scAMod: Generalizing Perturbation Prediction through Adaptive Gene Modules

Anonymous authors
Paper under double-blind review

Abstract

In silico perturbation prediction offers a scalable view of how cells respond to interventions. Sample variation, destructive sequencing, and dropout make this prediction hard to generalize to the response distributions of unseen contexts. Existing distributional perturbation models emphasize population-level matching, yet jointly capturing distributional shifts and coordinated gene responses within individual cells remains challenging. We propose **scAMod**, a distributional perturbation prediction model built on adaptive gene modules. We use prior gene interactions and balanced optimal transport reassignment to construct semantically anchored modules, whose gene composition adapts to cellular expression context while retaining shared module semantics. In our method, the Adaptive Module Tokenizer first converts cell-specific transcriptomes into module tokens, which are compressed through a module-centered bottleneck and reconstructed by a gene query decoder. We then use perturbation-conditioned flow matching to generate perturbed cell distributions. Finally, to model condition-specific responses, we introduce Perturbation Module Readout (PMR), which uses the perturbation condition to guide the readout of generated module states into gene-level predictions. On the PBMC donor leave-out benchmark, scAMod reduces DEG energy distance by 24.2% and improves DEG pseudobulk R^2 by 8.7% over the strongest evaluated baseline. Together with component ablations, these results support the effectiveness of module-centered modeling for generalizable distributional perturbation prediction. Code is available at `https://anonymous.4open.science/r/scAMod-A775`.

1 Introduction

In silico perturbation prediction aims to predict cellular expression changes following interventions (Miladinovic et al., 2025) and is a key component of modeling state transitions in virtual cells (Qian et al., 2026). Responses to the same intervention vary with cellular states and biological backgrounds, differing in both magnitude and the genes affected. Predicting perturbed cell distributions is therefore necessary to characterize variation within populations beyond average expression changes. Meanwhile, experiments cannot exhaustively cover combinations of perturbations, doses, and cellular contexts, requiring models to generalize to unmeasured settings (Dixit et al., 2016; Replogle et al., 2022). Together, these needs motivate generalizable distributional perturbation prediction to support drug discovery and precision medicine.

Perturbation responses vary across cells, motivating distributional prediction beyond changes in mean expression. For unpaired observations, optimal transport establishes probabilistic correspondences between control and perturbed populations (Dong et al., 2023). Building on this distributional perspective, neural optimal transport learns response mappings (Bunne et al., 2023; 2022), while conditional generative approaches such as flow matching learn to generate perturbed cell populations (Klein et al., 2025; Yu et al., 2026). However, capturing population-level shifts together with coordinated gene responses within individual cells remains challenging.

Biological priors provide shared information for modeling cellular responses. Existing approaches incorporate this information through pretrained gene representations derived from protein language models (Adduri et al., 2026) or through cross-attention to perturbation-associated biological knowledge (Wang et al., 2026).These approaches enrich model representations with biological knowledge.

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.

However, prior relationships describe shared biological structure, while the coordinated expression of related genes varies with cellular context (Tesson et al., 2010; Su et al., 2023). Effective prior integration therefore requires accounting for how shared relationships manifest in different cellular states, allowing coordinated responses within individual cells to inform the prediction of heterogeneous population responses.

We introduce scAMod (single-cell Adaptive gene Modules), a framework for distributional perturbation prediction centered on context-adaptive gene module composition. Prior knowledge of gene interactions establishes a shared semantic reference for module construction. To retain cell-level heterogeneity, the Adaptive Module Tokenizer uses cellular expression context to adjust prior gene–module associations, yielding semantic module tokens with cell-specific compositions. A module-centered autoencoder compresses these tokens and reconstructs gene-level expression through a gene query decoder. To learn perturbation response distributions, we establish pseudo-pairs between control and perturbed cells using optimal transport and train conditional flow matching to generate perturbed cell populations under encoded perturbation conditions. Finally, Perturbation Module Readout (**PMR**) uses the perturbation condition to guide the contribution of generated modules to gene-level responses. Together, this framework supports generalizable prediction of perturbed populations while accounting for variation in individual cellular responses.

We evaluate scAMod on SciPlex3 for generalization to unseen drug perturbations and on PBMC for cytokine response prediction across heterogeneous donor backgrounds. On the PBMC donor leave-out benchmark, scAMod reduces DEG energy distance by 24.2% and improves DEG pseudobulk R^2 by 8.7% over the strongest evaluated baseline. Ablation studies and module analyses further assess the roles of its components and their adaptation to cellular context. Our contributions are as follows:

- We introduce **adaptive gene modules** as a structural modeling bias for distributional perturbation prediction, enabling prior integration that adapts to cellular context.
- We develop **scAMod**, a perturbation prediction model organized around adaptive gene modules for conditional generation of perturbed cell distributions.
- We introduce **PMR**, which uses perturbation conditions to guide the readout of generated module states into gene-level responses.
- Through extensive experiments, we demonstrate the generalization capability of scAMod in distributional perturbation prediction across perturbations and cellular contexts.

2 Related Work

Distributional Perturbation Modeling. Early perturbation models learn cellular responses through latent-space transformations, including latent arithmetic in scGen (Lotfollahi et al., 2019) and compositional representations in chemCPA (Hetzel et al., 2022). biolord separates known attributes from unobserved cellular factors to support counterfactual state generation (Piran et al., 2024). For unpaired control and perturbed populations, CINEMA-OT constructs transport-based counterfactual matches (Dong et al., 2023), while CellOT and condOT learn neural optimal transport maps (Bunne et al., 2023; 2022). More recent generative approaches model population responses through conditional flow matching in CellFlow and scDFM (Klein et al., 2025; Yu et al., 2026), neural Schrödinger bridges in Departures (Chi et al., 2026), or set-level attention in STATE (Adduri et al., 2026). Existing methods increasingly capture perturbation-induced distributional shifts, yet jointly modeling these shifts and coordinated gene responses within individual cells remains challenging.

Knowledge-Informed Perturbation Modeling. Biological knowledge and pretrained representations guide perturbation modeling through network constraints, gene representations, and structured expression spaces. GEARS incorporates gene relationship graphs into genetic perturbation prediction (Roohani et al., 2024), while CellOracle simulates perturbation effects using inferred gene regulatory networks (Kamimoto et al., 2023). AdaPert selects perturbation-relevant subgraphs from biological knowledge, and ALIGNED integrates neural prediction with regulatory knowledge refinement (Piao et al., 2026; Xiang & Ai, 2026). Single-cell foundation models offer another route for incorporating pretrained knowledge. scGPT and GeneCompass learn gene representations that

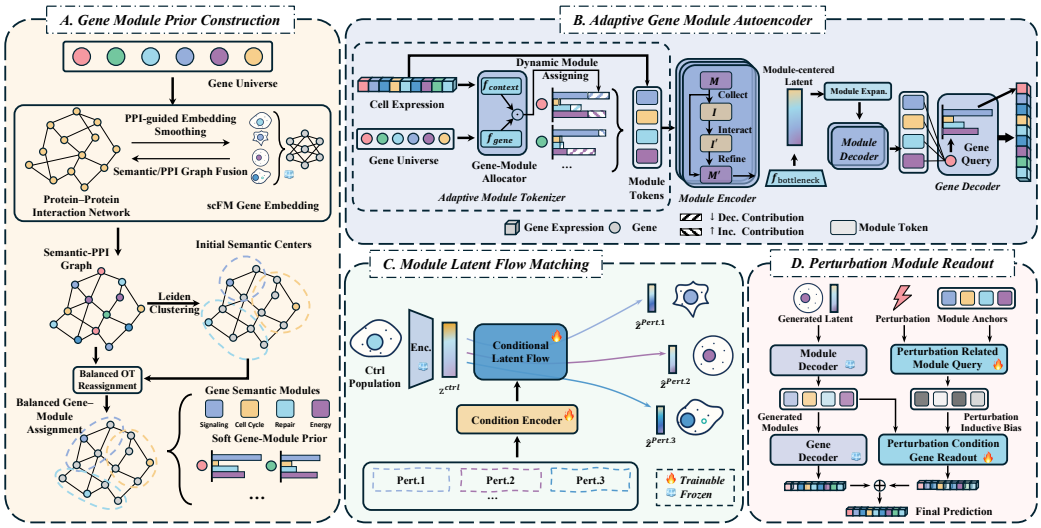


Figure 1: **Overview of scAMod.** (**A**) Prior gene interactions define module anchors and a soft gene–module prior. (**B**) The Adaptive Gene Module Autoencoder tokenizes expression, forms a module-centered bottleneck, and reconstructs gene expression. (**C**) Conditional latent flow generates perturbed cell states. (**D**) Perturbation Module Readout extracts perturbation-relevant gene information from generated modules.

support downstream cellular modeling (Cui et al., 2024; Yang et al., 2024). At the representation level, expiMap, VEGA, and Spectra organize expression through gene programs or biologically informed factors (Lotfollahi et al., 2023; Seninge et al., 2021; Kunes et al., 2024). Related approaches also exploit higher-level gene organization. SAVE uses gene blocks and gene-block attention for multi-condition single-cell generation (Li et al., 2026). These studies demonstrate different ways to integrate biological and pretrained knowledge into data-driven models of cellular responses.

However, integrating gene-level prior knowledge into population-level response generation while preserving cell-specific gene coordination presents a modeling challenge. scAMod incorporates prior gene interactions through context-adaptive gene module composition, retaining shared module semantics while allowing gene participation to vary with cellular expression. Perturbation Module Readout (PMR) further uses the perturbation condition to derive gene-level responses from generated module states.

3 Method

Overview. Given a control population $\mathcal{X}$ and an encoded perturbation condition p (Appendix A.5), the goal is to generate a population $\mathcal{Y}'$ that approximates the observed perturbed population $\mathcal{Y}$. Figure 1 summarizes scAMod. Prior gene interactions define shared module anchors and a soft gene–module prior, which the Adaptive Module Tokenizer combines with each cell's expression to form context-adaptive module tokens. The autoencoder maps these tokens into a module-centered latent space, where conditional flow matching generates perturbed states under p. The module decoder then recovers generated module states, and Perturbation Module Readout (PMR) reads them under p to produce gene-level expression profiles.

3.1 Gene Module Prior Construction

Before modeling cell-specific molecular responses, we establish a biologically grounded module coordinate system that provides transferable semantic anchors across perturbations and cells. Let G be the number of genes in the dataset, K the number of modules, and D the embedding dimension. The gene embedding matrix $\mathbf{E} \in \mathbb{R}^{G\times D}$ is initialized from a pretrained single-cell foundation model (scFM); genes without available embeddings are randomly initialized. Let $\mathbf{e}_g \in \mathbb{R}^D$ denote the embedding of gene g in $\mathbf{E}$. To incorporate prior gene interactions, we propagate these representations

over the protein–protein interaction (PPI) network, starting from $\mathbf{H}^{(0)} = \mathrm{norm}(\mathbf{E})$:

$$\mathbf{H}^{(\ell+1)} = \mathrm{norm}\Big(\alpha\mathbf{H}^{(0)} + (1-\alpha)\mathbf{A}_{\mathrm{ppi}}\mathbf{H}^{(\ell)}\Big). \tag{1}$$

Here, $\mathbf{A}_{\mathrm{ppi}}$ is the PPI adjacency matrix, and α controls the retention of the initial gene representations. We build a semantic kNN graph from $\mathbf{H}^{(L)}$ and combine its edge weights with PPI edge weights scaled by β. Leiden clustering (Traag et al., 2019) on the resulting graph provides initial gene groups, whose centroids initialize the modules.

We refine gene assignments through balanced optimal transport, using the gene–module cost

$$C_{gk} = -(\mathbf{H}_g^{(L)})^\top \mathbf{c}_k - \lambda B_{gk}, \tag{2}$$

where $\mathbf{c}_k$ is the current module center. The Leiden-based term B_{gk} regularizes reassignment toward the initial gene groups, limiting drift of the module centers. Gene reassignment and center updates alternate until the assignments stabilize or a fixed iteration limit is reached.

After balancing, we recompute the module centers from the original gene embeddings within each balanced group and convert gene–center similarities into a soft prior (Algorithm 1).

$$S_{gk} = \mathrm{softmax}_k(\mathbf{e}_g^\top \mathbf{c}_k/\tau), \tag{3}$$

where τ controls the prior temperature. The normalized matrix $\mathbf{S}$ provides a shared module reference for cell-specific routing and gene-level decoding (Figure 1**A**).

3.2 Adaptive Gene Module Autoencoder

The autoencoder instantiates the shared gene–module prior for each cell and reconstructs expression through a module-centered bottleneck.

Adaptive Module Tokenizer. For a cell expression vector $\mathbf{x} \in \mathbb{R}^G$, the tokenizer adjusts each gene's prior module participation using its embedding and the full cellular expression context:

$$A_{gk}(\mathbf{x}) = \mathrm{softmax}_k\left(\log S_{gk} + \eta[f_{\mathrm{res}}(\mathbf{e}_g, \mathbf{x})]_k\right), \tag{4}$$

The adjustment function f_{res} combines a learned projection of the gene embedding with module-specific context vectors obtained through a projection of the full expression profile. Their scaled dot products give the adjustment for each module, and η controls its strength relative to the shared prior. Thus, $A_{gk}(\mathbf{x})$ describes how strongly gene g participates in module k in this cell. The calculation is specified in Appendix A.6.1.

The tokenizer aggregates expression into module tokens using a trainable gene value vector $\mathbf{v}_g \in \mathbb{R}^{d_v}$ initialized from the scFM embedding of gene g:

$$\mathbf{m}_k = f_{\mathrm{tok}}\left(\frac{\sum_g A_{gk}(\mathbf{x}) x_g \mathbf{v}_g}{\sqrt{\sum_g A_{gk}(\mathbf{x})} + \epsilon}\right). \tag{5}$$

The denominator normalizes for module mass. Induced-attention blocks (Lee et al., 2019) then let module tokens exchange information before they are compressed into the cell state. We denote this interaction and compression by $\mathbf{z} = f_{\mathrm{enc}}([\mathbf{m}_1; \ldots; \mathbf{m}_K])$. This provides the module-centered bottleneck for subsequent response generation (Figure 1**B**).

Gene Query Decoder. To recover gene-level expression, the decoder expands $\mathbf{z}$ into module tokens $[\mathbf{m}'_1; \ldots; \mathbf{m}'_K] = f_{\mathrm{dec}}(\mathbf{z})$. A projected gene embedding $\mathbf{q}_g = f_{\mathrm{q}}(\mathbf{e}_g)$ queries these tokens, with the shared prior providing a semantic bias:

$$Q_{gk}(\mathbf{x}) = \mathrm{softmax}_k\left(\mathbf{q}_g^\top f_{\mathrm{k}}(\mathbf{m}'_k)/\sqrt{d} + \lambda_{\mathrm{prior}} \log S_{gk}\right). \tag{6}$$

Here, f_{k} and f_{v} are learned key and value maps, d is the common dimension of the gene queries and module keys, and λ_{prior} controls the prior bias. The readout weights $Q_{gk}(\mathbf{x})$ combine the recovered modules to reconstruct each gene:

$$\hat{x}_g = f_{\mathrm{gene}}\big(\mathbf{q}_g, \sum_k Q_{gk}(\mathbf{x}) f_{\mathrm{v}}(\mathbf{m}'_k)\big). \tag{7}$$

Unlike the encoder weights $A_{gk}(\mathbf{x})$, the decoder weights $Q_{gk}(\mathbf{x})$ are inferred from the recovered module states and are aligned during training as described in Section 3.5. Algorithm 2 summarizes autoencoder training.

3.3 Module Latent Flow Matching

We perform conditional flow matching in the latent space learned by the adaptive module autoencoder. For each perturbation embedding p, OT couples the control and perturbed latent distributions to construct training pairs $(\mathbf{z}_0, \mathbf{z}_1)$, where $\mathbf{z}_0$ represents a control cell and $\mathbf{z}_1$ its OT-paired perturbed cell. We learn a velocity field conditioned on p using the objective in Section 3.5. The conditional flow models the transition from control to perturbed latent distributions (Figure 1**C**).

$$\mathbf{z}_0 = \mathbf{z}^{\mathrm{ctrl}}, \qquad \mathbf{z}_1 = \mathbf{z}^{\mathrm{pert}}, \qquad \frac{d\mathbf{z}_t}{dt} = \mathbf{v}_\theta(t, \mathbf{z}_t, p). \tag{8}$$

The training and generation procedures are summarized in Algorithm 3.

3.4 Perturbation Module Readout

PMR prioritizes perturbation-relevant modules in generated cell states and uses their readout to augment gene-level response predictions (Figure 1**D**). The perturbation embedding p queries the module anchors through cross-attention, yielding condition-specific weights over all modules:

$$\mathbf{g}_p = \mathrm{CrossAttn}\left(p, \{\mathbf{a}_k\}_{k=1}^{K}\right). \tag{9}$$

Here, $\mathbf{a}_k$ is the learned decoder-space anchor for module k, and $\mathbf{g}_p$ denotes the resulting attention weights. Let $\{\mathbf{m}'_k\}_{k=1}^{K}$ be the module tokens decoded from a generated cell state. For each gene, PMR combines its query $\mathbf{q}_g$ with the generated module tokens. The attention and readout scores are separately learned projected dot products, denoted by $\langle\cdot,\cdot\rangle_{\mathrm{att}}$ and $\langle\cdot,\cdot\rangle_{\mathrm{read}}$, explicitly defined in Appendix A.6.1:

$$\begin{aligned} R_{gk} &= \mathrm{softmax}_k(\langle \mathbf{q}_g, \mathbf{m}'_k\rangle_{\mathrm{att}} + \rho \log S_{gk} + \log(g_{p,k} + \epsilon)), \\ h_g &= \sum_{k=1}^{K} R_{gk} \langle \mathbf{q}_g, \mathbf{m}'_k\rangle_{\mathrm{read}}. \end{aligned} \tag{10}$$

The coefficient ρ sets the static-prior strength. The gate $\mathbf{g}_p$ is shared across cells under the same perturbation, while R_{gk} and h_g also depend on each cell's generated module states. A gene-wise projection f_{out} maps $\mathbf{h} = (h_g)_{g=1}^{G}$ to a perturbation-specific expression contribution. Combined with the AE decoder's expression $\widetilde{\mathbf{y}}$, the prediction is

$$\mathbf{y}' = \widetilde{\mathbf{y}} + f_{\mathrm{out}}(\mathbf{h}). \tag{11}$$

PMR uses the generated module states without a control–generated subtraction (Figure 1**D**). Algorithm 4 details its readout and optimization.

3.5 Learning Objectives and Staged Optimization

We optimize scAMod sequentially in three stages: we first train the module-centered autoencoder; we then freeze it and train conditional flow in its latent space; finally, we freeze both components and train PMR to learn perturbation-specific module readout.

Autoencoder loss. We first learn the module-centered autoencoder using expression reconstruction together with two complementary routing objectives:

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{qry}}\mathcal{L}_{\mathrm{qry}} + \lambda_{\mathrm{anc}}\mathcal{L}_{\mathrm{anc}}, \tag{12}$$

Reconstruction term. This term recovers the observed expression profile,

$$\mathcal{L}_{\mathrm{rec}} = \mathbb{E}_{\mathbf{x}} \frac{1}{G} \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2. \tag{13}$$

Query alignment term. This term aligns decoder readout weights with the encoder routing profile while treating that profile as a detached target:

$$\mathcal{L}_{\mathrm{qry}} = \mathbb{E}_{\mathbf{x},g}\left[\mathrm{KL}\left(\mathrm{stopgrad}[\mathbf{A}_{g,:}(\mathbf{x})] \;\|\; \mathbf{Q}_{g,:}(\mathbf{x})\right)\right]. \tag{14}$$

Anchor term. To discourage module collapse while retaining cell-specific routing, this term aligns the router averaged over a minibatch $\mathcal{B}$ with the static prior,

$$\bar{\mathbf{A}}_{g,:} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} \mathbf{A}_{g,:}(\mathbf{x}), \qquad \mathcal{L}_{\text{anc}} = \mathbb{E}_g \left[\mathrm{KL} \left(\bar{\mathbf{A}}_{g,:} \parallel \mathbf{S}_{g,:} \right) \right]. \tag{15}$$

Conditional flow matching loss. With the autoencoder fixed, we sample latent pairs $(\mathbf{z}_0, \mathbf{z}_1) \sim \pi_p$, where π_p is the entropically regularized OT coupling between the control and perturbed latent distributions for perturbation p. For $t \sim \mathcal{U}(0,1)$, we construct a noise-regularized path $\mathbf{z}_t = \phi_t(\mathbf{z}_0, \mathbf{z}_1)$ and train the conditional latent vector field using

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{(\mathbf{z}_0, \mathbf{z}_1) \sim \pi_p,\, t} \left\| \mathbf{v}_\theta(t, \mathbf{z}_t, p) - \partial_t \phi_t(\mathbf{z}_0, \mathbf{z}_1) \right\|_2^2. \tag{16}$$

PMR loss. Finally, we freeze the autoencoder and conditional flow, and train PMR from generated module states and observed perturbed cells. For a training perturbation p, let $\mathcal{G}_{\text{DEG}}$ denote the DEG set derived from its observed training cells and matched controls. The symbols $\bar{\mathbf{y}}'_{\text{DEG}}$ and $\bar{\mathbf{y}}_{\text{DEG}}$ denote the pseudobulk means of $\mathcal{Y}'$ and $\mathcal{Y}$, restricted to $\mathcal{G}_{\text{DEG}}$. The PMR objective combines DEG-level mean accuracy with distributional agreement:

$$\mathcal{L}_{\text{PMR}} = \mathcal{L}_{\text{MSE}}^{\text{DEG}} + \lambda_{\text{ED}} \mathcal{L}_{\text{ED}}. \tag{17}$$

The DEG-level pseudobulk term is defined as

$$\mathcal{L}_{\text{MSE}}^{\text{DEG}} = \frac{1}{|\mathcal{G}_{\text{DEG}}|} \left\| \bar{\mathbf{y}}'_{\text{DEG}} - \bar{\mathbf{y}}_{\text{DEG}} \right\|_2^2. \tag{18}$$

The distributional term compares predicted and observed minibatch populations over all modeled genes, without DEG masking, using the energy-distance loss specified in Appendix A.6.1.

$$\mathcal{L}_{\text{ED}} = \text{EnergyDistance}(\mathcal{Y}', \mathcal{Y}). \tag{19}$$

DEG sets provide training supervision only. At inference, generation requires control cells and the perturbation condition, not observed test responses or a DEG set. Test-condition DEGs are used for post hoc DEG-space evaluation.

4 Experiments

4.1 Experimental Setup

Datasets and generalization settings. We evaluate scAMod in two complementary generalization settings. **(1) Cross-donor cytokine responses:** The original Parse PBMC screen profiles approximately ten million cells from twelve human donors under PBS control and 90 cytokine treatments (Parse Biosciences, 2025). We train a separate model for each leave-one-donor-out fold and evaluate all 90 cytokines per fold, yielding 1,080 donor–cytokine prediction settings. The held-out donor's PBS cells provide control inputs, while their stimulated cells are excluded from training. This large-scale setting tests whether responses observed in other donors can inform the target donor's transcriptional response given only their control cells (Klein et al., 2025). **(2) Unseen drug responses:** SciPlex3 measures single-drug treatments at multiple doses across three cancer cell lines (Srivatsan et al., 2020). We hold out complete drug identities across all doses and cell lines to test generalization to unseen compounds. Dataset preprocessing, split construction, and condition definitions are provided in Appendix A.1.

Baselines. We compare scAMod with control sampling, chemCPA (Hetzel et al., 2022), CellFlow (Klein et al., 2025), scGPT (Cui et al., 2024), and STATE (Adduri et al., 2026). All methods use the same split setting and are evaluated in the same gene space under a shared evaluation protocol. Implementation details for each baseline are provided in Appendix A.3.

Evaluation. We assess predictions over all modeled genes (**Full**) and differentially expressed genes defined for each perturbation condition (**DEG**). Full scores evaluate the overall response, whereas DEG scores focus on perturbation-responsive genes whose changes may be obscured in the full profile. We compute R^2 and mean squared error (**MSE**) on pseudobulk mean expression profiles, and energy distance (**ED**) and maximum mean discrepancy (**MMD**) between observed and predicted cell populations. Definitions and aggregation details are provided in Appendix A.2.

4.2 Cross-Donor Perturbation Generalization

Predicting a held-out donor's response requires transferring cytokine effects across heterogeneous cell populations while retaining donor-specific cellular context. In Table 1, control sampling outperforms all evaluated learned baselines on DEG R^2 and DEG ED, despite making no cytokine-specific adjustment. Improving beyond this no-response reference is therefore a demanding test of cross-donor response prediction. Relative to control sampling, scAMod improves DEG R^2 from 0.7602 to 0.826 (8.7%) and reduces DEG ED from 1.1006 to 0.8339 (24.2%). It also achieves the lowest MMD in both Full and DEG gene spaces, and results for each of the 12 held-out donors are provided in Appendix Table 7. Representative expression-space comparisons are shown in Appendix Figure 3. These results point toward precision medicine by supporting the prediction of heterogeneous cellular responses across individuals, beyond population-average effects.

Table 1: Full and DEG performance comparison on the PBMC donor leave-out benchmark. The best result in each column is shown in bold and the second best is underlined.

	R^2 ↑		ED ↓		MMD ↓		MSE ↓	
Model	Full	DEG	Full	DEG	Full	DEG	Full	DEG
Ctrl sampling	<u>0.9501</u>	<u>0.7602</u>	**1.0043**	<u>1.1006</u>	0.0115	<u>0.01887</u>	<u>0.01664</u>	<u>0.36227</u>
chemCPA	0.8197	0.5016	14.8829	6.9743	0.2043	0.2434	0.05963	0.7375
CellFlow	0.9029	0.7389	4.4313	1.8258	0.02051	0.04907	0.02359	0.38933
scGPT	0.4434	-0.2781	24.49	10.42	0.7797	0.8572	0.2001	2.115
State	0.7305	0.3255	4.4414	2.954	<u>0.00897</u>	0.03733	0.0971	0.98653
scAMod	**0.9670**	**0.826**	<u>1.2665</u>	**0.8339**	**0.0083**	**0.01544**	**0.01532**	**0.25396**

4.3 Generalization to Unseen Drug Perturbations

On held-out SciPlex3 drugs, scAMod leads on seven of the eight Full and DEG metrics (Table 2). It improves DEG R^2 from 0.2997 for the strongest baseline to 0.4364 and achieves the lowest Full and DEG ED and MMD, while ranking second on Full MSE. These complementary gains support more accurate prediction of both perturbation-responsive gene means and cell-population distributions. Because complete drug identities are withheld across doses and cell lines, the results also support the use of distributional perturbation models for in silico screening of compounds without measured single-cell responses. Results across the five random seeds are reported in Appendix Table 9.

Table 2: Full and DEG performance comparison on the SciPlex3 held-out drug benchmark. The scAMod row reports the mean over five random seeds. The best result in each column is shown in bold and the second best is underlined.

	R^2 ↑		ED ↓		MMD ↓		MSE ↓	
Model	Full	DEG	Full	DEG	Full	DEG	Full	DEG
Ctrl sampling	0.8126	-0.027	0.906	1.087	0.0288	0.0557	0.0034	0.0978
CellFlow	0.8502	0.1396	1.015	0.96	0.0317	0.0557	0.00267	0.0764
chemCPA	0.8351	0.2241	3.15	1.833	0.2872	0.3038	0.00303	0.0895
scGPT	<u>0.8607</u>	0.1693	4.324	2.267	0.6931	0.5174	**0.00245**	<u>0.0677</u>
State	0.8419	<u>0.2997</u>	<u>0.794</u>	<u>0.852</u>	<u>0.0268</u>	<u>0.0461</u>	0.00285	0.0724
scAMod	**0.8672**	**0.4364**	**0.7146**	**0.7373**	**0.02481**	**0.04157**	<u>0.00253</u>	**0.06222**

4.4 Contextual Module Analysis

Shared modules provide interpretable functional semantics. We test whether the prior modules organize functionally related genes through pathway enrichment. In Figure 2**A**, each bubble denotes a significant module–pathway association, with larger bubbles indicating stronger statistical evidence. Among 27 modules with less than 30% non-coding or pseudogene content, 18 have at least one significantly enriched pathway, including B-cell receptor signaling, antigen presentation, cell cycle, and metabolism. The panel displays significantly enriched modules, illustrating the func-

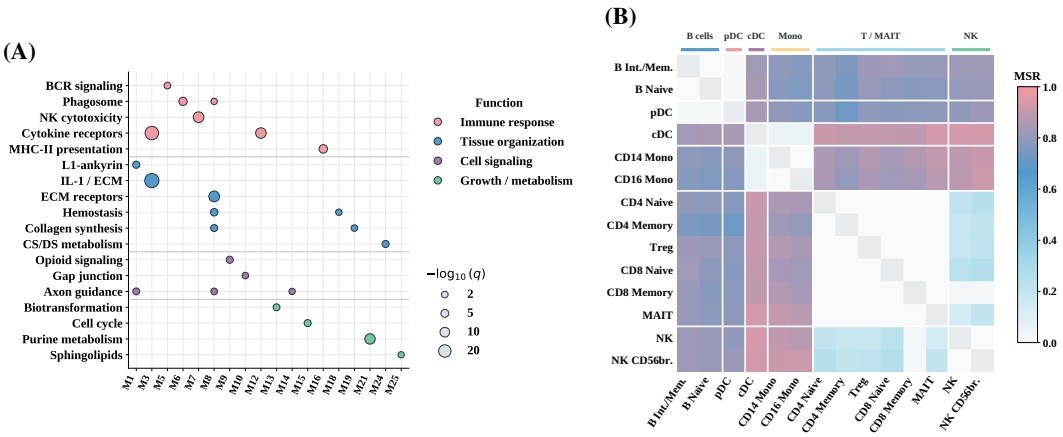


Figure 2: **Semantic module prior and adaptive routing.** **(A)** BioPlanet pathway enrichment of prior modules. Color denotes functional groups and bubble area denotes $-\log_{10}(q)$. **(B)** Pairwise module-switch rate (MSR) across cell types in held-out Donor1 PBS cells.

tional reference provided by shared module semantics. Gene-set construction, pathway references, and enrichment tests are detailed in Appendix B.3. Figure 4 reports all 32 modules.

Context shapes gene participation across modules. Figure 2**B** shows the fraction of comparable genes whose dominant module differs between two cell types. Higher values indicate more changes in gene participation. Contextual adaptation should preserve relative stability in similar backgrounds while accommodating different compositions in distinct backgrounds. Consistent with this expectation, CD4 Naive and CD8 Naive have no switches among 101 comparable genes, whereas B Naive and CD14 monocytes differ for 52 of 67 genes (77.6%). Similar context-dependent routing occurs across five held-out donors (Table 8). Gene-level soft-weight examples in Figure 5 further illustrate how the same gene participates in different modules across contexts. Together, these results support shared module semantics with context-dependent composition. Routing aggregation, stability criteria, and switch-rate calculations are provided in Appendix B.3.

Condition-guided readout focuses on perturbation-related modules. The PMR weight plot in Figure 6 shows the relative attention assigned to modules, with wider segments indicating greater weight in condition-guided readout. In the selected IFN-γ, IFN-α1, and M-CSF cases, M16 receives the highest weight and its prior genes are significantly enriched for MHC-II antigen presentation. Previous studies link these cytokines to MHC-II or HLA-DR expression in monocytes and monocyte-derived macrophages (Lee et al., 2017; Gkirtzimanaki et al., 2018; Hartigan-O'Connor et al., 2014; Kandalla et al., 2023). These qualitative examples illustrate agreement between condition-guided readout and known perturbation responses. Weight extraction, case selection, and enrichment procedures are described in Appendix B.3.

Shared functional semantics, contextual gene participation, and perturbation-guided readout thus provide complementary explanations of the module-centered design, whose predictive value is supported by the main results and component ablations.

4.5 Ablation Studies

Expression reconstruction. We assess the expression reconstruction error of the module-centered bottleneck used for perturbation prediction. At a matched bottleneck dimension of 128, the autoencoder achieves a cell-level reconstruction MSE of 0.00483 on held-out SciPlex3 cells, compared with 0.00914 for PCA, a 47.2% reduction (Appendix B.4). This lower reconstruction error supports preserving gene-level expression information through the compression and decoding stages of the perturbation prediction pipeline.

Module components. We first ablate the module backbone on SciPlex3 without PMR to isolate its effects from the downstream readout. The variants fix gene–module routing across cells, shuffle

genes in the prior while preserving module masses, or remove the routing anchor loss. The complete backbone leads on all eight metrics in Table 3. The static-routing variant has the largest ED and MMD, whereas the variant without anchor loss has the lowest DEG R^2. Shuffling the prior reduces distributional agreement, suggesting that biologically informed module semantics matter beyond module size alone. The routing and anchor-loss ablations further support combining context-dependent module composition with a shared semantic reference for response modeling.

Table 3: Module component ablation on SciPlex3. All variants exclude PMR; the full row therefore differs from the complete scAMod model in Table 2.

	R^2 ↑		ED ↓		MMD ↓		MSE ↓	
Variant	Full	DEG	Full	DEG	Full	DEG	Full	DEG
Full (without PMR)	**0.8522**	**0.1508**	**0.7371**	**0.8364**	**0.02539**	**0.04672**	**0.002606**	**0.07071**
w/o adaptive module	0.8434	0.0849	1.3543	0.9952	0.05278	0.06846	0.002786	0.07663
w/o prior	0.8499	0.1449	1.1903	0.9226	0.04413	0.05904	0.002691	0.07359
w/o anchor loss	0.8409	0.0709	0.8082	0.9243	0.02751	0.04955	0.002844	0.08059

Perturbation module readout. We add PMR to the same frozen autoencoder and flow generator, training it with DEG MSE, ED, or both (Table 4). DEG MSE alone yields the strongest DEG scores but degrades all Full metrics relative to the base model. ED alone improves Full-profile prediction but yields a smaller gain in DEG R^2. Combining both objectives improves all eight metrics over the base model. These results show that full-profile distributional agreement and perturbation-responsive gene accuracy provide complementary training signals. Their combination enables condition-guided module readout to capture response-specific changes while retaining the broader expression profile.

Table 4: Additive PMR objective comparison on SciPlex3 using a common frozen generator. Base omits PMR; each "+" row adds PMR trained with the indicated objective. The combined row is the matched ablation reference, whereas Table 2 reports the five-seed mean. The best result in each column is shown in bold and the second best is underlined.

	R^2 ↑		ED ↓		MMD ↓		MSE ↓	
Variant	Full	DEG	Full	DEG	Full	DEG	Full	DEG
Base (no PMR)	0.8522	0.1508	0.7371	0.8364	0.02539	0.04672	0.002606	0.07071
+ DEG MSE	0.8178	**0.4883**	0.8675	**0.7269**	0.02829	**0.04097**	0.003246	**0.06158**
+ ED	<u>0.8574</u>	0.176	**0.7065**	0.8134	**0.02456**	0.04602	**0.002503**	0.06837
+ DEG MSE + ED	**0.8672**	<u>0.4364</u>	<u>0.7146</u>	<u>0.7373</u>	<u>0.02478</u>	<u>0.04172</u>	<u>0.00253</u>	<u>0.06221</u>

5 Conclusion

We introduced scAMod, a module-centered framework for distributional single-cell perturbation prediction that combines shared gene relationships with cell-context-adaptive module composition. The framework generates perturbed cell populations while retaining gene-level readout under each condition. Experiments on unseen drug perturbations and donor leave-out cytokine responses show improvements in distributional and DEG-level prediction, with ablations supporting the contributions of adaptive modules and perturbation module readout. The routing and readout analyses provide associative interpretations rather than establishing causal gene or pathway mechanisms. Future work could examine whether the learned adaptive module organization transfers across broader perturbation types, cellular contexts, and related single-cell prediction tasks.

AI use statement

Generative AI tools were used for code organization, language polishing, and assistance with literature searches. All resulting content was independently reviewed and confirmed by the authors.

Ethics statement

This work uses only publicly available datasets and involves no new human-subject experiments or data collection. Additional ethical approval is therefore not applicable.

Reproducibility statement

Dataset preprocessing, model configurations, training procedures, and evaluation protocols are described in Appendix A. Code is available at `https://anonymous.4open.science/r/scAMod-A775`.

References

Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Mohsen Naghipourfar, Alishba Imran, Rohan Shah, Noam Teyssier, Rishi Verma, Christopher Carpenter, Basak Eraslan, Francis Chalissery, Rajesh Ilango, Vishvak Subramanyam, Chiara Ricci-Tam, Sanjay Nagaraj, Aidan Winters, Mingze Dong, Stefanie Fellinger, Adam Krejci, Tilmann Burckstummer, Sravya Tirukkovular, Jeremy Sullivan, Brian S. Plosky, Nicholas D. Youngblut, Jure Leskovec, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Alexander Dobin, Dave P. Burke, Hani Goodarzi, and Yusuf H. Roohani. Predicting cellular responses to perturbation across diverse contexts with state. *Cell*, August 2026. ISSN 0092-8674. doi: 10.1016/j.cell.2026.07.052. URL `http://dx.doi.org/10.1016/j.cell.2026.07.052`.

Charlotte Bunne, Andreas Krause, and Marco Cuturi. Supervised training of conditional monge maps. *Advances in Neural Information Processing Systems*, 35:6859–6872, 2022.

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

Changxi Chi, Yufei Huang, Jun Xia, Jiangbin Zheng, Yunfan Liu, Zelin Zang, and Stan Z Li. Departures: Distributional transport for single-cell perturbation prediction with neural schrödinger bridges. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20544–20552, 2026.

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

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

Mingze Dong, Bao Wang, Jessica Wei, Antonio H de O. Fonseca, Curtis J Perry, Alexander Frey, Feriel Ouerghi, Ellen F Foxman, Jeffrey J Ishizuka, Rahul M Dhodapkar, et al. Causal identification of single-cell experimental perturbation effects with cinema-ot. *Nature methods*, 20(11): 1769–1779, 2023.

Katerina Gkirtzimanaki, Eleni Kabrani, Dimitra Nikoleri, Alexander Polyzos, Athanasios Blanas, Prodromos Sidiropoulos, Antonis Makrigiannakis, George Bertsias, Dimitrios T Boumpas, and Panayotis Verginis. Ifnα impairs autophagic degradation of mtdna promoting autoreactivity of sle monocytes in a sting-dependent fashion. *Cell reports*, 25(4):921–933, 2018.

Dennis J Hartigan-O'Connor, Din Lin, James C Ryan, Valentina A Shvachko, Myrna L Cozen, Mark R Segal, Norah A Terrault, Lewis L Lanier, M Michele Manos, and Joseph M McCune. Monocyte activation by interferon α is associated with failure to achieve a sustained virologic response after treatment for hepatitis c virus infection. *The Journal of infectious diseases*, 209 (10):1602–1612, 2014.

Leon Hetzel, Simon Boehm, Niki Kilbertus, Stephan Günnemann, Fabian Theis, et al. Predicting cellular responses to novel drug perturbations at a single-cell resolution. *Advances in Neural Information Processing Systems*, 35:26711–26722, 2022.

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

Prashanth K Kandalla, Julien Subburayalu, Clement Cocita, Berengere de Laval, Elena Tomasello, Johanna Iacono, Jessica Nitsche, Maria M Canali, Wilfried Cathou, Gilles Bessou, et al. M-csf directs myeloid and nk cell differentiation to protect from cmv after hematopoietic cell transplantation. *EMBO Molecular Medicine*, 15(11):EMMM202317694, 2023.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, et al. Cellflow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, pp. 2025–04, 2025.

Russell Z Kunes, Thomas Walle, Max Land, Tal Nawy, and Dana Pe'er. Supervised discovery of interpretable gene programs from single-cell data. *Nature biotechnology*, 42(7):1084–1095, 2024.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Justin Lee, Hanson Tam, Lital Adler, Alexandra Ilstad-Minnihan, Claudia Macaubas, and Elizabeth D. Mellins. The mhc class ii antigen presentation pathway in human monocytes differs by subset and is regulated by cytokines. *PLOS ONE*, 12(8):e0183594, 2017. doi: 10.1371/journal.pone.0183594.

Jiahao Li, Jiayi Dong, Peng Ye, Xiaochi Zhou, Haohai Lu, and Fei Wang. SAVE: A generalizable framework for multi-condition single-cell generation with gene block attention. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=l7QEoK4uDP.

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

Mohammad Lotfollahi, Sergei Rybakov, Karin Hrovatin, Soroor Hediyeh-Zadeh, Carlos Talavera-López, Alexander V Misharin, and Fabian J Theis. Biologically informed deep learning to query gene programs in single-cell atlases. *Nature Cell Biology*, 25(2):337–350, 2023.

Djordje Miladinovic, Tobias Höppe, Mathieu Chevalley, Andreas Georgiou, Lachlan Stuart, Arash Mehrjou, Marcus Bantscheff, Bernhard Schölkopf, and Patrick Schwab. In silico biological discovery with large perturbation models. *Nature Computational Science*, 5(11):1029–1040, 2025.

Parse Biosciences. 10 million human PBMCs in a single experiment. https://www.parsebiosciences.com/datasets/10-million-human-pbmcs-in-a-single-experiment/, 2025. Dataset webpage. Accessed: 2025-01-29.

Yinhua Piao, Hyomin Kim, Seonghwan Kim, Yunhak Oh, Junhyeok Jeon, Sang-Yeon Hwang, Jaechang Lim, Woo Youn Kim, Chanyoung Park, and Sungsoo Ahn. Learning adaptive perturbation-conditioned contexts for robust transcriptional response prediction. In *ICLR 2026 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2026. URL https://openreview.net/forum?id=Vnx6ZyV6cw.

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

Liujia Qian, Zizhuo Zhou, Peijie Zhou, Zhen Dong, Xudong Zhang, Zhenwu Dai, Zhangyang Gao, Siqi Sun, Kevin R Roy, Shuaiyao Wang, et al. Towards the construction of a virtual yeast. *Nature*, 655(8121):59–70, 2026.

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

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature biotechnology*, 42(6):927–935, 2024.

Lucas Seninge, Ioannis Anastopoulos, Hongxu Ding, and Joshua Stuart. Vega is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nature communications*, 12(1):5684, 2021.

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

Chang Su, Zichun Xu, Xinning Shan, Biao Cai, Hongyu Zhao, and Jingfei Zhang. Cell-type-specific co-expression inference from single cell rna-sequencing data. *Nature Communications*, 14(1): 4846, 2023.

Bruno M Tesson, Rainer Breitling, and Ritsert C Jansen. Diffcoex: a simple and sensitive method to find differentially coexpressed gene modules. *BMC bioinformatics*, 11(1):497, 2010.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

Chloe Wang, Mehran Karimzadeh, Neal G Ravindra, Lexi R Bounds, Nader Alerasool, Ann C Huang, Shihao Ma, Daniel R Gulbranson, Haotian Cui, Yongju Lee, et al. X-cell: Scaling causal perturbation prediction across diverse cellular contexts via diffusion language models. *bioRxiv*, pp. 2026–03, 2026.

Yuanfang Xiang and Lun Ai. Adaptive data-knowledge alignment in genetic perturbation prediction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CxLaZWbUjc.

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

Chenglei Yu, Chuanrui Wang, Bangyan Liao, and Tailin Wu. scDFM: Distributional flow matching model for robust single-cell perturbation prediction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QSGanMEcUV.

A Additional Experimental Details

A.1 Datasets and Splits

Parse PBMC Cytokine Perturbation Data. We analyze the Parse Biosciences PBMC cytokine screen, which includes PBS controls and 90 cytokine treatments across 12 donors (Parse Biosciences, 2025). The count matrix is restricted to 2,000 highly variable genes identified with Scanpy's `highly_variable_genes`. Counts are then normalized to a total of 10,000 per cell and transformed with $\log(1 + x)$. Donor, cytokine, and cell-type annotations are retained for evaluation. We evaluate all 90 non-PBS cytokines in 12 leave-one-donor-out folds, for 1,080 donor–cytokine conditions. In each fold, the held-out donor's PBS cells remain available as control inputs, whereas none of that donor's stimulated cells are used for training. We use this dataset to evaluate perturbation prediction across heterogeneous donor backgrounds.

SciPlex3 Chemical Perturbation Data. We use the sciPlex3 single-drug screen across three cancer cell lines and multiple drug doses (Srivatsan et al., 2020). Drug–dose–cell-line conditions with fewer than 100 cells are excluded. Counts are then normalized per cell with Scanpy's `normalize_total`, transformed with `log1p`, and filtered to 2000 highly variable genes. Training and validation contain 179 drug identities and the test split holds out nine other drugs at every measured dose and in every cell line. Its 83 treated conditions enter the main-table average, with three cell-line-matched vehicle controls providing source populations. We use this dataset to evaluate generalization to unseen perturbations.

Differentially Expressed Genes. DEGs are ranked by comparing observed perturbed cells with matched controls. For SciPlex3, the top 50 genes are selected for each cell-line–drug–dose condition. For PBMC, a Wilcoxon test ranks genes within each donor–cytokine–cell-type stratum relative to PBS controls, and the top 50 are retained. For PMR optimization, DEG masks are matched to conditions in the training split. Validation masks are used to compute validation losses. Held-out conditions have separate DEG sets for post hoc evaluation. These evaluation sets are not inputs to the generator, which predicts all modeled genes before either Full- or DEG-space scoring.

A.2 Evaluation Protocol and Metrics

We assess perturbation predictions across two complementary dimensions: evaluation space and statistical level. In evaluation space, metrics are computed over either all modeled genes (**Full**) or condition-specific differentially expressed genes (**DEG**), defined in Appendix A.1. At the statistical level, R^2 and **MSE** assess pseudobulk responses, whereas **MMD** and **ED** assess population distributions. Together, these comparisons evaluate mean response accuracy and distributional agreement across both gene spaces.

For a fixed perturbation condition, let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{y}'$ denote individual cell-expression vectors drawn from the matched control, observed perturbed, and predicted perturbed populations, respectively. Metrics use the G genes selected for Full or DEG evaluation.

Pseudobulk Metrics. The bar denotes the mean across cells in a population, so $\bar{\mathbf{y}}$ and $\bar{\mathbf{y}}'$ are the observed and predicted perturbed mean expression profiles. Let $m = G^{-1} \sum_{g=1}^{G} \bar{y}_g$ be the mean across genes in the observed profile. The mean squared error (**MSE**) and coefficient of determination (R^2) are

$$\mathrm{MSE} = \frac{1}{G} \|\bar{\mathbf{y}}' - \bar{\mathbf{y}}\|_2^2, \tag{20}$$

$$R^2 = 1 - \frac{\|\bar{\mathbf{y}}' - \bar{\mathbf{y}}\|_2^2}{\sum_{g=1}^{G} (\bar{y}_g - m)^2}. \tag{21}$$

The R^2 denominator is the squared error of predicting every observed gene mean with the same value m, rather than with the control profile. Neither metric pairs individual observed and predicted cells.

Distributional Metrics. Distributional metrics compare the populations of $\mathbf{y}'$ and $\mathbf{y}$ rather than their means. Let $\mathbf{y}_1, \mathbf{y}_2$ be independent observed cells and $\mathbf{y}'_1, \mathbf{y}'_2$ independent predicted cells; all four draws are independent, and subscripts 1 and 2 index cells. Following CellFlow (Klein et al., 2025) and Bunne et al. (2023), the maximum mean discrepancy (**MMD**) uses the Gaussian kernel $k_\gamma(\mathbf{a}, \mathbf{b}) = \exp(-\gamma\|\mathbf{a} - \mathbf{b}\|_2^2)$, and the energy distance (**ED**) uses Euclidean distances:

$$\mathrm{MMD}^2_\gamma = \mathbb{E}\, k_\gamma(\mathbf{y}_1, \mathbf{y}_2) + \mathbb{E}\, k_\gamma(\mathbf{y}'_1, \mathbf{y}'_2) - 2\,\mathbb{E}\, k_\gamma(\mathbf{y}_1, \mathbf{y}'_1), \tag{22}$$

$$\mathrm{ED} = 2\,\mathbb{E}\|\mathbf{y}_1 - \mathbf{y}'_1\|_2 - \mathbb{E}\|\mathbf{y}_1 - \mathbf{y}_2\|_2 - \mathbb{E}\|\mathbf{y}'_1 - \mathbf{y}'_2\|_2. \tag{23}$$

Neither metric requires paired observed and predicted cells. ED uses Euclidean rather than squared Euclidean distances.

We report the average of MMD^2_γ over $\gamma \in \{2, 1, 0.5, 0.1, 0.01, 0.005\}$, which covers both local and global differences between the populations. We refer to this averaged squared discrepancy as MMD.

For PBMC, metrics are averaged across cell types within each cytokine, then across 90 cytokines for each held-out donor, and finally across donors. For SciPlex3, metrics are averaged equally over the 83 held-out cell-line–drug–dose conditions.

A.3 Baseline Implementations

Control Sampling. Control sampling uses cells from the matched untreated population as predictions. It provides a no-response reference for assessing whether a model improves on the original control distribution.

chemCPA. chemCPA (Hetzel et al., 2022) is designed for predicting cellular responses to chemical compounds using molecular representations. Its encoder–decoder framework combines a basal cell representation with perturbation and covariate embeddings in a compositional latent space to predict expression. We adapt the perturbation encoding to the intervention type and available condition information in each dataset while retaining the compositional modeling framework. For PBMC, the baseline uses ESM-2 cytokine embeddings and the 300k-step chemCPA checkpoints.

CellFlow. CellFlow learns a condition-dependent velocity field through conditional flow matching to transport control cell populations toward perturbed distributions. Perturbation and covariate representations condition the transport dynamics. We use the PCA-based implementation, integrate the learned dynamics from control cell representations, and reconstruct the predicted expression profiles using the corresponding PCA transformation.

scGPT. scGPT represents genes and their expression values as input embeddings and uses a Transformer to model dependencies across genes. For perturbation prediction, we adapt its generation model by adding projected perturbation and covariate representations to the input embeddings, with condition encodings matched to each dataset. The resulting gene representations are decoded into predicted expression values.

STATE. STATE's transition model predicts perturbed cell populations from sets of control cells and perturbation representations. Its Transformer applies attention across cells to learn a set-to-set mapping, with a distributional objective comparing predicted and observed perturbed populations. We use the transition-model implementation with dataset-specific condition inputs and evaluate its gene-expression predictions under the shared protocol.

A.4 Gene Module Prior Construction

The construction proceeds from interaction-informed gene organization to a soft participation prior. We first smooth scFM gene embeddings over the PPI network while retaining their original semantic information. We use GeneCompass embeddings (Yang et al., 2024), whose pretraining integrates biological priors including gene regulatory and co-expression relationships, and retain high-confidence STRING v12.0 interactions with a combined score of at least 700 (on the 0–1000 scale). STRING protein identifiers are matched to modeled gene symbols through the release's preferred-name field; unmatched interactions are omitted. To establish interaction-informed gene groups, we apply Leiden clustering to a graph that combines neighborhood relationships in the smoothed embedding

space with PPI edges. To balance module sizes while preserving semantic affinity and community structure, we refine these groups through balanced OT, alternating capacity-constrained gene reassignment with module-center updates. The resulting balanced groups determine module centers in the gene embedding space. Finally, gene–center similarities are converted into soft participation weights through a temperature-scaled softmax. Thus, the balanced assignments define the module centers rather than restricting each gene to a single module in the final prior. This prior provides the shared reference for subsequent cell-specific adaptation.

Construction Procedure. Algorithm 1 summarizes the gene-to-module reassignment, which is distinct from the cell-level OT pairing used for flow training.

Algorithm 1 Gene Module Prior Construction

Input: gene embeddings $\mathbf{E}$, PPI edge weights $\mathbf{W}_{\text{ppi}}$, module capacities $\{n_k\}_{k=1}^{K}$, $\alpha, \beta, \lambda, \tau, L, T$.
Output: module centers $\{\mathbf{c}_k\}_{k=1}^{K}$ and soft gene–module prior $\mathbf{S}$.
$\mathbf{A}_{\text{ppi}} \leftarrow \text{RowNorm}(\mathbf{W}_{\text{ppi}} + \mathbf{I})$; $\mathbf{H}^{(0)} \leftarrow \text{norm}(\mathbf{E})$.
for $\ell = 0, \ldots, L-1$ **do**
  $\mathbf{H}^{(\ell+1)} \leftarrow \text{norm}(\alpha\mathbf{H}^{(0)} + (1-\alpha)\mathbf{A}_{\text{ppi}}\mathbf{H}^{(\ell)})$.
end for
$\mathbf{W} \leftarrow \text{kNN}(\mathbf{H}^{(L)}) + \beta\mathbf{W}_{\text{ppi}}$; $b \leftarrow \text{Leiden}(\mathbf{W}, K)$; $B_{gk} \leftarrow \mathbf{1}[b_g = k]$.
Initialize $\mathbf{c}_k$ as the normalized mean of $\{\mathbf{H}_g^{(L)} : b_g = k\}$.
for $t = 1, \ldots, T$ **do**
  $C_{gk} \leftarrow -(\mathbf{H}_g^{(L)})^\top \mathbf{c}_k - \lambda B_{gk}$.
  $z \leftarrow \text{BalancedAssign}(\mathbf{C}, \{n_k\}_{k=1}^{K})$.
  Update $\mathbf{c}_k$ from $\{\mathbf{H}_g^{(L)} : z_g = k\}$; **break** if assignments stabilize.
end for
Recompute $\mathbf{c}_k$ from $\{\mathbf{e}_g : z_g = k\}$ and set $S_{gk} \leftarrow \text{softmax}_k(\mathbf{e}_g^\top \mathbf{c}_k/\tau)$.
return $\{\mathbf{c}_k\}_{k=1}^{K}, \mathbf{S}$.

A.5 Model Architecture and Hyperparameters

Table 5 summarizes the module-prior, autoencoder, flow, and PMR settings used for each benchmark. The staged training and inference procedures are described in Appendix A.6. Explicit routing and PMR score definitions, together with component pseudocode, are provided in Appendix A.6.1.

Perturbation Encoding. The vector p encodes the intervention and its associated treatment and sample covariates. For SciPlex3, the inputs are 1,024-bit Morgan fingerprints, $\log_{10}$ dosages in nM, and precomputed cell-line embeddings. For PBMC, the inputs are precomputed ESM-2 cytokine embeddings and donor-specific mean expression profiles of PBS cells (Klein et al., 2025). Separate learnable projections encode these inputs, which are concatenated, aggregated by attention pooling, and projected to obtain p. The output dimensions are 1024 for SciPlex3 and 256 for PBMC. The molecular and protein input features remain fixed. The resulting vector conditions both the latent flow and PMR readout. Constructing p at inference does not require observed perturbed test-cell expression.

A.6 Training Schedule and Inference

Staged Optimization Schedule. Training has three sequential stages, matching Section 3.5. The adaptive module autoencoder learns expression reconstruction and the query-alignment and anchor objectives. Its trained tokenizer, encoder, module decoder, and gene decoder are then held fixed while the condition encoder and latent velocity field learn OT-coupled control-to-perturbed transitions. Finally, the autoencoder and flow, including the condition encoder, are frozen. PMR reuses the encoded condition p and learns to read the generated module states under that condition. No gradients from PMR are propagated into the first two stages. The SciPlex3 autoencoder configuration specifies 300 epochs; PBMC flow runs reuse donor-specific trained autoencoder checkpoints rather than retraining the autoencoder during the final all-cytokine evaluation.

Table 5: Architecture and hyperparameter configurations of scAMod.

Component	Hyperparameter	SciPlex3	PBMC
Gene module prior	Number of modules	32	32
	PPI propagation steps / mixing weight	2 / 0.65	2 / 0.65
	Semantic graph neighbors / PPI edge weight	30 / 0.5	30 / 0.5
	Leiden resolution	2.81	2.44
	Balanced OT iterations / initial-label bonus	5 / 0.25	5 / 0.25
	Prior temperature	0.25	0.25
Adaptive Module Tokenizer	Router dimension / hidden width	64 / 128	64 / 128
	Gene value dimension	32	32
	Routing temperature / residual initialization	1 / 0.01	1 / 0.01
Module encoder and decoder	Attention layers / inducing tokens	4 / 32	4 / 32
	Attention heads / embedding dimension	8 / 384	8 / 384
	Bottleneck dimension	128	128
	Query-alignment / anchor-loss weight	0.01 / 0.0005	0.01 / 0.0005
Conditional flow	Condition embedding dimension	1024	256
	Velocity field hidden widths	4096×4	2048×3
PMR	Condition attention / readout hidden dimensions	64 / 16	64 / 16
	Attention temperature / prior scale	0.5 / 0.25	0.5 / 0.25

Table 6: Training and inference settings for the reported SciPlex3 and PBMC models. Batch sizes are global across two GPUs for AE and flow; PMR uses one GPU.

Setting	SciPlex3	PBMC leave-donor-out
AE: learning rate / batch size	1.5×10^{-4} / 512	1.5×10^{-4} / 512
AE: training / checkpoint	300 epochs; best validation reconstruction	100 epochs; best validation reconstruction
Flow: learning rate / batch size	5×10^{-5} / 1024	5×10^{-5} / 1024
Flow: training / checkpoint	245 condition epochs ($\sim$500k updates); final epoch	500k updates; best validation R^2
Flow: minibatch OT	Latent-space Sinkhorn within cell line, squared-Euclidean cost, $\varepsilon = 1.0$	Latent-space Sinkhorn within donor, squared-Euclidean cost, $\varepsilon = 0.5$
PMR: learning rate / batch size	2×10^{-4} / 2 conditions $\times$ 16 cells	2×10^{-4} / 2 conditions $\times$ 16 cells
PMR: training / checkpoint	50 epochs; best validation loss	50 epochs; final checkpoint
PMR: objective / ED weight	DEG-mean MSE + nonsquared ED; 0.003	DEG-mean MSE + nonsquared ED; 0.5
Inference: flow integration	Diffrax Tsit5, $t \in [0, 1]$, adaptive steps, $\mathrm{rtol} = \mathrm{atol} = 10^{-5}$	

Model Selection. For SciPlex3, we select hyperparameters using the validation split. For PBMC, pilot experiments on donor 1 determine a shared hyperparameter configuration, which is then applied unchanged to the remaining donor folds. Donor 1 therefore informs PBMC model selection.

Inference. For each test condition, matched control cells are encoded into the fixed latent space, transported by integrating the learned conditional velocity field from control states, and decoded into generated module states. PMR reads these states under the test condition to produce expression profiles. PBMC predictions use a deterministic sample of at most 10,000 control cells per held-out donor; evaluation then compares the generated and observed cells within eligible source-control cell-type strata. No observed perturbed test-cell expression is supplied to the generator.

Compute Resources. The local compute node is equipped with Intel Xeon Platinum 8362 CPUs, one NVIDIA A100-SXM4 GPU with 80 GB of memory, and approximately 227 GiB of host memory.

A.6.1 Computational Procedures

Algorithms 2–4 detail the three stages using the notation of Section 3. Training steps use observed cells from the training split. Generation uses only control cells and the requested perturbation condition. The soft prior $\mathbf{S}$ is fixed throughout these stages.

Adaptive Routing and Autoencoder. Let r be the routing dimension. A bias-free projection $\mathbf{W}$ projects the gene embedding, followed by layer normalization. For the cell context, layer-normalized expression passes through a two-layer projection f_{ctx} with a GELU activation and dropout between the layers. Its Kr outputs are reshaped into one r-dimensional context vector per module. Thus, the adjustment in Equation 4 is

$$[f_{\mathrm{res}}(\mathbf{e}_g, \mathbf{x})]_k = \frac{\mathrm{LayerNorm}(\mathbf{W}\mathbf{e}_g)^\top \left[\mathrm{reshape}(f_{\mathrm{ctx}}(\mathrm{LayerNorm}(\mathbf{x})), K, r)\right]_k}{\sqrt{r}}. \tag{24}$$

Routing dimensions and context-projection hidden widths are listed in Table 5.

Algorithm 2 Adaptive Gene Module Autoencoder

Input: training expression vectors, gene embeddings $\{\mathbf{e}_g\}$, fixed prior $\mathbf{S}$, and loss weights.
Output: trained tokenizer, encoder, module decoder, and gene query decoder.
for each training minibatch $\mathcal{B}$ **do**
  for each $\mathbf{x} \in \mathcal{B}$ **do**
    Compute $f_{\mathrm{res}}(\mathbf{e}_g, \mathbf{x})$ by Equation 24.
    $A_{gk}(\mathbf{x}) \leftarrow \mathrm{softmax}_k(\log S_{gk} + \eta[f_{\mathrm{res}}(\mathbf{e}_g, \mathbf{x})]_k)$.
    Aggregate $x_g \mathbf{v}_g$ with $A_{gk}(\mathbf{x})$ and module-mass normalization to obtain $\mathbf{m}_k$ (Equation 5).
    $\mathbf{z} \leftarrow f_{\mathrm{enc}}([\mathbf{m}_1; \ldots; \mathbf{m}_K])$; $[\mathbf{m}'_1; \ldots; \mathbf{m}'_K] \leftarrow f_{\mathrm{dec}}(\mathbf{z})$.
    Compute gene query weights $\mathbf{Q}(\mathbf{x})$ and reconstruct $\hat{\mathbf{x}}$ using Equations 6–7.
  end for
  Compute reconstruction MSE and query alignment $\mathrm{KL}(\mathrm{stopgrad}[\mathbf{A}] \| \mathbf{Q})$.
  Average routing over $\mathcal{B}$ and compute the anchor loss $\mathbb{E}_g \mathrm{KL}(\bar{\mathbf{A}}_{g,:} \| \mathbf{S}_{g,:})$.
  Update autoencoder parameters using $\mathcal{L}_{\mathrm{AE}}$ (Equation 12).
end for
return trained autoencoder components.

Conditional Flow Training and Generation. The tokenizer and autoencoder are frozen. Within each matched training context, latent-space Sinkhorn OT uses squared Euclidean costs and the regularization settings in Table 6. The probability path uses linear interpolation with constant Gaussian noise of scale σ. Its target velocity is $\mathbf{z}_1 - \mathbf{z}_0$. At inference, the learned velocity field is integrated from encoded control cells, without OT pairing to observed test responses.

Algorithm 3 Module Latent Flow Matching

Input: frozen autoencoder, training control and perturbed populations, perturbation embeddings p, OT settings, and noise scale σ.
Output: trained conditional velocity field $\mathbf{v}_\theta$ and generated latent states.
for each training update **do**
  Sample a training condition and matched control and perturbed minibatches.
  Encode both minibatches with the frozen tokenizer and f_{enc}.
  Compute $\pi_p \leftarrow \mathrm{Sinkhorn}(\{\|\mathbf{z}_0 - \mathbf{z}_1\|_2^2\}, \text{OT settings})$.
  Sample latent pairs $(\mathbf{z}_0, \mathbf{z}_1) \sim \pi_p$, $t \sim \mathcal{U}(0,1)$, and $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
  $\mathbf{z}_t \leftarrow (1-t)\mathbf{z}_0 + t\mathbf{z}_1 + \sigma\boldsymbol{\xi}$.
  Update flow parameters to minimize $\|\mathbf{v}_\theta(t, \mathbf{z}_t, p) - (\mathbf{z}_1 - \mathbf{z}_0)\|_2^2$, averaged over sampled pairs.
end for
Generation: encode each matched control cell to obtain $\mathbf{z}_0$.
Integrate $d\mathbf{z}_t/dt = \mathbf{v}_\theta(t, \mathbf{z}_t, p)$ from $t = 0$ to $t = 1$, initialized at $\mathbf{z}_0$.
return generated latent states for module decoding and PMR readout.

PMR Projections and Readout. The two score operators in Equation 10 use four independent, bias-free learned projections $\mathbf{W}_1, \ldots, \mathbf{W}_4$, each with output dimension d. Writing norm for ℓ_2 normalization of a vector, their exact computations are

$$\langle \mathbf{q}_g, \mathbf{m}'_k \rangle_{\mathrm{att}} = \sqrt{d}\ \mathrm{norm}(\mathbf{W}_1 \mathbf{q}_g)^\top \mathrm{norm}(\mathbf{W}_2 \mathbf{m}'_k), \tag{25}$$

$$\langle \mathbf{q}_g, \mathbf{m}'_k \rangle_{\mathrm{read}} = \frac{\mathrm{norm}(\mathbf{W}_3 \mathbf{q}_g)^\top \mathbf{W}_4 \mathbf{m}'_k}{\sqrt{d}}. \tag{26}$$

The condition gate similarly uses normalized learned projections of p and the frozen decoder anchors $\mathbf{a}_k$. Their dot products are multiplied by $\sqrt{d}$, divided by the gate temperature, and passed through a softmax over all K modules. The expression map f_{out} is a shared gene-wise two-layer projection with GELU, followed by a scaled $\tanh$ output; its output layer is zero-initialized. The frozen decoder supplies the gene queries and generated module states, while PMR learns the condition projections, the four score projections, and this expression map.

DEG Supervision and Distributional Loss. For each training condition, the DEG set is fixed before PMR optimization and restricts only the pseudobulk MSE term. The predicted minibatch mean is compared with a precomputed observed mean for that training condition, rather than a newly sampled observed minibatch mean. Energy-distance loss uses all modeled genes in the sampled predicted and observed training populations, with the loss computed as

$$\mathcal{L}_{\mathrm{ED}} = \mathbb{E}\|\mathbf{y}' - \mathbf{y}\|_2 - \tfrac{1}{2}\mathbb{E}\|\mathbf{y}'_1 - \mathbf{y}'_2\|_2 - \tfrac{1}{2}\mathbb{E}\|\mathbf{y}_1 - \mathbf{y}_2\|_2. \tag{27}$$

Expectations are estimated by uniform averages over all cross- and within-population pairs in each training minibatch, including self pairs. For numerical stability, Euclidean distances are computed as $\sqrt{\max(\|\mathbf{a} - \mathbf{b}\|_2^2, 10^{-8})}$. Neither energy-distance loss nor generation requires a test-condition DEG set. Held-out evaluation responses and their DEG sets do not enter the generation step. They are used afterward to score the predicted population.

Algorithm 4 Perturbation Module Readout

Input: frozen autoencoder and flow, fixed prior $\mathbf{S}$, control cells $\mathcal{X}$, perturbation embedding p, and trainable PMR parameters.
Output: full-gene predicted population $\mathcal{Y}'$.
$\mathbf{g}_p \leftarrow \mathrm{CrossAttn}(p, \{\mathbf{a}_k\}_{k=1}^{K})$ over all modules.
for each control cell $\mathbf{x} \in \mathcal{X}$ **do**
  Encode and transport $\mathbf{x}$ using the frozen autoencoder and flow (Algorithm 3).
  Decode generated tokens $\{\mathbf{m}'_k\}$ and base expression $\widetilde{\mathbf{y}}$.
  Compute attention and readout scores with the independent projections in Equation 26.
  $R_{gk} \leftarrow \mathrm{softmax}_k(\langle \mathbf{q}_g, \mathbf{m}'_k \rangle_{\mathrm{att}} + \rho \log S_{gk} + \log(g_{p,k} + \epsilon))$.
  $h_g \leftarrow \sum_k R_{gk} \langle \mathbf{q}_g, \mathbf{m}'_k \rangle_{\mathrm{read}}$; $\mathbf{y}' \leftarrow \widetilde{\mathbf{y}} + f_{\mathrm{out}}(\mathbf{h})$.
end for
Collect generated profiles into $\mathcal{Y}'$.
Training only: compare a generated minibatch with observed cells from the same training condition.
Compare the predicted minibatch mean with the precomputed observed mean on that training condition's DEG set to obtain $\mathcal{L}_{\mathrm{MSE}}^{\mathrm{DEG}}$.
Compute $\mathcal{L}_{\mathrm{ED}}$ on all modeled genes by Equation 27.
Update only PMR parameters using $\mathcal{L}_{\mathrm{MSE}}^{\mathrm{DEG}} + \lambda_{\mathrm{ED}} \mathcal{L}_{\mathrm{ED}}$.
Inference: return $\mathcal{Y}'$ from steps 3–11, without observed perturbed cells or DEG masks.

B Additional Results and Analysis

B.1 PBMC Results by Held-Out Donor

Table 7 reports scAMod performance for each held-out donor. Within each donor, metrics are averaged over eligible cell types and then over all 90 cytokines, following the main evaluation protocol.

Table 7: scAMod performance for each held-out PBMC donor.

	$R^2 \uparrow$		ED ↓		MMD ↓		MSE ↓	
Donor	Full	DEG	Full	DEG	Full	DEG	Full	DEG
1	0.9665	0.7925	1.3904	1.0754	0.00830	0.01847	0.01631	0.32859
2	0.9681	0.8568	1.1569	0.7042	0.00858	0.01450	0.01477	0.20724
3	0.9626	0.8143	1.5908	0.9332	0.00863	0.01708	0.01741	0.28196
4	0.9589	0.7572	1.3760	0.9218	0.00769	0.01614	0.01673	0.28284
5	0.9677	0.7831	1.5371	1.0892	0.01081	0.02022	0.01833	0.33087
6	0.9605	0.8559	1.1677	0.7163	0.00698	0.01336	0.01411	0.21515
7	0.9601	0.8426	0.8813	0.7354	0.00603	0.01152	0.01252	0.24207
8	0.9705	0.8597	1.0285	0.6967	0.00791	0.01328	0.01339	0.21418
9	0.9647	0.8514	1.3367	0.7948	0.00891	0.01587	0.01628	0.23437
10	0.9687	0.8355	1.1574	0.7867	0.00827	0.01452	0.01435	0.24386
11	0.9785	0.8238	1.3884	0.8074	0.00900	0.01612	0.01478	0.23663
12	0.9767	0.8394	1.1863	0.7462	0.00843	0.01420	0.01484	0.22973
Donor macro	0.9670	0.8260	1.2665	0.8339	0.00830	0.01544	0.01532	0.25396

B.2 Distributional Visualization

For each displayed held-out donor–cytokine–cell-type condition, we fit a two-component PCA to sampled PBS and observed stimulated cells after normalization to 10,000 counts per cell and `log1p` transformation. Predictions from every method are transformed with this fixed PCA; all panels within a condition therefore share coordinates. The same 150 PBS source cells are used across methods, with 150 observed cells sampled separately. The projection illustrates population coverage but captures only 5.0–8.0% of the sampled real-cell variance across these conditions, so it is not a substitute for full-expression evaluation.

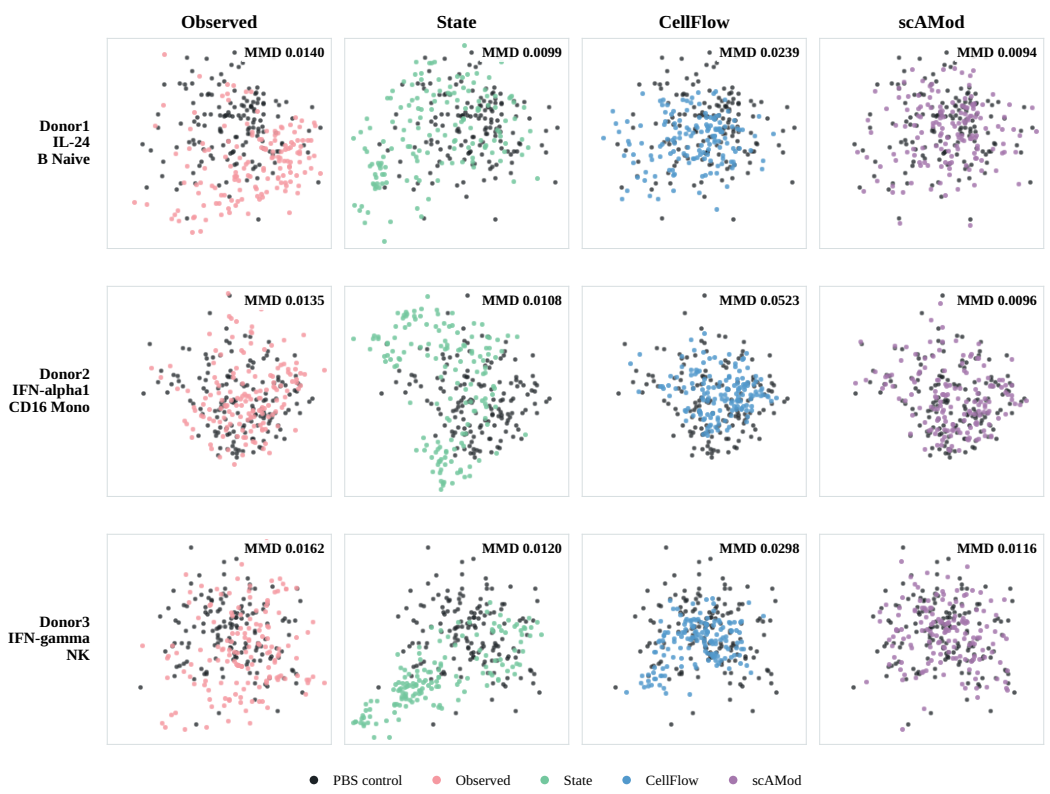


Figure 3: **Expression-space PCA of representative PBMC leave-donor-out predictions.** Each row shows one held-out donor–cytokine–cell-type condition. Dark points are the same PBS controls in every panel; colored points are observed or model-predicted stimulated cells. The PCA is fit to real cells only within each row. Insets report Full-expression MMD from the complete evaluation group, not from the displayed 150-cell sample.

B.3 Mechanistic Analysis Protocols and Metrics

Enrichment Protocol. We use the balanced assignment only to define gene sets for testing the semantic coherence of all 32 prior modules; the adaptive model itself retains soft gene–module participation. Each gene belongs to one balanced group for this enrichment analysis. The pathway reference for the main figure is the BioPlanet 2019 gene-set library (`BioPlanet_2019.gmt`). Each original pathway gene set is intersected with the 2,000 modeled genes, which also form the test background. Terms with fewer than five or more than 500 intersecting genes are excluded.

For a module containing n genes and a pathway containing t background genes, let o be their observed overlap. A one-sided hypergeometric test computes the probability of at least this overlap,

$$\Pr(J \geq o) = \sum_{j=o}^{\min(n,t)} \frac{\binom{t}{j}\binom{2000-t}{n-j}}{\binom{2000}{n}}. \tag{28}$$

Benjamini–Hochberg correction is applied over all retained pathway terms separately within each module and gene-set library. A module has significant enrichment if at least one original term has $q < 0.05$. Broad function groups and shortened pathway names are display labels only and do not replace the terms used in testing.

For the display filter, gene symbols are matched to HGNC annotations, including aliases and previous symbols. The non-coding/pseudogene fraction counts genes annotated as `non-coding RNA` or `pseudogene`, divided by the full module size. Unmatched genes are recorded separately rather than assumed to be protein coding. Of the 32 modules, 27 have a fraction below 30%, and 18 of

Table 8: Control-state routing across five held-out PBMC donors under the common 12-cell-type protocol. The adaptive fraction is conditional on the fixed expression and split-half confidence criteria.

Held-out donor	Adaptive/comparable	Adaptive fraction	Effective modules / 32
1	243/480	50.6%	28.6
2	259/500	51.8%	26.9
3	239/444	53.8%	28.7
4	257/490	52.4%	28.0
5	277/502	55.2%	25.4

these have significant BioPlanet enrichment. The main panel displays the modules meeting both criteria. Its columns identify modules, rows identify pathways, colors denote broad display-only function groups, and bubble area represents $-\log_{10}(q)$. The complete figure below includes all 32 modules and their non-coding composition. These display filters are not used during model training or module construction.

Control-State Adaptivity Protocol and Replication. For each held-out donor, we route two disjoint 128-cell halves of PBS cells within each eligible cell type using that donor's leave-donor-out tokenizer. The tokenizer is frozen for all cell types within a donor. Donor1 sampling uses seed 44 and selects 256 cells per type without replacement before dividing them into two halves. For half h of cell type t, the mean soft routing weights are

$$\bar{A}_{gk}^{(h,t)} = \frac{1}{128} \sum_{\mathbf{x} \in \mathcal{B}_{h,t}} A_{gk}(\mathbf{x}), \qquad h \in \{1, 2\}. \tag{29}$$

These are assignment probabilities, not expression-weighted contributions $A_{gk}(\mathbf{x})x_g$. We call the dominant module stable only if both halves have mean log-normalized expression at least 0.5, maximum mean assignment at least 0.5, top-minus-second assignment at least 0.2, and the same top module. A gene is comparable if stable in at least two cell types, and adaptive if its dominant module differs between any two such types. These thresholds are applied before summarizing Donor1–5. The main Donor1 heatmap uses 14 cell types; the shared five-donor analysis uses the 12 cell types with at least 256 PBS cells in every donor, excluding pDC and Treg. Table 8 summarizes the latter cohort. The effective module number is $\exp(-\sum_k p_k \log p_k)$, where p_k is the fraction of stable gene–cell-type dominant assignments to module k among comparable genes. This measures pooled utilization rather than per-cell module diversity. For each stable gene–cell-type pair, let $m_g^{(t)}$ be the module with the largest mean routing probability in cell type t. The adaptive fraction is the proportion of comparable genes with more than one dominant module across their eligible cell types. For a pair of cell types t, t', let $\mathcal{C}_{tt'}$ contain genes stable in both; the module-switch rate is

$$\mathrm{MSR}(t, t') = \frac{\left|\{g \in \mathcal{C}_{tt'} : m_g^{(t)} \neq m_g^{(t')}\}\right|}{|\mathcal{C}_{tt'}|}. \tag{30}$$

Each heatmap entry is this pairwise fraction. Cell types are ordered by predefined immune-family annotations, rather than clustered using the measured routing values. Colored underlines mark those groups. The diagonal is a self-comparison, while near-white off-diagonal entries indicate little or no switching. Pairwise denominators range from 25 to 189, so the displayed fractions are interpreted with their eligible-gene counts. CD4 Naive versus CD8 Naive gives $0/101$, whereas B Naive versus CD14 monocytes gives $52/67$ (77.6%).

In the 14-cell-type Donor1 analysis, 502 genes are comparable and 265 (52.8%) have different dominant modules across at least two eligible types. Their 2,295 stable gene–cell-type assignments cover all 32 modules, with 30 modules each receiving at least 1% of calls. The effective module number is 28.7 and the largest module accounts for 6.7% of calls. These statistics describe pooled utilization, not uniform module use within every cell type.

All five donors use all 32 modules somewhere among these stable dominant assignments. This excludes global dominant-module collapse under this operational criterion, but not cell-type-specific concentration: Donor5 cDCs have an effective number of 9.5/32 and assign 50.8% of stable calls

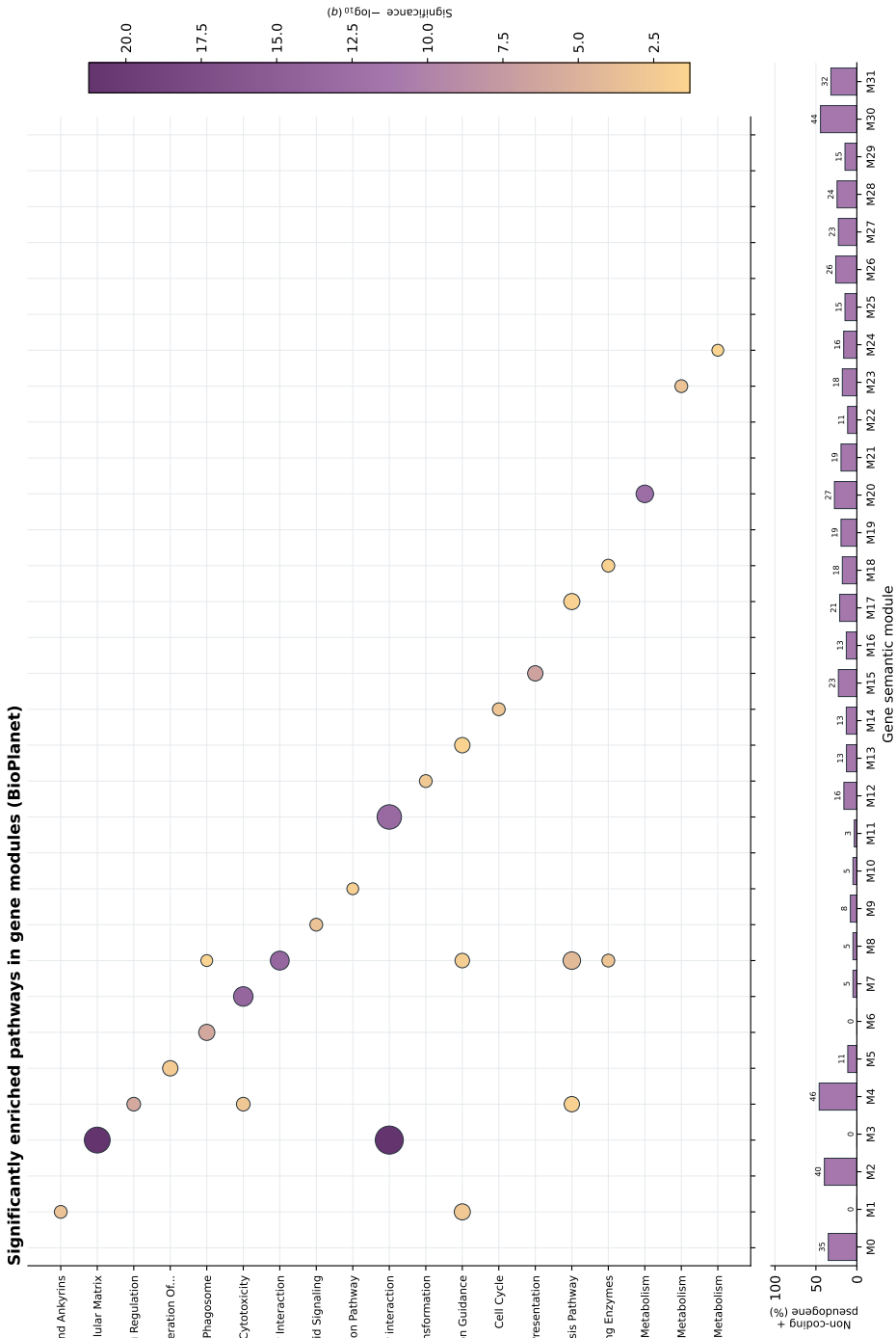


Figure 4: **Complete BioPlanet enrichment of static prior modules.** All 32 modules are shown with their non-coding and pseudogene fractions. Marker color gives $-\log_{10}(q)$ and area gives gene overlap.

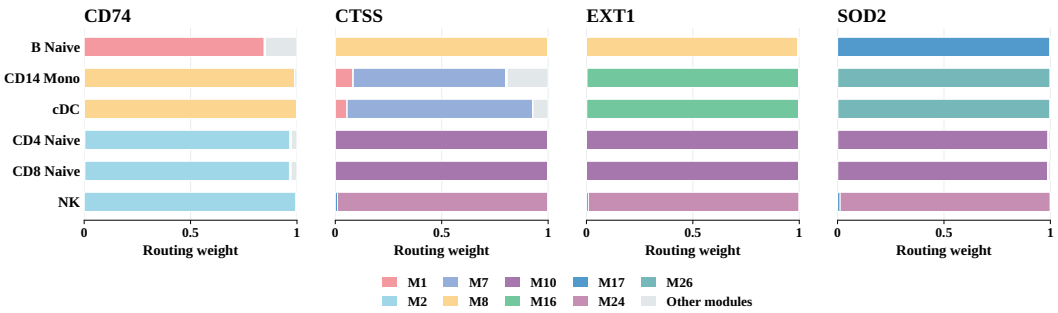


Figure 5: **Control-state gene-to-module routing examples in held-out Donor1.** Soft routing weights for four genes across the same six cell types. Colors identify modules; gray pools the remaining modules. All displayed gene–cell-type pairs pass the split-half confidence criterion.

to one module. Among 310 genes comparable in all five donors, 110 are adaptive in all five. The balanced prior may contribute to broad pooled utilization. These pooled statistics are distinct from within-cell-type concentration.

Gene-Level Routing Examples. For CD74, CTSS, EXT1, and SOD2, the plotted weights are the averages of the two half-specific mean routing vectors, using the same six cell types and module colors in every panel. All displayed gene–cell-type pairs pass the split-half stability criteria. EXT1 and SOD2 were selected as recognizable genes with distinct stable routing patterns, not by optimizing a prediction metric. Gray pools the undisplayed module weights. CD74 has dominant assignments to M1 in B Naive, M8 in CD14 monocytes and cDCs, and M2 in naive T cells. These examples show changes in participation across shared modules, rather than changes in expression-weighted module contributions.

Perturbation Module Readout Cases. We use the trained leave-Donor1-out PMR and its frozen decoder anchors to extract one gate vector $\mathbf{g}_p$ for each of the 90 non-PBS cytokines. The corresponding condition embedding is passed through the learned condition query and anchor projections, with normalization, temperature scaling, and softmax as specified in Appendix A.6.1. This vector depends on the condition rather than on a particular cell, so no cell-level averaging is required. The gate weights sum to one across all 32 modules.

We rank modules by gate weight for each cytokine and match its top three modules to the independently computed static-prior BioPlanet enrichment table, retaining terms with $q < 0.05$. Literature associations are reviewed separately from this numerical screen. IFN-γ, IFN-α1, and M-CSF are selected as three literature-linked examples whose highest-weight module is M16. This is a qualitative case selection, not an exhaustive estimate of agreement across cytokines.

IFN-γ, IFN-α1, and M-CSF each assign their highest PMR gate weight to M16 (0.561, 0.599, and 0.386, respectively; Figure 6). The independent BioPlanet enrichment of M16's static prior genes includes MHC class II antigen presentation ($q = 2.24 \times 10^{-7}$; eight overlapping genes). Previous human-monocyte studies link IFN-γ and IFN-α to MHC-II or HLA-DR expression (Lee et al., 2017; Gkirtzimanaki et al., 2018; Hartigan-O'Connor et al., 2014). M-CSF also supports differentiation into HLA-DR-positive monocyte-derived macrophages (Kandalla et al., 2023). The cited IFN-α studies concern the cytokine family rather than the IFN-α1 subtype specifically. In the figure, stacked segment widths give the gate weights of M16, M11, M0, and M31, with the remaining modules pooled as Other. The enrichment panel reports the three most significant original BioPlanet terms for M16 using $-\log_{10}(q)$. These selected examples illustrate convergence on one module rather than distinct responses for each cytokine. Across the full screen, 63 of the 90 cytokines select M11, whose prior lacks a comparably clear immune-pathway annotation. The gate is used here as a condition-dependent readout weight.

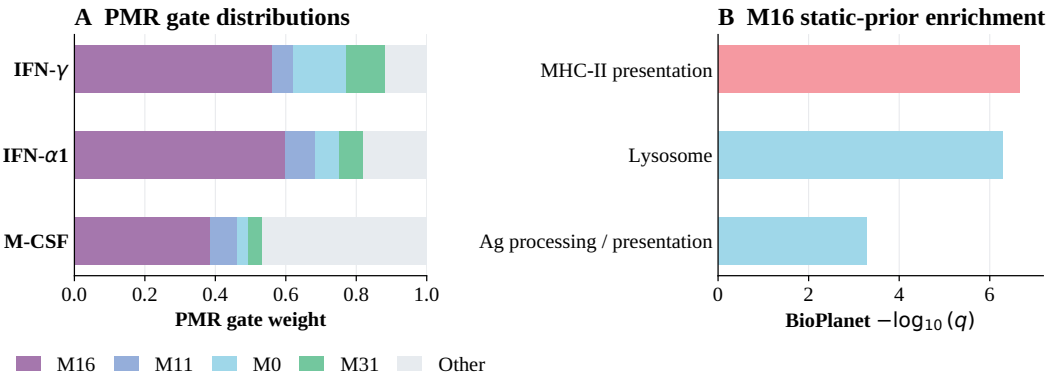


Figure 6: PMR gate distributions for three held-out Donor 1 cytokines and independent BioPlanet enrichment of their shared top-gate module M16.

B.4 Additional Ablations and Robustness

Autoencoder Reconstruction. On the 20,173 held-out SciPlex3 cells, the AE achieves a full-gene reconstruction MSE of 0.00483, compared with 0.00914 for PCA. Both use 128-dimensional representations of the same normalized log1p expression; PCA is fitted only on the training cells.

Seed Variance. To assess run-to-run variability, we train scAMod with five different random seeds and report the SciPlex3 held-out-drug results in Table 9. The final two rows give the mean and sample standard deviation across seeds. DEG R^2 is 0.4364 ± 0.0147, while DEG ED is 0.7373 ± 0.0106.

Table 9: SciPlex3 held-out-drug performance across five training seeds.

	R^2 ↑		ED ↓		MMD ↓		MSE ↓	
Seed	Full	DEG	Full	DEG	Full	DEG	Full	DEG
1	0.8653	0.4517	0.7083	0.7352	0.02362	0.04152	0.002387	0.06453
2	0.8689	0.4198	0.7128	0.7453	0.02544	0.04331	0.002732	0.06316
3	0.8776	0.4225	0.7161	0.7511	0.02761	0.03944	0.002660	0.05950
4	0.8661	0.4487	0.7145	0.7283	0.02557	0.04121	0.002432	0.06259
5	0.8579	0.4394	0.7214	0.7268	0.02182	0.04238	0.002432	0.06131
Mean	0.8672	0.4364	0.7146	0.7373	0.02481	0.04157	0.002529	0.06222
SD	0.0071	0.0147	0.0048	0.0106	0.00219	0.00145	0.000156	0.00191

Computational Efficiency. We measured warmed training throughput on one A100 80GB GPU using the SciPlex3 stage configurations. Data preparation, compilation, validation, and checkpointing are excluded.

Stage	Cells per batch	Batches/s
AE	512	10.93
Flow	1024	64.06
PMR	32	683.46