ± *0.0219*	0.3892 ± *0.0211*
Norman	**SDFM (ours)**	**1.8767** ± *0.0481*	**2.7611** ± *0.1117*	**0.3390** ± *0.0109*	**0.1621** ± *0.0091*
	scDFM	<u>1.9538</u> ± *0.0557*	<u>3.1072</u> ± *0.1307*	<u>0.3649</u> ± *0.0139*	0.2387 ± *0.0143*
	CellFlow	1.9817 ± *0.0613*	3.2048 ± *0.1079*	0.3718 ± *0.0159*	0.1924 ± *0.0081*
	AttentionPert	2.0869 ± *0.0669*	3.6123 ± *0.1417*	0.3908 ± *0.0129*	0.2234 ± *0.0121*
	TxPert	2.0235 ± *0.0741*	3.3506 ± *0.1531*	0.3776 ± *0.0179*	<u>0.1876</u> ± *0.0097*
	MeanShift-Delta	2.2522 ± *0.0979*	5.5672 ± *0.2513*	0.4259 ± *0.0241*	0.3812 ± *0.0229*

H.1 SDFM Architecture and Training Configuration

SDFM integrates knowledge-conditioned perturbation representation, Cross-Aware perturbation composition, optimal transport pairing, and conditional flow matching. Table 23 summarizes the architecture and optimization settings used in all experiments.

The Cross-Aware composer preserves the original additive representation for single-gene perturbations and introduces a symmetric interaction residual only for combinatorial perturbations. This design allows the same composition function to be used during training and inference.

H.2 Exact Optimal Transport Scaling

SDFM constructs population-level correspondences using exact mini-batch optimal transport. Table 24 reports memory scaling under increasing condition batch sizes.

The observed memory increase follows the expected quadratic scaling of pairwise transport matrices. The mini-batch formulation keeps exact OT coupling computationally feasible during training.

H.3 Baseline Configurations and Computational Comparison

Table 25 combines baseline architectural characteristics with computational measurements. All methods are evaluated under their corresponding recommended implementations.

Table 23: SDFM architecture and training configuration.

Component	Configuration	Details
State representation	Identity backbone	Normalized-expression space
Gene encoder	Exphormer	STRING+GO graph; 4 layers
Graph encoder	Width / heads	128 / 2; dropout 0.1
Graph construction	Topology	Reverse edges, self-loops, degree-3 expander, top-20 graph
Gene representation	Output dimension	64-dimensional embedding
Perturbation composition	Inter-Aware composer	Additive embedding + symmetric interaction residual
Velocity input	Conditioned state	$[x_t, x_0, x_t - x_0, t, e_p]$
State branch	MLP	$3G \to 512 \to 512$
Condition branch	Perturbation/time	$(64 + 64) \to 256 \to 256$
Cross-attention	Architecture	4 residual blocks, 4 heads, 4 condition tokens
Block MLP	Hidden layers	$512 \to 2048 \to 512$
OT coupling	Transport method	Exact mini-batch OT with squared-Euclidean Hungarian pairing
OT batch	Training configuration	8 conditions $\times$ 16 cells
Objective	Loss weights	FM / teacher-Δ / rollout-Δ / endpoint $1/2/0.1/0.05$
Optimization	Optimizer	AdamW
Learning rates	Parameter groups	Encoder: 3×10^{-4}; Velocity: 10^{-4}
Schedule	Regularization	5-epoch warmup; gradient clip 1
Training budget	Updates	35$\times$100 (Adamson); $\leq 60 \times 100$ (Norman/HepG2)

Table 24: Exact OT computational scaling.

Cells / condition	OT cells / step	Temporary memory
8	64	0.618 MiB
16	128	1.237 MiB
32	256	2.479 MiB
64	512	4.973 MiB
128	1024	10.008 MiB
256	2048	20.266 MiB

The comparison shows that SDFM maintains moderate parameter size while integrating biological knowledge conditioning and continuous transport learning. Compared with large pretrained or flow-based baselines, SDFM requires lower memory while retaining explicit perturbation conditioning.

LIMITATIONS

SDFM learns population-level transport paths between control and perturbed cells. Since perturbation datasets typically contain independently sampled populations rather than longitudinal measurements, these paths should be interpreted as distribution-level transitions rather than experimentally observed single-cell trajectories. In addition, interaction scores derived from single-gene perturbation datasets represent model-based interaction-like signals rather than experimentally validated genetic interactions. The model may also be less reliable for perturbations substantially outside the training distribution, including unseen cellular contexts or rare combinatorial perturbations.

Reproducibility: All experiments use fixed training protocols and reported hyperparameter configurations. We will release the code, model configurations, and evaluation scripts to facilitate independent reproduction and further investigation.

Data Availability: All datasets and biological annotations used in this study are publicly available. We provide dataset preprocessing details and evaluation protocols to facilitate reproducibility.

Table 25: Baseline configurations and computational comparison. Parameters, GPU hours, and peak VRAM are reported in the Adamson/Norman/HepG2 order when applicable.

Method	Architecture	Params (M)	GPU hours	VRAM (GB)
SDFM	Knowledge-conditioned flow	30.1	3.48	8.5
scGPT	12-layer Transformer	51.9	29.39	20
TxPert	MLP + Exphormer	7.3	7.94	5.5
GEARS	GO/coexpression GNN	1.9	15.83	6.5
AttentionPert	GNN + perturbation attention	4.6	50.93	16
STATE	Set Transformer	66.4	7.76	6.5
CPA	Autoencoder	4.2	24.35	3.5
CellFlow	Conditional flow model	65.0	14.10	11